

**OPTIMALISASI KINERJA SISTEM SEKURITAS RUMAH  
MODERN BERBASIS JARINGAN SARAF TIRUAN (JST)**

***PERFORMANCE OPTIMALIZATION OF MODERN HOUSE  
SECURITY SYSTEM BASED BACKPROPAGATION NEURAL  
NETWORK (BNN)***

**ANTHOINETE P.Y WAROH  
P2700211443**



**PROGRAM PASCASARJANA  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2013**

**OPTIMALISASI KINERJA SISTEM SEKURITAS RUMAH  
MODERN BERBASIS JARINGAN SARAF TIRUAN (JST)**

Tesis

Sebagai Salah Satu Syarat Untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan diajukan oleh

**Anthoinete P.Y. Waroh**

Kepada

**PROGRAM PASCASARJANA  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2013**

## PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan dibawah ini:

Nama : Anthoinete P.Y. Waror

Nomor mahasiswa : P2700211443

Program Studi : Teknik Elektro

Konsentrasi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, Agustus 2013

Yang menyatakan

Anthoinete P.Y. Waroh

## PRAKATA

Pujian dan Hormat penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena hanya perkenaanNya sehingga saya dapat menyelesaikan Tesis yang berjudul "Optimalisasi Kinerja Sistem Sekuritas Rumah Modern Berbasis Jaringan Saraf Tiruan (JST)".

Diharapkan melalui Tesis ini dapat memberikan masukan pada masyarakat bahwa dibutuhkan suatu Teknologi yang dapat mengoptimalkan sistem untuk keamanan rumah yang lebih baik dan terhindar dari pencurian .

Besar Harapan Penulis bahwa proposal yang merupakan pengajuan penelitian ini dapat diterima berdasarkan relevansi yang nyata di masyarakat. Pada kesempatan ini kami ucapkan terima kasih yang sebesar-besarnya kepada:

1. Suamiku tercinta Saul Yan Makalew serta anakku marcellino makalew dan keluargaku yang telah memberikan doa dan motivasi kepada penulis selama study sampai penulisan tesis.
2. Bapa Prof.Dr.Ir. Salama Manjang,MT, Selaku Ketua Program Study.
3. Bapak Dr.Ir.Andani Achmad,MT,Selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Hasanuddin.
4. Bapak Dr.Elyas Palantei,ST.M.Eng.Selaku pembimbing I.
5. Bapak Dr. Adnan Achmad,MT.Selaku pembimbing II
6. Seluruh dosen pengajar program pascasarjana program studi teknik elektro universitas hasanuddin.
7. Seluruh staf karyawan Jurusan Teknik Elektro Fakultas Teknik Universitas Hasanuddin atas bantuan yang diberikan kepada saya.
8. Kepada rekan-rekan seangkatan,Tracy Kereh, Ali Ramchie, Ronny Katuuk, Olga Melo, Sulastri Eksan, Daud Salemba, Tony

Wungkana, Johan Pongoh, Anritsu Polii, Doostenreyk Kantohe, Robby Lumbu, Nikita Sajangbati, Harson Kapoh, Robby Tangkudung.

9. Kepada Marson Budiman MT ,Veny Pongawa MT bersedia menjadi objek penelitain tesis saya.

Akhir kata semoga tesis ini dapat memberikan manfaat bagi pembaca, khususnya para pengguna yang ingin menggunakan sistem ini agar saat ditinggal pergi rumah tetap aman.

Manado, Agustus 2013

Penulis,

Anthoinete P.Y Waroh

## **ABSTRAK**

**ANTHOINETE P.Y.WAROH .** Optimalisasi Kinerja Sistem Sekuritas Rumah Modern Berbasis Jaringan Saraf Tiruan (JST) (dibimbing oleh Elyas Palantei dan Adnan).

Perkembangan yang pesat dibidang teknologi komputer, elektronik, telekomunikasi maupun mekanik telah menghasilkan berbagai aplikasi canggih . Tujuan penelitian ini untuk membuat sistem sekuritas masuk rumah modern yang cerdas dengan memanfaatkan pola pencitraan wajah, menganalisis akses pencitraan yang cerdas dan membuat prototype dari sistem agar teroptimalisasi kinerjanya. Metode yang digunakan adalah analisis akuisisi. Sistem pengenalan wajah menggunakan citra wajah yang diambil dari 7 wajah manusia dengan 5 posisi wajah yang berbeda-beda, yaitu posisi biasa, tersenyum, serong ke kiri, serong ke kanan, dan mulut terbuka. Dalam prakteknya, algoritma backpropagation capturing (perekaman) wajah dengan jarak konstan dikumpulkan melalui proses mengakses data terhadap citra wajah dengan jarak konstan 30 cm ( dekat kamera webcam) dengan kondisi pencahayaan yang relatif sama. Selanjutnya diadakan pengujian dan pengukuran kepada 7 wajah manusia yang tidak dikenal (tidak diteliti). Hasil penelitian menunjukkan bahwa dari 35 citra wajah yang diteliti dan 35 citra wajah yang tidak diteliti memiliki ukuran yang bervariasi, yaitu mulai dari 300 sampai dengan 500 piksel. Dapat disimpulkan bahwa klasifikasi yang menunjukkan angka keberhasilannya adalah ketika diaplikasikan pada wajah yang dikenal oleh sistem maka sinyal lampu akan menyala (pintu terbuka) dan ketika wajah tak dikenalnya lampu akan mati (pintu tertutup). Data pengujian keberhasilan pengenalan pola wajah 96% dan yang tidak sesuai pengenalan pola wajah 98 %.Sistem yang dibuat adalah sebuah sekuritas rumah modern berbasis jaringan saraf tiruan (JST) yang diuji pada software matlab R2009a dengan pengenalan pola citra wajah face parttern recognition.

Kata kunci : algoritma backpropagation, pengenalan pola wajah.

**ABSTRACT**

**ANTHOINETE P.Y.WAROH.** *Performance Optimization of Modern House Security System Based Backpropagation Neural Network (BBN) (supervised by Elyas Palantei and Adnan).*

*Security is staying home to be a depressing thing coveted by everyone, at the time of leaving the house to do a trip. The face recognizing system used the face image taken from 7 human faces with 5 different face positions i.e.the normal face position, smiling face, left tilt face, right tilt face and face with open mouth. In practice, the back propagation capturing algorithm (recording) of the faces with the constant distance was collected through the data accessing process on the face image with the constant distance of 30 cm (near to webcam camera),with relatively similar lightning condition, the examination and measurement were then carried out of the 7 human faces determined and 7 human faces who were not known by the program because they were entered into the image taking data through the research process, this was proven that the system recognize 35 face images determined and 35 face which were not entered into data system. The image measurements could be various starting from 300 to 500 pixel. It can be concluded the classification indicated the success figures, the application when the faces were recognized, the light signal was on (the door opened), and if the faces were not recognized the light would be off ( the door opened).The success examination data of the face pattern recognition was 96 % and which was not suitable with the face pattern recognition was 98%. The research result indicates that the system made is the artificial nerve network based a modern house security which is examined on the software matlab R2009a with the face pattern recognition.*

**Key-words :** *Back propagation Algorithm, face pattern recognition*

## DAFTAR ISI

PRAKATA .....	v
ABSTRAK .....	vii
ABSTRACT .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiii
DAFTAR LAMPIRAN .....	xiv
I. PENDAHULUAN .....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah .....	2
C. Tujuan Penelitian .....	2
D. Manfaat Penelitian .....	3
E. Ruang Lingkup Penelitian.....	3
II. TINJAUAN PUSTAKA .....	4
A. Road Map .....	4
B. Jaringan Syaraf Tiruan .....	8
C. Arsitektur Jaringan.....	10
1. Jaringan Layer Tunggal (single layer network).....	5
2. Jaringan Layer Jamak (multi layer network).....	11
3. Model JST Dua Lapisan Dengan Umpan Balik.....	12
4. Model JST Lapisan Kompetitif.....	13
D. Fungsi Aktifasi.....	14



E. Backpropagation .....	20
1. Arsitektur Backpropagation.....	20
2. Neuron.....	21
3. Threshold.....	22
4. Pelatihan Standart Backpropagation.....	23
5. Optimalitas Arsitektur Backpropagation.....	28
6. Variasi Backpropagation.....	31
F. Pemograman Backpropagation dengan Matlab .....	33
1. Membentuk Jaringan.....	33
G. Konsep Pemrosesan Citra .....	36
III. METODE PENELITIAN .....	40
A. Tempat Dan Waktu Penelitian .....	40
B. Jenis Penelitian Dan Metode Penelitian .....	40
C. Rancangan Penelitian .....	40
D. Rancangan Sistem .....	40
E. Pelatihan Backpropagation .....	45
F. Teknik Meningkatkan Kecepatan Proses Image Wajah Pada Sistem JST Backpropagation .....	49
G. Proses Awal Sistem .....	56
H. Proses Pembentukan Pola Jaringan Saraf Tiruan Propagasi Balik Dan Pelatihan .....	65
I. Pengujian Sistem melalui Lampu Indikator .....	66

IV. HASIL DAN PEMBAHASAN .....	69
A. Pendeteksi Wajah .....	69
B. Proses Inisial Dan Input Wajah .....	69
C. Proses Image-Neural Network .....	72
D. Pengujian Kondisi Citra Biner.....	75
E. Neural Network.....	81
F. Kelemahan Sistem Yang Dibangun.....	129
V. PENUTUP .....	131
1. Kesimpulan.....	131
2. Saran.....	132

Daftar Pustaka

Lampiran

## DAFTAR GAMBAR

<b>Nomor</b>	<b>Halaman</b>
1. Diagram Alir riset JST untuk rumah modern.....	4
2. Model Sederhana Jaringan Syaraf Tiruan .....	9
3. Jaringan Layar Tunggal.....	10
4. Jaringan Layar Jamak .....	11
5. Model JST Dua Lapisan Dengan Umpan Balik .....	13
6. Model JST Lapisan Kompetitif .....	13
7. Struktur Neuron Jaringan Syaraf .....	14
8. Fungsi Aktivasi Pada Jaringan Syaraf Sederhana.....	16
9. Fungsi Aktivasi Sigmoid Biner .....	18
10. Fungsi Aktivasi Sigmoid Bipolar .....	19
11. Fungsi Aktivasi Linear (Identitas).....	19
12. Arsitektur Backpropagation .....	21
13. Konfigurasi Sistem Sekuritas Rumah Modern Berbasis JST ....	41
14. Tampilan aplikasi matlab .....	43
15. Diagram alir pengenalan citra wajah .....	55
16. Diagram alir pengidentifikasian wajah.....	56
17. Diagram alir pengidentifikasian wajah (lanjutan) .....	57
18. Aplikasi pengenalan wajah .....	70
19. Pemasukan citra wajah sebagai input .....	72
20. Tampilan hasil Citra wajah warna yang sudah	

diproses dalam program.....	73
21. Citra wajah dalam bentuk citra <i>Grayscale</i> , dalam bentuk citra biner dan nilai biner.....	73
22. Wajah dikenal Sinyal yang keluar pada PC Lampu Hidup.....	76
23. Citra wajah dalam bentuk citra <i>Grayscale</i> , citra biner dan nilai biner.....	76
24. Wajah Tidak dikenal Sinyal yang keluar pada PC Lampu Mati.....	77
25. Tampilan hasil training JST.....	81

## DAFTAR TABEL

<b>Nomor</b>	<b>Halaman</b>
1. Citra wajah dan data referensi.....	71
2. karakter objek wajah.....	74
3. hasil pengujian citra wajah pertama .....	78
4. hasil pengujian citra wajah .....	80
5. Hasil uji data testing sesuai input wajah untuk inisial eta .....	82
6. Hasil uji data testing sesuai input wajah untuk inisial lino ... ..	85
7. Hasil uji data testing sesuai input wajah untuk inisial eta.....	89
8. Hasil uji data testing sesuai input wajah untuk inisial mami.....	92
9. Hasil uji data testing sesuai input wajah untuk inisial jeine .....	95
10. Hasil uji data testing sesuai input wajah untuk inisial veny .....	99
11. Hasil uji data testing tidak sesuai input wajah marson.....	102
12. Hasil uji data testing tidak sesuai input wajah inisial bela.....	106
13. Hasil uji data testing tidak sesuai input wajah untuk orang yang berinisial seren .....	110
14. Hasil uji data testing wajahnya inisial meisi mirip inisial eta .....	113
15. Hasil uji data testing tidak sesuai input wajah untuk inisial felix ..	116
16. Hasil uji data testing tidak sesuai input wajah untuk inisial asrid .	120
17. Hasil uji data testing tidak sesuai input wajah untuk inisial asrid .	123
18. Hasil uji data testing tidak sesuai input wajah .....	126

**DAFTAR LAMPIRAN**

<b>Nomor</b>		<b>Halaman</b>
1.	Lampiran 1 .....	134
2.	Lampiran 2 .....	158
3.	Lampiran 3.....	165

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Perkembangan yang pesat dibidang teknologi komputer, elektronik, telekomunikasi maupun mekanik telah menghasilkan berbagai aplikasi canggih dan cerdas yang merubah kehidupan manusia pada saat ini dan mendatang khususnya di Indonesia.

Sejumlah literature dan peneliti sebelumnya membuat aplikasi dengan metode segmentasi warna kulit, untuk pencitraan wajah, kekurangannya disaat warna kulit berubah makah sistem tidak mengenali orang tersebut. Peneliti kedua pengenalan wajah secara real time menggunakan algoritma (JST) dengan wajah tampak depannya saja , untuk mengaktifkan pintu terbuka dan tertutup, program tidak mengenal wajah tampak samping kiri dan wajah samping kanan wajah manusia.

Sehingga para pelaku tindak kriminal tetap melakukan kejahatan didalam rumah mereka memiliki cara tersendiri untuk mengatasi hal tersebut sehingga sangat sulit untuk melindungi rumah terutama pintu rumah.

Untuk mengatasi permasalahan diatas, perlu dibuat suatu sistim pengaman pintu secara otomatis yang hanya dapat mengenali wajah yang keterkaitan yang direset terlebih dahulu mendapatkan hak akses

masuk rumah modern. Pada tesis ini, dibuat suatu sistem identifikasi wajah seseorang dengan metode JST (Jaringan Saraf Tiruan) menggunakan algoritma pola pengenalan imege wajah (face recognition) BackPropagation sebagai parameter untuk membuka pintu. Pelatihan dilakukan terhadap citra wajah yang berjarak konstan 30 cm dengan pencahayaan yang diatur konstan pada webcam TECGO.

Adapun sensor yang digunakan adalah sensor kamera webcam , dimana data output dari sensor, kemudian data input tersebut dibandingkan dengan nilai set point (data referensi foto wajah) untuk proses akses masuk rumah modern.

## **B. Rumusan Masalah**

Kajian tentang sistem sekuritas rumah moderen atau biasa dikenal sebagai smart home yang mengoptimalisasi sistem jaringan saraf tiruan yang cukup luas cakupannya, maka perlu dilakukan pembatasan-pembatasan pada sejumlah permasalahan yang akan diteliti, diantaranya:

1. Pencitraan wajah dengan pencahayaan yang konstan pada sistem sekuriti akses masuk rumah moderen.
2. Pendeteksian wajah pada ke lima posisi dengan jarak konstan 30 cm.
3. Konsep dasar pengenalan pola berbasis Jaringan Saraf Tiruan dengan algoritma backpropagation.

## **C. Tujuan Penelitian**

Penelitian ini bertujuan untuk :



1. Membuat suatu sistem sekuritas masuk rumah modern yang cerdas dengan memanfaatkan pola pencitraan wajah
2. Menganalisis teknik akuisisi data untuk suatu sistem akses pencitraan yang cerdas
3. Membuatan prototype dari sistem agar dapat teroptimalisasi kinerja sistem

#### **D. Manfaat Penelitian**

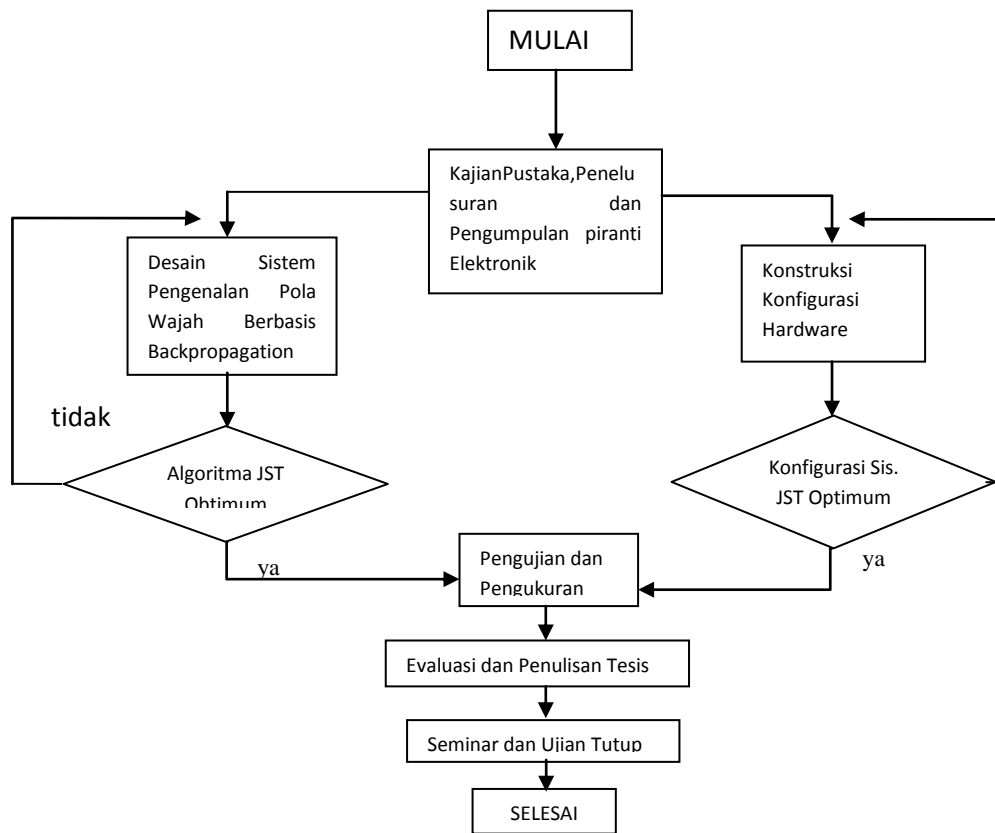
Dengan adanya penelitian ini diharapkan dapat memberikan situasi yang aman bagi penghuni rumah moderen dan adapun manfaat bagi penulis yaitu :

1. Menciptakan sistem sekuriti yang cerdas dan aman
2. Mendesain Teknik akuisisi data digital secara programmable
3. Dapat mengoptimalisasi kerja sistem sekuritas rumah moderen

#### **E. Ruang Lingkup Penelitian**

Sistem sekuritas rumah modern berbasis jaringan saraf tiruan bekerja secara optimal, Adapun cakupan batasan penelitian saya adalah :

1. Desain Kontrol mencakup: control pintu bagi pengguna dalam mengakses masuk ruangan,
2. Sekuriti akses masuk mencakup: deteksi citra wajah dengan metode JST backpropagation, dalam kondisi lima fariasi wajah.



**Gambar 1.** Diagram Alir riset JST untuk rumah modern

Pada gambar.1, menjelaskan tahapan-tahapan yang akan dilaksanakan dalam proses penelitian pada rumah modern, langkah pertama yaitu memulai dengan mengkaji materi sesuai dengan tesis dan mengumpulkan piranti elektronik. langkah kedua ada dua tahapan yaitu pertama mendesain sistem pengenalan pola wajah berbasis backpropagation, kemudian membuat algoritma JST optimum jika sudah diproses dan benar (ya) lanjut pengujian program jika tidak maka kembali ke desain sistem dan pengenalan pola wajah. Tahapan kedua konstruksi konfigurasi hardware sistem JST memproses konfigurasi sistem JST optimum jika tidak maka kembali ke konstruksi konfigurasi hardware

sistem JST. Jika terkonfigurasi sistem JST optimum maka proses pengujian dan pengukuran dilakukan. Langkah ketiga evaluasi dan penulisan tesis, Langkah keempat seminar dan ujian tutup setelah proses ini maka penelitian ini selesai dilaksanakan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Road Map**

Penelitian yang terkait dalam desain control sekuritas akses masuk rumah modern yang akan saya buat sebagai berikut:

1. Sistem deteksi dan verifikasi wajah pada sistem pembuka pintu (SOFTWARE). EEPIS Final Project.

Hasil yang diperoleh : Aplikasi penelitian ini mengenai deteksi wajah menggunakan metode segmentasi warna kulit, pencarian nilai bobot untuk data referensi menggunakan LVQ (Learning Vector Quantization) dan algoritma verifikasi (pengenalan pola wajah) wajah atau bukan wajah menggunakan metode Euclidian distance dan titik kritis distribusi. Kelemahan pada peneliti ini yaitu pada metode pendeteksi warna kulit, dimana warna kulit dari seseorang akan berubah sesuai dengan keadaan orang tersebut, contohnya jika orang tersebut berada dalam keadaan yang kurang sehat, maka warna kulit dari orang tersebut akan berubah, hal ini akan membuat sistim tidak dapat mengenali orang tersebut.

Sistem pengenalan wajah yang akan dibangun oleh penulis yaitu sistem yang dapat mendeteksi wajah orang yang ditetapkan pada inputnya berdasarkan bentuk wajah orang tersebut, disamping itu set

point (referensi) yang digunakan lebih dari lima posisi dengan 7 citra wajah untuk keakurasian data.

Leonardo , Kamajaya and Eru Puspita, Puspita and Ronny , Susetyoko (2009) (<http://www.cepis.its.edu/id/ta/776/>)

2. Sistem pengenalan wajah manusia real-time menggunakan algoritma jaringan saraf tiruan .

Hasil dari penelitian ini yaitu sistem dapat mengenal wajah dengan metode jaringan saraf tiruan (JST) mengenal wajah tampak depannya saja ketika tampak samping kiri dan samping kanan wajah program tidak dapat mengenal, saya akan membuat sistem dengan metode yang sama, dan di kembangkan lagi yaitu dapat mengenal wajah sesuai dengan yang di akses yaitu wajah biasa, wajah tersenyum, wajah serong kiri, wajah serong kanan, wajah dengan mulut terbuka aplikasinya lampu akan menyala jika wajah dikenal dan pintu akan tertutup jika wajah tidak dikenal dan pintu akan tertutup.

(Abdul Fadlil, Ikhson Hidayat, Sunardi ) dapat di lihat pada alamat email ([jurnal.uii.id/index.php/snati/article/view/881/842](http://jurnal.uii.id/index.php/snati/article/view/881/842))

3. Prototype mengenal wajah melalui webcam dengan menggunakan algoritma principal component analysis (PCA) dan Linier discriminant analisis (LDA)

Hasil jurnal ini yaitu pengenalan wajah melalui webcam dengan metode PCA dan LDA.

Oleh : Jimmy E Purwanto

<http://journal.ipb.ac.id/index.php/artical/3202/2158>)

4. Sistem pengenalan wajah real-time ruang eigen dengan segmentasi berdasarkan warna kulit.

Penelitian ini dapat menghasilkan system yang dapat mengenal wajah, deteksi waja dengan metode jaringan syaraf tiruan (JST), Agus Bueno, Amad Ridha, Harief Batian)

<http://journal.ipb.ac.id/index.php/artical/3202/2158>)

Jurnal-jurnal penelitian ini menjadi acuan bagi saya untuk membuat desain suatu alat sistem sekuritas akses masuk rumah modern secara optimal, dari petunjuk dan saran dari peneliti diatas maka saya akan mengembangkan ide dari peneliti-peneliti sebelumnya, dengan judul “Optimalisasi Kinerja Sistem Sekuritas Ruamah Moderen Berbasis Jaringan Saraf Tiruan (JST)”.

## **B. Jaringan Syaraf Tiruan**

Jaringan syaraf tiruan (*artificial neural networks*) atau disingkat JST adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi.

JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi :

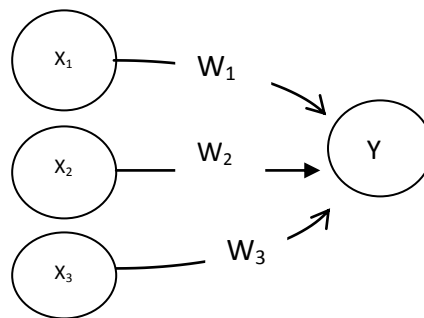
1. Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*)
2. Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung-penghubung

3. Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal
4. Untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batas ambang.

JST ditentukan oleh tiga hal :

- a. Pola hubungan antar *neuron* (disebut arsitektur jaringan)
- b. Metode untuk menentukan bobot penghubung (disebut metode *training* / *learning* / algoritma)
- c. Fungsi aktivasi

Sebagai contoh, perhatikan *neuron* Y pada Gambar 2.



**Gambar 2.** Model Sederhana Jaringan Syaraf Tiruan

Y menerima *input* dari *neuron*  $x_1$ ,  $x_2$  dan  $x_3$  dengan bobot hubungan masing-masing adalah  $w_1$ ,  $w_2$  dan  $w_3$ . Ketiga impuls *neuron* yang ada dijumlahkan

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 \quad (1)$$

Besarnya impuls yang diterima oleh  $Y$  mengikuti fungsi aktivasi  $y = f(\text{net})$ . Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

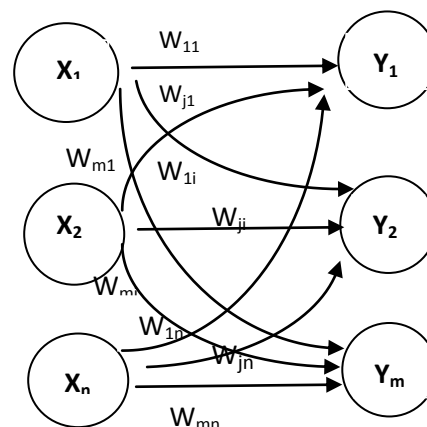
### C. Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain :

#### 1. Jaringan Layar Tunggal (*single layer network*)

JST dengan layar tunggal pertamakali dirancang oleh Widrow dan Hoff pada tahun 1960. Walaupun JST layar tunggal ini sangat terbatas penggunaannya, namun konsep dan gagasannya banyak dipakai oleh beberapa pakar untuk membuat model JST layar jamak.

Dalam jaringan ini, sekumpulan *input neuron* dihubungkan langsung dengan sekumpulan *output*. Dalam beberapa model (misal perceptron), hanya ada sebuah unit *neuron output*.



**Gambar 3.** Jaringan Layar Tunggal

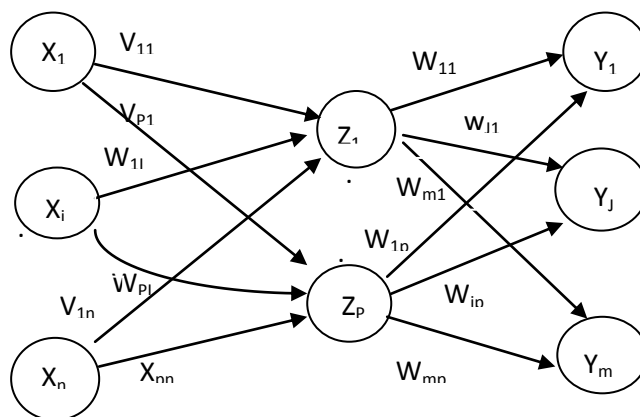


Gambar 2.2. menunjukkan arsitektur jaringan dengan  $n$  unit *input* ( $x_1, x_2, \dots, x_n$ ) dan  $m$  buah unit *output* ( $Y_1, Y_2, \dots, Y_m$ ). Perhatikan bahwa dalam jaringan ini, semua unit *input* dihubungkan dengan semua unit *output*, meskipun dengan bobot yang berbeda-beda. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya. Demikian pula dengan unit *output*.

Besarnya  $w_{ji}$  menyatakan bobot hubungan antara unit ke- $i$  dalam *input* dengan unit ke- $j$  dalam *output*. Bobot-bobot ini saling independen. Selama proses pelatihan, bobot-bobot tersebut akan dimodifikasi untuk meningkatkan keakuratan hasil.

## 2. Jaringan Layer Jamak (*multi layer network*)

Jaringan layer (lapisan jamak merupakan perluasan dari layer tunggal. Dalam jaringan ini, selain unit *input* dan *output*, ada unit-unit lain (sering disebut lapisan tersembunyi). Dimungkinkan pula ada beberapa layer tersembunyi. Sama seperti pada unit *input* dan *output*, unit-unit dalam satu layer tidak saling berhubungan.



**Gambar 4.** Jaringan Layer Jamak

Gambar 4. adalah jaringan dengan  $n$  buah unit *input* ( $x_1, x_2, \dots, x_n$ ), sebuah layar tersembunyi yang terdiri dari  $p$  buah unit ( $z_1, \dots, z_p$ ) dan  $m$  buah unit *output* ( $Y_1, Y_2, \dots, Y_m$ )

Jaringan layar jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

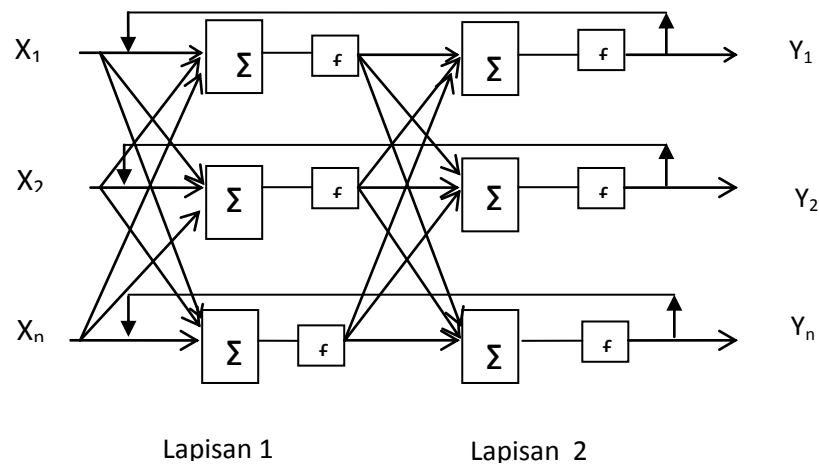
### 3. Model JST dua lapisan dengan umpan balik

Tokoh yang pertamakali mencetuskan ide tentang model jaringan syaraf tiruan dengan umpan balik adalah John Hopfield dari California Institute of Technology pada tahun 1982. Hopfield berpendapat bahwa kumpulan *neuron* tiruan dalam jumlah yang sangat besar dapat melakukan tugas-tugas tertentu.

Hopfield juga membandingkan antara jumlah *neuron* pada binatang dengan jumlah *neuron* diperkirakan sekitar 1000 buah dan bila dibandingkan dengan manusia, jumlah *neuron*-nya mencapai 100 trilyun buah. Sungguh jumlah yang sangat fantastis.

Dengan jumlah *neuron* yang sangat besar, JST memiliki sifat yaitu *fault tolerance*. Sifat ini mengandung maksud kerusakan sedikit atau sebagian pada sel-sel dalam jaringan tidak akan mempengaruhi *output* yang akan dikeluarkan.

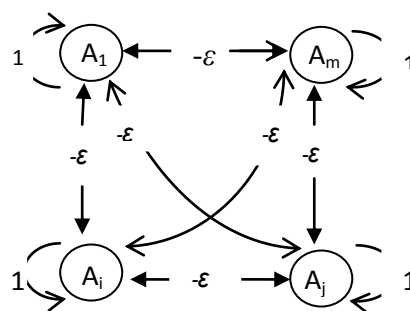
Model JST dua lapisan ini mempunyai sifat umpan balik, sehingga *output* yang dihasilkan akan mempengaruhi *input* yang akan masuk lagi ke dalam jaringan syaraf tersebut.



**Gambar 5.** Model JST Dua Lapisan Dengan Umpan Balik

#### 4. Model JST lapisan kompetitif

Bentuk dari lapisan kompetitif merupakan bagian dari jumlah yang besar pada jaringan syaraf. Pada dasarnya, hubungan antara *neuron* satu dengan *neuron* yang lain pada lapisan kompetitif tidak ditunjukkan secara arsitektur pada beberapa jaringan syaraf. Contoh dari model atau arsitektur lapisan kompetitif dapat dilihat pada Gambar 6, dimana koneksi dari lapisan tersebut memiliki bobot  $-\varepsilon$ .

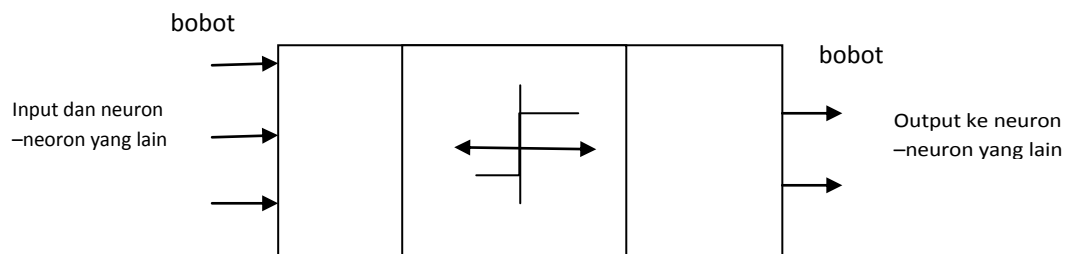


**Gambar 6.** Model JST Lapisan Kompetitif

### D. Fungsi Aktivasi

Jaringan syaraf merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa *neuron*, dan ada hubungan antara *neuron-neuron* tersebut. *Neuron-neuron* tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke *neuron-neuron* yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 7 menunjukkan struktur *neuron* pada jaringan syaraf.

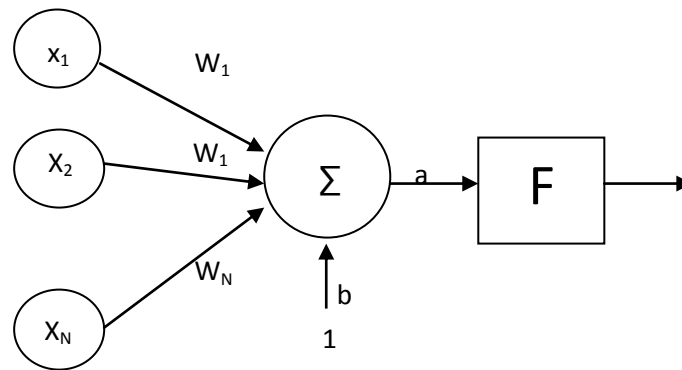


**Gambar 7.** Struktur Neuron Jaringan Syaraf

Jika dilihat, *neuron* buatan ini sebenarnya mirip dengan sel *neuron* biologis. *Neuron-neuron* buatan tersebut bekerja dengan cara yang sama pula dengan *neuron-neuron* biologis. Informasi (disebut dengan : *input*) akan dikirim ke *neuron* dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka *neuron* tersebut akan diaktifkan, tapi kalau tidak, maka *neuron* tersebut tidak akan diaktifkan. Apabila *neuron* tersebut diaktifkan, maka *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *output* nya ke semua *neuron* yang berhubungan dengannya. Demikian seterusnya.

Pada jaringan syaraf, *neuron-neuron* akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan *neuron* (*neuron layers*). Biasanya *neuron-neuron* pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara

mundur pada jaringan. Gambar 8 menunjukkan jaringan syaraf sederhana dengan fungsi aktivasi F.



**Gambar 8.** Fungsi Aktivasi Pada Jaringan Syaraf Sederhana

Pada Gambar 8. tersebut sebuah *neuron* akan mengolah  $N$  *input* ( $x_1, x_2, \dots, x_N$ ) yang masing-masing memiliki bobot  $w_1, w_2, \dots, w_N$  dan bobot bias  $b$ , dengan rumus :

$$a = b + \sum_{i=1}^N x_i w_i \quad (2)$$

kemudian fungsi aktivasi  $F$  akan mengaktivasi  $a$  menjadi *output* jaringan  $y$ .

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan. Fungsi Aktivasi yang digunakan pada *Backpropagation* antara lain :

- a. Fungsi sigmoid biner
- b. Fungsi sigmoid bipolar
- c. Fungsi linear

a. Fungsi sigmoid biner

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki range (0,1).

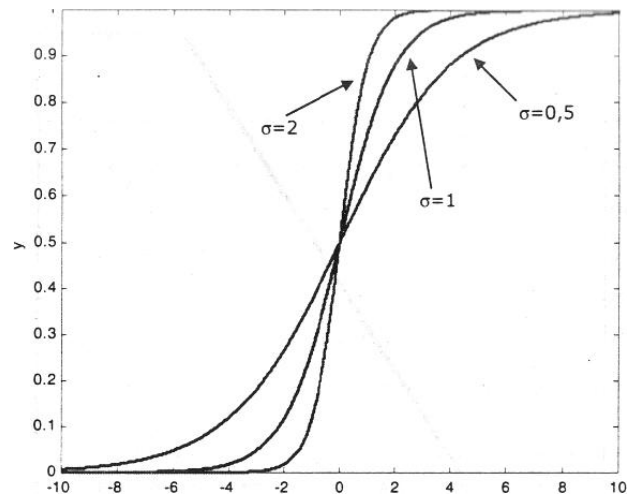
Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai *output* nya 0 atau 1 (Gambar 9).

Fungsi sigmoid biner dirumuskan sebagai :

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

dengan :

$$f'(x) = \sigma(x) (1 - \sigma(x)) \quad (4)$$



**Gambar 9.** Fungsi Aktivasi Sigmoid Biner

b. Fungsi sigmoid bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja *output* dari fungsi ini memiliki *range* antara 1 sampai -1 (Gambar 10).

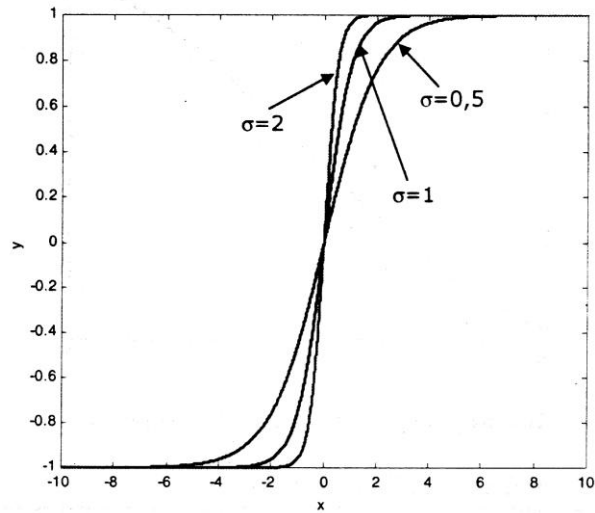
Fungsi sigmoid bipolar dirumuskan sebagai :

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

dengan:

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$





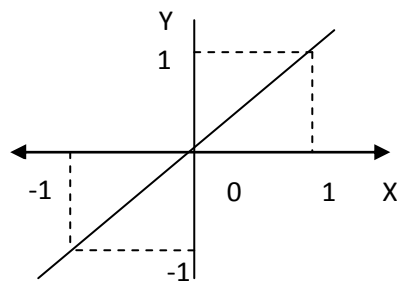
**Gambar 10.** Fungsi Aktivasi Sigmoid Bipolar

c. Fungsi linear (identitas)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *input* (Gambar 11).

Fungsi linear dirumuskan sebagai :

$$y = x$$



**Gambar 11.** Fungsi Aktivasi Linear (Identitas)

## E. Backpropagation

Kelemahan JST yang terdiri dari layer tunggal membuat perkembangan JST menjadi terhenti pada sekitar tahun 1970 an. Penemuan *backpropagation* yang terdiri dari beberapa layer membuka kembali cakrawala. Terlebih setelah berhasil ditemukannya berbagai aplikasi yang dapat diselesaikan dengan *backpropagation*, membuat JST semakin diminati orang.

JST dengan layer tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layer tersembunyi diantara layer masukan dan keluaran. Meskipun penggunaan lebih dari satu layer tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, tapi pelatihannya memerlukan waktu yang lama. Maka umumnya orang mulai mencoba dengan sebuah layer tersembunyi lebih dahulu.

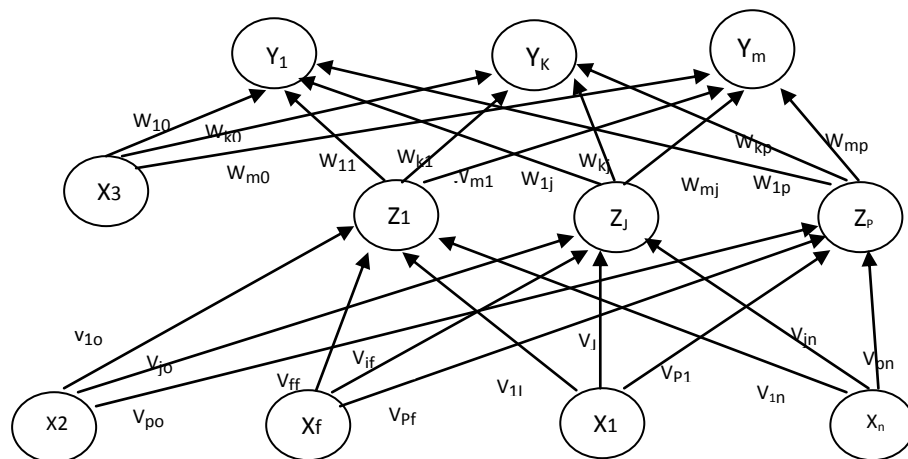
Seperti halnya model JST lain, *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

### 1. Arsitektur Backpropagation

*Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 12 adalah arsitektur *backpropagation*

dengan  $n$  buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit keluaran.

$v_{ji}$  merupakan bobot garis dari unit masukan  $x_i$  ke unit layar tersembunyi  $z_j$  ( $v_{j0}$  merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi  $z_j$ ).  $w_{kj}$  merupakan bobot dari unit layar tersembunyi  $z_j$  ke unit keluaran  $y_k$  ( $w_{k0}$  merupakan bobot dari bias di layar tersembunyi ke unit keluaran  $z_k$ ).



**Gambar 12.** Arsitektur Backpropagation

## 2. Neuron

*Neuron* adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan. *Neuron* terdiri dari tiga elemen pembentuk :

- a. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot/kekuatan yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawanya. Jumlah, struktur dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan (dan juga model jaringan yang terbentuk).
- b. Suatu unit penjumlahan yang akan menjumlahkan *input-input* sinyal yang sudah dikalikan dengan bobotnya.  
 Misalkan  $x_1, x_2, \dots, x_m$  adalah unit-unit *input* dan  $w_{j1}, w_{j2}, \dots, w_{jm}$  adalah bobot penghubung dari unit-unit tersebut ke unit keluaran  $Y_j$ , maka unit penjumlah akan memberikan keluaran sebesar  $u_j = x_1w_{j1} + x_2w_{j2} + \dots + x_mw_{jm}$
- c. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atautkah tidak.

### 3. Threshold

*Threshold* adalah suatu perubahan antara *gray-level image* dan *bilevel image*. *Bilevel image* terdiri dari banyak informasi yang penting dari sebuah *image* (contoh : angka, posisi dan bentuk objek), tetapi tidak dapat dibandingkan dengan informasi dari *gray-level image*. Kebanyakan dari *time pixels* dengan *gray levels* yang sama mempunyai objek yang sama juga. Selanjutnya, pengelompokan *image* oleh *pixel gray-level* dapat

memperkecil dan memudahkan beberapa pengoperasian proses *image* seperti membentuk *recognition* dan *classification*.

Beberapa operasi *thresholding* yang *essential* akan diseleksi oleh sebuah hasil tunggal *threshold*. Seluruh *gray-levels* menunjukkan hasil yang dapat diklasifikasikan menjadi *black* (0), dan disekelilingnya adalah *white* (1). Untuk waktu tertentu, bagian *image* tidak mungkin menjadi objek dan *background* dengan hasil tunggal *threshold* disebabkan oleh adanya *noise*. Pendekatan percobaan yang mudah harus menggunakan tujuan *gray-level* dalam *image* seperti *threshold*. Ini dapat disebabkan sebagian *pixels* menjadi *white* dan bagian lain menjadi *black*.

#### 4. Pelatihan Standar Backpropagation

Pelatihan *Backpropagation* meliputi tiga fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di lapisan keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

##### a. Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan ( $=x_i$ ) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan.

Keluaran dari setiap unit layar tersembunyi ( $=z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $=y_k$ ).

Berikutnya, keluaran jaringan ( $=y_k$ ) dibandingkan dengan target yang harus dicapai ( $=t_k$ ). Selisih  $t_k - y_k$  adalah kesalahan yang terjadi. 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 kesalahan yang terjadi.

b. Fase II : Propagasi mundur

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k$  ( $k = 1, 2, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase III : Perubahan bobot

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  *neuron* di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas  $\delta_k$  yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

*Algoritma Backpropagation*

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

$$z\_in_j = b1_j + \sum_{i=1}^n x_i v_{ij}$$

gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$z_j = f(z\_in_j)$$

dan 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.

$$y\_in_k = b2_k + \sum_{i=1}^n z_i w_{jk} \quad (6)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$y_k = f(y\_in_k) \quad (7)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Catatan :

Langkah (b) dilakukan sebanyak jumlah lapisan tersembunyi.

### *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 :

$$\delta 2_k = (t_k - y_k) f'(y\_in_k) \quad (8)$$



$$\varphi_{2jk} = \delta_k z_j \quad (9)$$

$$\beta_{2k} = \delta_k \quad (10)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ ) :

$$\Delta w_{jk} = \alpha \varphi_{2jk} \quad (11)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b_{2k}$ ) :

$$\Delta b_{2k} = \alpha \beta_{2k} \quad (12)$$

Langkah (d) ini juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi error dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

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

$$\delta_{in_j} = \sum_{k=1}^m \delta_{2k} w_{jk} \quad (13)$$

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

$$\delta_{1j} = \delta_{in_j} f'(z_{in_j}) \quad (15)$$

$$\varphi_{1ij} = \delta_{1j} x_j \quad (16)$$

$$\beta_{1j} = \delta_{1j} \quad (17)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ) :

$$\Delta v_{ij} = \alpha \varphi_{1ij} \quad (18)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b_{1j}$ ) :

$$\Delta b_{1j} = \alpha \beta_{1j} \quad (19)$$

f. Tiap-tiap unit *output* ( $Y_k$ ,  $k=1,2,3,\dots,m$ ) memperbaiki bias dan bobotnya ( $j=0,1,2,\dots,p$ ) :

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (20)$$

$$b_{2k}(\text{baru}) = b_{2k}(\text{lama}) + \Delta b_{2k} \quad (21)$$

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

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (22)$$

$$b_{1j}(\text{baru}) = b_{1j}(\text{lama}) + \Delta b_{1j} \quad (23)$$

4. Tes kondisi berhenti.

## 5. Optimalitas Arsitektur Backpropagation

Masalah utama yang dihadapi dalam *Backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu orang berusaha meneliti bagaimana parameter-parameter jaringan dibuat sehingga menghasilkan jumlah iterasi yang relatif lebih sedikit.

a. Inisialisasi bobot awal secara *random*

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global (atau mungkin hanya lokal saja)

terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Apabila nilai bobot awal terlalu besar, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Sebaiknya, apabila nilai bobot awal terlalu kecil, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan sangat kecil, yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara *random* dengan nilai antara -0.5 sampai 0.5 (atau -1 sampai 1, atau interval yang lainnya).

b. Jumlah unit tersembunyi

Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah layer tersembunyi sudah cukup bagi *Backpropagation* untuk mengenali sembarang pola antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah layer tersembunyi kadangkala membuat pelatihan lebih mudah.

Dalam propagasi maju, keluaran harus dihitung untuk tiap layer, dimulai dari layer tersembunyi paling bawah (terdekat dengan masukan). Sebaliknya, dalam propagasi mundur, faktor  $\delta$  perlu dihitung untuk tiap layer tersembunyi, dimulai dari layer keluaran.

c. Jumlah pola pelatihan

Tidak ada kepastian tentang berapa banyak pola yang diperlukan agar jaringan dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya bobot dalam jaringan

serta tingkat akurasi yang diharapkan. Aturan kasarnya dapat ditentukan berdasarkan rumusan :

$$\text{Jumlah pola} = \text{Jumlah bobot} / \text{tingkat akurasi}$$

Untuk jaringan dengan 80 bobot dan tingkat akurasi 0.1, maka 800 pola masukan diharapkan akan mampu mengenali dengan benar 90 % pola diantaranya.

d. Lama iterasi

Tujuan utama penggunaan *Backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0.

Umumnya data dibagi menjadi dua bagian, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan (misal setiap 10 *epoch*), kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi. Jaringan sudah mulai mengambil sifat yang hanya

dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

## 6. Variasi Backpropagation

Disamping model standar *Backpropagation*, kini sudah berkembang berbagai variasinya. Variasi tersebut bisa berupa model *Backpropagation* yang digunakan untuk keperluan khusus, atau teknik modifikasi bobot untuk mempercepat pelatihan dalam kasus tertentu.

### a. Momentum

Pada standar *Backpropagation*, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke (t+1) didasarkan atas bobot pada waktu t dan (t-1). Disini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika  $\mu$  adalah konstanta ( $0 \leq \mu \leq 1$ ) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan :

$$w_{kj}^{(+1)} = w_{kj}^{(+)} + \alpha \delta_k z_j + \mu (w_{kj}^{(+)} - w_{kj}^{(-)})$$

(24)

dan

$$v_{ji}^{(+1)} = v_{ji}^{(+)} + \alpha \delta_j x_i + \mu (v_{ji}^{(+)} - v_{ji}^{(-)}) \quad (25)$$

#### b. Delta - Bar - Delta

Dalam standar *Backpropagation*, laju pemahaman ( $\alpha$ ) merupakan suatu konstanta yang dipakai dalam seluruh iterasinya. Perubahan dapat dilakukan dengan memberikan laju pemahaman yang berbeda-beda untuk setiap bobotnya (atau bahkan laju pemahaman yang berbeda-beda untuk tiap bobot dalam tiap iterasinya). Apabila perubahan bobot berada dalam arah yang sama dalam beberapa pola terakhir (dapat dilihat dari tanda suku  $\delta_k z_j$  yang selalu sama), maka laju pemahaman yang bersesuaian dengan bobot  $w_{kj}$  ditambah. Sebaliknya apabila arah perubahan bobot dua pola terakhir berbeda (ditandai dengan suku  $\delta_k z_j$  yang berselang-seling positif - negatif) maka laju pemahaman untuk bobot tersebut harus dikurangi.

Perubahan bobot dalam aturan delta - bar - delta adalah sebagai berikut :

$$w_{kj}^{(k+1)} = w_{kj}^{(k)} + \alpha_{kj}^{(k+1)} \delta_k z_j \quad (26)$$

## F. Pemrograman Backpropagation Dengan Matlab

### 1. Membentuk Jaringan

#### a. Inisialisasi Jaringan

Langkah pertama yang harus dilakukan untuk memprogram *backpropagation* dengan Matlab adalah membuat inisialisasi jaringan. Perintah yang dipakai untuk membentuk jaringan adalah `newff` yang formatnya adalah sebagai berikut :

```
net = newff(PR,[S1 S2...SN],{TF1 TF2...TFN},BTF,BLF,PF)
```

Dengan

`net` = jaringan Backpropagation yang terdiri dari `n` lapisan

`PR` = matriks ordo `Rx2` yang berisi nilai minimum dan maksimum `R` buah elemen masukannya

`Si` (`i=1,2,...,n`) = jumlah unit pada layar ke-`i` (`i=1,2,...,n`)

`Tfi` (`i=1,2,...,n`) = fungsi aktivasi yang dipakai pada layar ke-`i` (`i=1,2,...,n`).

Default = `tansig` (sigmoid bipolar)

`BTF` = fungsi pelatihan jaringan. Default = `traingdx`

`BLF` = fungsi perubahan bobot/bias. Default = `learngdm`

`PF` = fungsi perhitungan error. Default = `mse`

Beberapa fungsi aktivasi yang dipakai Matlab dalam pelatihan *backpropagation* adalah :

a. tansig (sigmoid bipolar)  $f(\text{net}) = \frac{2}{1+e^{-\text{net}}} - 1$ . Fungsi ini adalah

*default* yang dipakai. Fungsi sigmoid bipolar memiliki *range* [-1,1]

b. logsig (sigmoid biner)  $f(\text{net}) = \frac{1}{1+e^{-\text{net}}}$ . Fungsi sigmoid biner

memiliki bentuk serupa dengan sigmoid bipolar, hanya rangenya adalah [0,1]

c. purelin (fungsi identitas)  $f(\text{net}) = \text{net}$

Pelatihan yang dilakukan dalam Matlab dapat menggunakan berbagai fungsi, tujuannya adalah mempercepat pelatihan. Fungsi *default* yang dipakai oleh Matlab adalah *traingdx*. Dalam fungsi ini, perubahan bobot dilakukan dengan menambahkan momentum. Perubahan dilakukan dengan memperhatikan perubahan bobot pada iterasi sebelumnya. Disamping itu laju pemahaman (*learning rate* =  $\alpha$ ) bukan merupakan konstanta yang tetap, tetapi dapat berubah-ubah selama iterasi.

Umumnya, pelatihan *backpropagation* dalam Matlab dilakukan secara berkelompok (*batch training*). Semua pola dimasukkan dulu, baru kemudian bobot diubah. Dalam pelatihan berkelompok, semua data masukan harus diletakkan dalam sebuah matriks.

#### 1. Inisialisasi Bobot



Setiap kali membentuk jaringan *backpropagation*, Matlab akan memberi nilai bobot dan bias awal dengan bilangan acak kecil. Bobot dan bias ini akan berubah setiap kali dibentuk jaringan. Akan tetapi jika diinginkan memberi bobot tertentu, dapat dilakukan dengan memberi nilai pada `net.IW`, `net.LW` dan `net.b`.

Perhatikan perbedaan antara `net.IW` dan `net.LW`. `net.IW{j,i}` digunakan sebagai variabel untuk menyimpan bobot dari unit masukan layar *i* ke unit tersembunyi (atau unit keluaran) layar *j*. Karena dalam *backpropagation*, unit masukan hanya terhubung dengan layar tersembunyi paling bawah, maka bobotnya disimpan dalam `net.IW{1,1}`.

Sebaliknya, `net.LW{k,j}` dipakai untuk menyimpan bobot dari unit di layar tersembunyi ke-*j* ke unit di lapisan tersembunyi ke-*k*. Sebagai contoh, `net.LW{2,1}` adalah penyimpan bobot dari layar tersembunyi paling bawah (lapisan tersembunyi ke-1) ke layar tersembunyi di atasnya (lapisan tersembunyi ke-2).

## 2. Simulasi Jaringan

Perintah `sim` digunakan pada *backpropagation* untuk menghitung keluaran jaringan berdasarkan arsitektur, pola masukan dan fungsi aktivasi yang dipakai.

`[Y,Pf,Af,E,perf] = sim(net,P,Pi,Ai,T)`

Dengan parameter masukan

net : nama jaringan dalam perintah newff

P : vektor masukan jaringan

Pi : kondisi delay awal masukan. Default = zeros

Ai : kondisi delay layar. Default = zeros

T : vektor target jaringan. Default = zeros

Dan parameter hasil

Y : keluaran jaringan

Pf : kondisi akhir delay masukan

Af : kondisi akhir delay layar

E : error jaringan = T-Y

Perf: unjuk kerja jaringan

Pi, Ai, Pf, Af hanya dipakai bagi jaringan yang memiliki *delay* masukan dan layar. Untuk sekedar menghitung keluaran jaringan, dapat dipakai statemen sederhana :

y = sim (net,p);

Perhatikan bahwa untuk menghitung keluaran jaringan, tidak perlu diketahui targetnya. Akan tetapi jika ingin dihitung *error* yang terjadi (selisih antara target dengan keluaran jaringan), maka harus diketahui targetnya.

## G. Konsep Pemrosesan Citra

Tujuan utama dari pemrosesan citra adalah memperbaiki kualitas suatu gambar agar mudah diinterpretasikan oleh mata manusia dan

untuk mengolah informasi pada gambar untuk pengenalan objek secara otomatis oleh suatu mesin. Dalam pengenalan wajah dan pengenalan pola wajah diproses dicapture sampai pada ekstraksi cirinya. Pada pengenalan wajah proses capture ini sangat menentukan tingkat kesulitan dalam komputasinya, salah satunya bahwa dalam setiap proses capture ternyata cahaya, warna, posisi, jarak dan kemiringan menjadi suatu masalah yang diperhatikan. (Alan Budi (2007) Pengolahan Citra

#### 1. Elemen Dasar Suatu Citra

##### a. Pixel ( picture element)

Gambar yang bertipe bitmap tersusun dari pixel-pixel. pixel disebut juga dengan dot. Ukuran relative kecil pembentuk gambar bitmap.

Banyaknya pixel tiap satuan luas tergantung pada bit depth yang dipakai. Semakin banyak jumlah piksel tiap satuan luas, semakin baik kualitas gambar yang dihasilkan dan tentu semakin besar ukuran filenya.

##### b. Bit depth

Bit depth (kedalam warna)/ *pixel depth/ color depth*. Bit depth menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan/dicetak dalam setiap pixel, semakin besar nilainya semakin bagus kualitas gambar yang dihasilkan, tentu ukuran juga semakin besar.

c. Resolusi

Resolusi adalah jumlah pixel persatuan luas yang ada disuatu gambar. Satuan pixel sering dipakai adalah dpi (*dot per inch*)/ppi (*pixel per inch*). Satuan dpi menentukan jumlah pixel yang ada setiap satu satuan luas. Yang dalam hal ini adalah satu inch kuadrat. Resolusi sangat berpengaruh pada detil dan perhitungan gambar.

d. Kecerahan dan Kontras

Kecerahan (*brightness*) adalah intensitas yang terjadi pada suatu titik citra. Sistem visual manusia mampu menyesuaikan diri dengan tingkat kecerahan dengan jangkauan.

2. Operasi Pengolahan Citra

Pengambilan citra wajah di klasifikasikan secara umum yaitu :

a. Peningkatan kualitas citra (*image enhancement*)

Tujuannya untuk memperbaiki kualitas citra dibandingkan dengan citra semula. Contoh operasi perbaikan citra :

1. Perbaikan kontras gelap/terang
2. Perbaikan tepian objek (*edge enhacement*)
3. Penajaman (*sharpening*)
4. Penapisan derau (*noise filtering*)

b. Perbaikan citra (*imge restoration*)

Bertujuan menghilangkan cacat pada citra, bertujuan pemugaran citra hamper sama dengan operasi perbaikan citra. yaitu :

1. Penghilang kesamaran (*debluring*)
2. Penghilang derau (*noise*)
3. Pelembutan citra (*smoothing*)

c. Segmentasi Citra (*image analysis*)

Pengenalan pola yang sesuai kriteria dari segmen yang di operasikan .

d. Analisis citra (*image analysis*)

Bertujuan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. dengan cara :

1. Pendeteksi Objek (*egde detection*)
2. Ekstraksi batas (*boundary*)
3. Representasi area wilayah (*region*)

e. Rekonstruksi citra (*image reconstruction*)

Bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. ( 8)

## **BAB III**

### **METODE PENELITIAN**

#### **A. Tempat dan Waktu Penelitian**

Berdasarkan tabulasi dari seluruh rangkaian kegiatan yang akan difokuskan pada riset program Magister penelitian dilaksanakan di bulan februari sampai bulan juni 2013. Seluruh rangkaian kegiatan riset akan dilakukan pada lab Teknik Telekomunikasi Radio dan Gelombang Pendek, Jurusan Teknik Elektro, Universitas Hasanuddin, Makassar , lab dasar pemograman, Jurusan Teknik Elektro, Politeknik Negeri Manado serta pada sejumlah lokasi tempat tinggal.

#### **B. Jenis Penelitian**

Penelitian ini termasuk penelitian software untuk membangun sistem dan Hardware yaitu aplikasi draiver inverter sinayal yang di keluarkan untuk membuktikan pada saat wajah dikenal dan wajah tidak dikenal.untuk mengoptimalkan sistem sekuritas rumah moderen berbasis jaringan saraf tiruan ( JST).

#### **C. Rancangan Penelitian**

##### **1. Perangkat Keras Sistem**

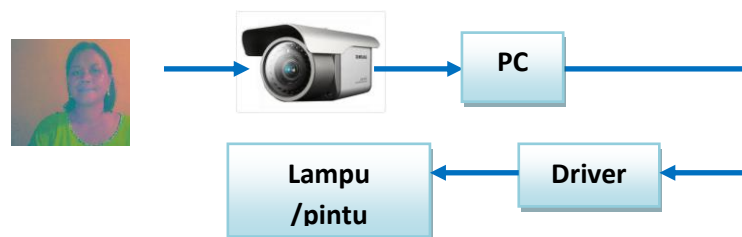
Perangkat keras yang digunakan pada proses simulasi yaitu sebuah komputer dengan spesifikasi sebagai berikut :

- a. ACER aspire 4738z dengan Processor P6200, 320 GB HDD
- b. Webcam ACER 4738z (TECGO)
- c. Sistem operasi : Windows XP
- d. Memori : 2 GB DDR3
- e. Hard Disk : 320 GB HDD

## 2. Perangkat Lunak Sistem

- a. MATLAB version R2009a
- b. Microsoft office Word 2007

Perambatan galat mundur (Backpropagation) adalah sebuah metode sistematis untuk pelatihan jaringan saraf tiruan. Metode ini memiliki dasar sistematis yang kuat obyektif dan algoritma untuk mendapat bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat error melalui model yang dikembangkan (training set)



**Gambar 13.** Konfigurasi Sistem Sekuritas Rumah Modern Berbasis JST.

Aplikasi pengontrolan yang dikembangkan dalam penulisan ini adalah Teknik Akuisisi Data Pada Akses Masuk Rumah Modern. Dimana prinsip kerja dari sistem yaitu :

1. Kamera menangkap/ mengambil citra wajah dari orang yang terdeteksi oleh kamera tersebut dikirim ke PC.
2. PC akan mengolah data citra menggunakan program matlab dengan metoda backpropagation.
3. Data hasil pengolahan citra tersebut dibandingkan dengan data citra yang telah tersimpan sebagai data set point.
4. Jika data citra tersebut tidak sama dengan data set point, maka pintu tidak akan terbuka karena pintu tidak aktif ( data yang dikirim dari computer ke driver berlogika 0)
5. Jika data citra sama dengan data set point, pintu akan terbuka, karena yang dikirim dari computer ke driver berlogika 1, Driver akan aktif dan menghidupkan motor untuk membuka pintu.

#### **D. Rancangan Sistem**

Sistem yang akan dirancang adalah sebuah sistem yang dapat mengenal bentuk wajah dan jenis kelamin laki-laki dan perempuan. Sistem ini dapat mengenal ke 7 citra wajah manusia dengan 5 posisi wajah yang di data set pointkan dan mengenal ke 7 citra wajah yang tidak dimasukan pada data set point. Secara umum, perancangan tampilan dan aplikasi matlab pada gambar 14.





**Gambar 14.** Tampilan aplikasi matlab

## 1. Inisialisasi wajah (data set point)

### a) Start

Bagian start berfungsi untuk mengaktifkan keseluruhan program sebagai berikut:

```
global vid
vid = videoinput('winvideo');
start(vid);
%vidRes = get(vid, 'VideoResolution');
%imWidth = vidRes(1);
%imHeight = vidRes(2);
%nBands = get(vid, 'NumberOfBands');
%hImage = image( zeros(imHeight, imWidth, nBands)
);
preview(vid);%(vid, hImage);
%axes(handles.axes1); imshow(vid);
```

### b) Capture

Fungsinya untuk pengambilan gambar, dimana codingnya sebagai berikut :

Executes on button press in CAPTURE1.

```
function CAPTURE1_Callback(hObject, eventdata, handles)
```

```
    hObject    handle to CAPTURE1 (see GCBO)
```

```
    eventdata reserved - to be defined in a future version of MATLAB
```

```
    handles    structure with handles and user data (see GUIDATA)
```

### c) Save

Fungsinya yaitu untuk menyimpan semua program pada folder, dengan perintah programnya yaitu :

```
% --- Executes on button press in SAVE.
```

```
function SAVE_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to SAVE (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

## 2. Input wajah

Fungsinya yaitu mengambil gambar dan diproses pada histogram citra selanjutnya siap di uji apakah gambar sesuai dengan data yang di set point atau disimpan atau tidak, ke image netural network.

## 3. Image ntural network

Fungsinya yaitu untuk memprosesan saraf tiruan

a. Inpu image

Fungsinya yaitu memanggil m-file yang berisikan semua data gambar yang di simpan dan membandingkan dengan data gambar pada langkah 2 .

b. Image Procecing

Fungsinya mengkonversi citra wajah ke dalam bentuk grayscale,dalam citra warna menjadi hitam putih.dengan perintah programnya sebagai berikut :

```
% --- Executes on button press in IMANGE_PROCC.
function IMANGE_PROCC_Callback(hObject,
    eventdata, handles)
% hObject handle to IMANGE_PROCC (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)
global rgb1 data2 BW1

I1 = rgb2gray(rgb1);
BW1 = im2bw(I1);%(I2,graythresh(I2));

axes(handles.axes1);imshow(BW1);
data2=bwarea(BW1);

%=====set(handles.edit8,'string',data2);

%I = rgb2gray(rgb);
%BW = im2bw(I);%(I2,graythresh(I2));
%axes(handles.axes1);imshow(BW);
%data1=bwarea(BW)
%set(handles.edit7,'string',data1);
```

c. Training

Fungsinya yaitu untuk mendapatkan hasil pengujian data yang sudah di inputkan pada input training jaringan saraf tiruan ,dengan perintah programnya yaitu :

```
% --- Executes on button press in TRAINING.
function TRAINING_Callback(hObject, eventdata,
handles)
```

d. Testing

Fungsinya yaitu untuk mengenali gambar disetiap user, apakah terdeteksi/open,dan tidak terdeteksi close,perintah programnya yaitu :

```
% --- Executes on button press in TESTING.
function TESTING_Callback(hObject, eventdata,
handles)
% hObject handle to TESTING (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
```

4. Input training

Fungsinya mengisi input training yaitu variable maksimum epoch adalah 1500, target error adalah  $1e-3$ , learning adalah 0,1 dan empoh show 200, untuk proses jaringan saraf tiruan.

a. Maksimum epoch

Fungsinya yaitu untuk mengetahui Jumlah iterasi perintah program maksimum sebanyak 1500 terasi

b. Target eror

Fungsinya yaitu untuk mengetahui jumlah kesalahan dari pencapaian program

c. Learning rate

Fungsinya yaitu untuk mengetahui proses pembelajaran program

d. Epoch show

Fungsinya yaitu untuk mengetahui jumlah iterasi perintah saat program mengenal wajah dan tidak mengenal wajah.

### **E. Pelatihan Backpropagation**

Matlab menyediakan berbagai variasi pelatihan *backpropagation*. Dalam sub bab ini akan dibahas pelatihan standar yang digunakan untuk melatih jaringan.

Pelatihan *backpropagation* menggunakan metode pencarian titik minimum untuk mencari bobot dengan *error* minimum. Dalam proses pencarian ini dikenal dua macam metode yaitu metode *incremental* dan metode kelompok (*batch*).

Dalam metode *incremental*, bobot diubah setiap kali pola masukan diberikan ke jaringan. Sebaliknya, dalam metode kelompok, bobot diubah setelah semua pola masukan diberikan ke jaringan. *Error* (suku perubahan bobot) yang terjadi dalam setiap pola masukan dijumlahkan untuk menghasilkan bobot baru. Matlab menggunakan metode pelatihan kelompok dalam iterasinya. Perubahan bobot dilakukan per *epoch*.

Untuk melatih jaringan digunakan perintah `train` yang formatnya adalah sebagai berikut:

```
[net,tr,Y,E,Pf,Af] = train(net,P,T,Pi,Ai,VV,TV)
```

Dengan

`net` : jaringan yang didefinisikan dalam `newff`

`P` : masukan jaringan

`T` : target jaringan. Default = zeros

`Pi` : kondisi delay awal masukan. Default = zeros

`Ai` : kondisi delay awal layar. Default = zeros

`VV` : struktur validasi vektor. Default = []

`TV` : struktur vektor uji. Default = []

Perintah `train` akan menghasilkan

`net` : jaringan yang baru

`tr` : record pelatihan (epoch dan performa)

`Y` : keluaran jaringan

`E` : error jaringan

`Pf` : kondisi akhir delay masukan

`Af` : kondisi akhir delay lapisan

Metode paling sederhana untuk merubah bobot adalah metode penurunan gradien (*gradient descent*). Bobot dan bias diubah pada arah dimana unjuk kerja fungsi menurun paling cepat, yaitu dalam arah negatif gradiennya.

Jika  $w_k$  adalah vektor bobot pada iterasi ke- $k$ ,  $g_k$  adalah gradien dan  $\alpha_k$  adalah laju pemahaman, maka metode penurunan gradien memodifikasi bobot dan bias menurut persamaan  $w_{k+1} = w_k - \alpha_k g_k$ .

Matlab menyediakan beberapa metode pencarian titik minimumnya. Pencarian titik minimum dengan metode penurunan gradien dilakukan dengan memberikan parameter 'traingd' dalam parameter setelah fungsi aktivasi pada perintah `newff`.

Ada beberapa parameter pelatihan dapat diatur sebelum pelatihan dilakukan. Dengan memberi nilai yang diinginkan pada parameter-parameter tersebut dapat diperoleh hasil yang lebih optimal.

#### 1. Jumlah *epoch* yang akan ditunjukkan kemajuannya.

Menunjukkan berapa jumlah *epoch* berselang yang akan ditunjukkan kemajuannya.

Instruksi : `net.trainParam.show = EpochShow`

Nilai *default* untuk jumlah *epoch* yang akan ditunjukkan adalah 25.

#### 2. Maksimum *epoch*

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*.

Instruksi : `net.trainParam.epochs = MaxEpoch`

Nilai *default* untuk maksimum epoch adalah 10.

### 3. Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Instruksi : `net.trainParam.goal = TargetError`

Nilai *default* untuk kinerja tujuan adalah 0.

### 4. Learning rate

*Learning rate* adalah laju pembelajaran. Semakin besar nilai *learning rate* akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika *learning rate* diset terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya, jika *learning rate* diset terlalu kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama.

Instruksi : `net.trainParam.lr = LearningRate`

Nilai *default* untuk *learning rate* adalah 0,01.

### 5. Waktu maksimum untuk pelatihan

Menunjukkan waktu maksimum yang diijinkan untuk melakukan pelatihan. Iterasi akan dihentikan apabila waktu pelatihan melebihi waktu maksimum.

Instruksi : `net.trainParam.time = MaxTime`

Nilai *default* untuk waktu maksimum adalah tak terbatas (inf).



## **F. Teknik Meningkatkan Kecepatan Proses Imeg Wajah pada Sistem JST Backpropagation.**

Metode standar *backpropagation* seringkali terlalu lambat untuk keperluan praktis. Beberapa modifikasi dilakukan terhadap standar *backpropagation* dengan cara mengganti fungsi pelatihannya.

Secara umum, modifikasi dapat dikelompokkan dalam dua kategori. Kategori pertama adalah metode yang menggunakan teknik heuristik yang dikembangkan dari metode penurunan tercepat yang dipakai dalam standar *backpropagation*. Kategori kedua adalah menggunakan metode optimisasi numerik selain penurunan tercepat. Beberapa metode yang dipakai sebagai modifikasi adalah metode gradien conjugate, quasi Newton, dan lain-lain. Dalam sub bab berikut ini dibicarakan dahulu tentang beberapa modifikasi yang masuk dalam kategori pertama (*backpropagation* dengan momentum, variabel laju pemahaman, dan *backpropagation resilient*). Berikutnya barulah dibahas tentang beberapa metode yang masuk dalam kategori kedua.

### **1. Metode Penurunan Gradien dengan Momentum (traingdm)**

Meskipun metodenya paling sederhana, tapi metode penurunan gradien sangat lambat dalam kecepatan proses iterasinya. Ini terjadi karena kadang-kadang arah penurunan tercepat bukanlah arah yang tepat untuk mencapai titik minimum globalnya.

Modifikasi metode penurunan tercepat dilakukan dengan menambahkan momentum. Dengan momentum, perubahan bobot tidak

hanya didasarkan atas *error* yang terjadi pada *epoch* pada waktu itu. Perubahan bobot saat ini dilakukan dengan memperhitungkan juga perubahan bobot pada *epoch* sebelumnya. Dengan demikian kemungkinan terperangkap ke titik minimum lokal dapat dihindari.

Besarnya efek perubahan bobot terdahulu (disebut faktor momentum) bisa diatur dengan suatu bilangan antara 0 dan 1. Faktor momentum = 0 berarti perubahan bobot hanya dilakukan berdasarkan *error* saat ini (penurunan gradien murni). Dalam Matlab, pelatihan *backpropagation* dengan menggunakan metode penurunan gradien dengan momentum dilakukan dengan mendefinisikan fungsi pelatihan 'traingdm' dalam pembentukan jaringannya. Besarnya faktor momentum dilakukan dengan memberi nilai antara 0 – 1 pada `net.trainParam.mc` (*default* = 0,9). Parameter lain yang dapat diatur dalam `traingdm` sama dengan `traingd`.

## 2. Variabel Laju Pemahaman (`traingda`, `traingdx`)

Dalam standar *backpropagation*, laju pemahaman berupa suatu konstanta yang nilainya tetap selama iterasi. Akibatnya, unjuk kerja algoritma sangat dipengaruhi oleh besarnya laju pemahaman yang dipakai. Secara praktis, sulit untuk menentukan besarnya laju pemahaman yang paling optimal sebelum pelatihan dilakukan. Laju pemahaman yang terlalu besar maupun terlalu kecil akan menyebabkan pelatihan menjadi lambat.

Pelatihan akan lebih cepat apabila laju pemahaman dapat diubah-ubah besarnya selama proses pelatihan. Jika *error* sekarang lebih besar dibandingkan *error* sebelumnya, maka laju pemahaman diturunkan. Jika sebaliknya, maka laju pemahaman diperbesar. Dengan demikian laju pemahaman dapat dibuat sebesar besarnya dengan tetap mempertahankan kestabilan proses.

Dalam Matlab, penggunaan variabel laju pemahaman dilakukan dengan menggunakan 'traingda' pada parameter fungsi pelatihan *newff*.

Penggunaan laju pemahaman juga bisa dikombinasikan dengan menambahkan faktor momentum. Fungsi pelatihan yang dipakai di Matlab adalah 'traingdx'. Fungsi pelatihan ini memiliki kecepatan pelatihan yang tinggi sehingga dipakai sebagai *default* dalam pelatihan *backpropagation* di Matlab.

### 3. Resilient Backpropagation (trainrp)

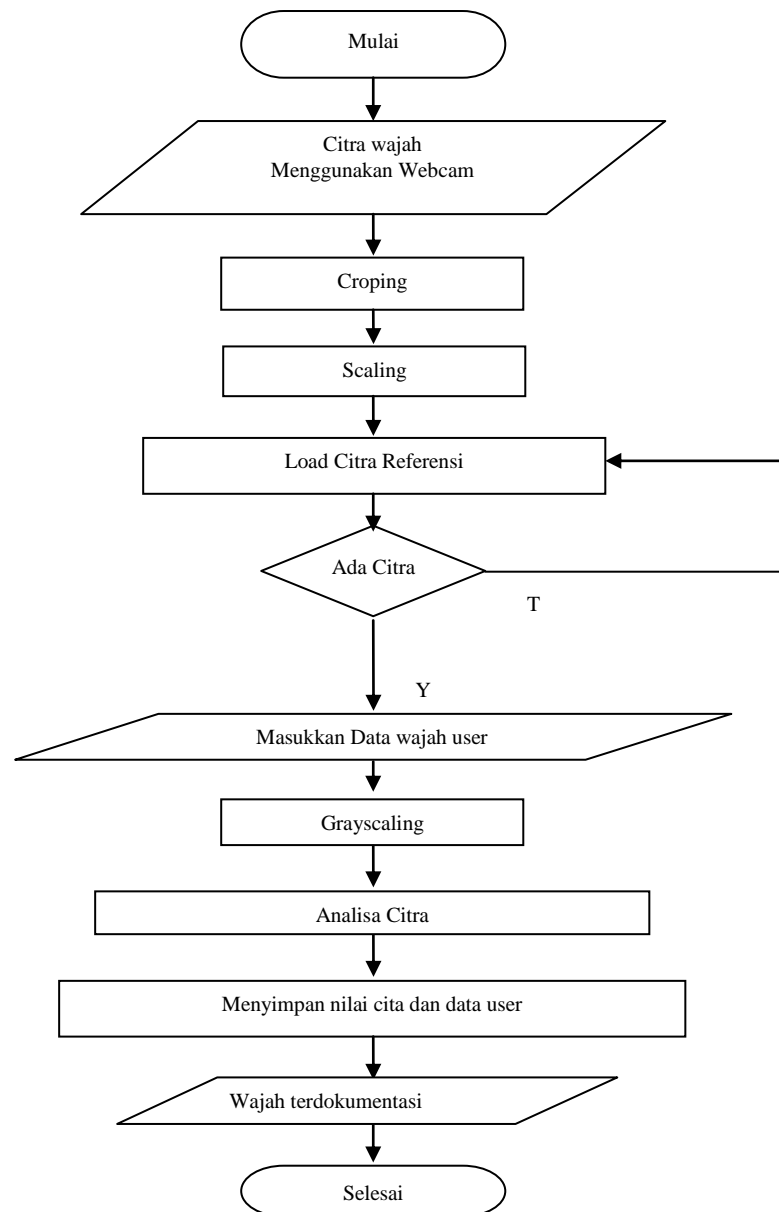
Jaringan *backpropagation* umumnya menggunakan fungsi aktivasi sigmoid. Fungsi sigmoid akan menerima masukan dari range tak berhingga menjadi keluaran pada *range* [0,1]. Semakin jauh titik dari  $x = 0$ , semakin kecil gradiennya. Pada titik yang cukup jauh dari  $x = 0$ , gradiennya mendekati 0. Hal ini menimbulkan masalah pada waktu menggunakan metode penurunan tercepat (yang iterasinya didasarkan atas gradien). Gradien yang kecil menyebabkan perubahan bobot juga kecil, meskipun masih jauh dari titik optimal.

Masalah ini diatasi dalam *resilient backpropagation* dengan cara membagi arah dan perubahan bobot menjadi dua bagian yang berbeda. Ketika menggunakan penurunan tercepat, yang diambil hanya arahnya saja. Besarnya perubahan bobot dilakukan dengan cara lain.

Dalam Matlab *resilient backpropagation* dilakukan dengan menuliskan 'trainrp' pada fungsi pelatihannya.

#### 4. Algoritma Gradien Conjugate (traincgf, traincgp, traincgb)

Dalam standar *backpropagation*, bobot dimodifikasi pada arah penurunan tercepat. Meskipun penurunan fungsi berjalan cepat, tapi tidak menjamin akan konvergen dengan cepat. Dalam algoritma gradien konjugate, pencarian dilakukan sepanjang arah conjugate. Dalam banyak kasus, pencarian ini lebih cepat. Ada berbagai metode pencarian yang dilakukan berdasarkan prinsip gradien conjugate, antara lain Fletcher-Reeves ('traincgf'), Polak-Ribiere ('traincgp'), Powel Beale ('traincgb'). Pengambilan citra wajah pada gambar.



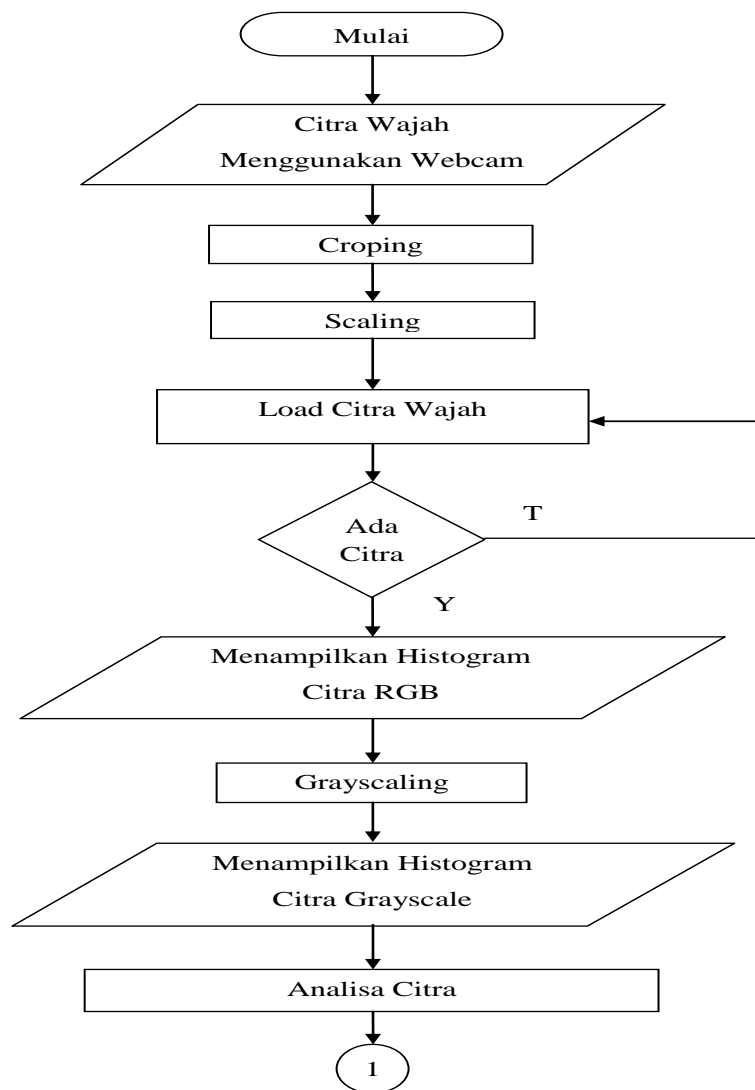
**Gambar 15.** Diagram alir pengenalan citra wajah

Langkah-langkah pengambilan citra wajah yaitu :

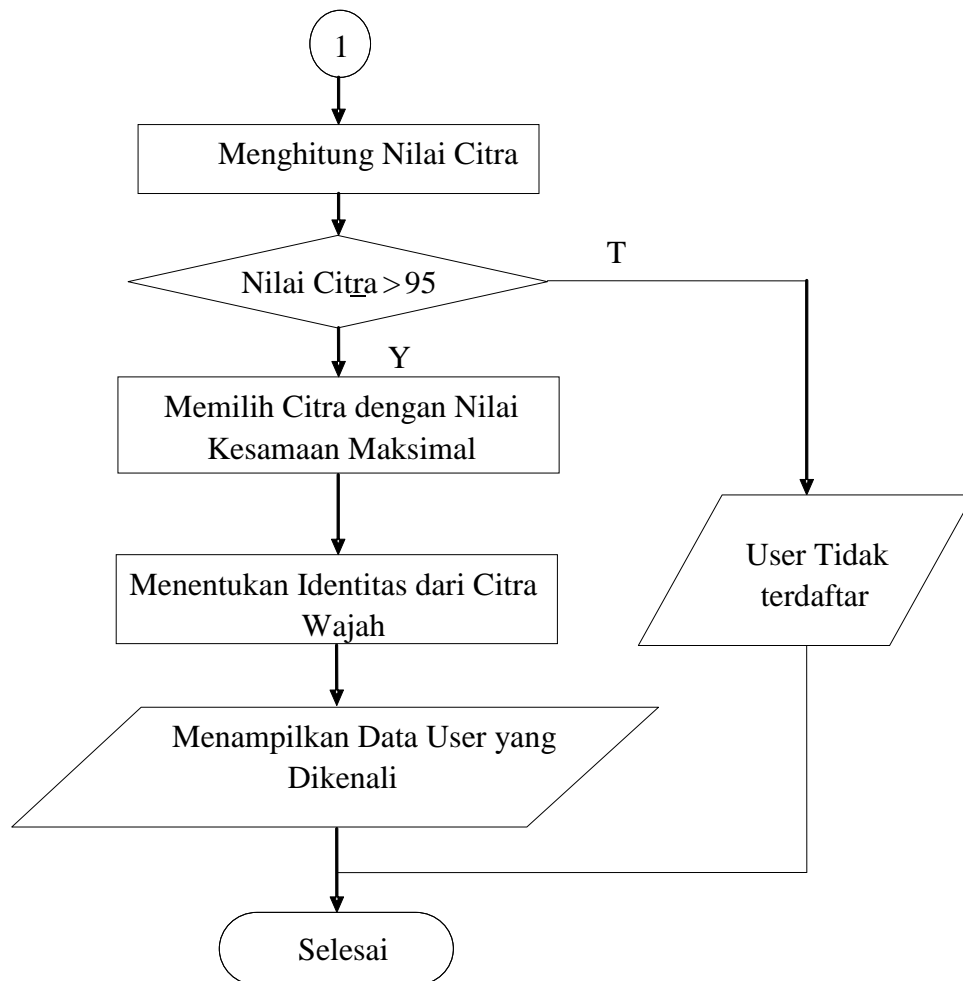
1. Masukan berupa citra wajah menggunakan wabcam
2. Selanjutnya citra wajah diproses dengan melakukan cropping dan scaling
3. Proses selanjutnya dilakukan load citra referensi

Jika citra tidak tersedia maka kembali kelangka 3 dan jika citra tersedia maka masukan data wajah user yang tersimpan pada m-file

4. Proses grayscaleing di tampilkan pada histogram citra kemudian dianalisa citra,selanjutnya data user dan nilai ciri disimpan dan didokumentasikan.
5. Selesai



**Gambar 16.** Diagram alir pengidentifikasian wajah



**Gambar 17.** Diagram alir pengidentifikasian wajah (lanjutan)

Langkah-langkah pengidentifikasi wajah yaitu :

1. Masukan berupa citra wajah menggunakan wabcam
2. Selanjutnya citra wajah diproses dengan melakukan cropping dan scaling
3. Proses selanjutnya dilakukan load citra wajah ,Jika ada citra maka akan tampil pada histogram cita RGB foto wajah warna, jika tidak ada citra maka kembali mengload citra wajah.
4. Proses selanjutnya grayscale dari tampilan citra wajah,

5. Proses analisa citra apakah dikenal program atau tidak
6. Proses menghitung nilai citra yang tentukan
7. Jika nilai citra lebih besar 95 maka program memilih citra dengan nilai kesamaan maksimal dan jika tidak maka user tidak terdaftar proses selesai
8. Selanjutnya Program dapat menentukan identitas dari citra wajah
9. Program menampilkan data user yang dikenali atau terdeteksi proses selesai

## **G. Proses Awal Sistem**

### **1. Data Input**

Proses mengimput data dilakukan dengan cara yaitu satu persatu dari delapan wajah yang berbeda dengan kondisi pengambilan gambar pada webcam TECGO yang sudah diatur pencahayaan konstan, webcam ini dapat di atur intensitas cahayanya pada scroll yaitu terang , sedang dan gelap. Melalui format file gambar diambil tampak depan wajah manusia dengan 5 posisi wajah yang berbeda secara langsung dan benar-benar akan terdeteksi. Dengan perintah programnya yaitu

```
% --- Executes on button press in START.
function START_Callback(hObject, eventdata, handles)
% hObject    handle to START (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of START
global a
a=0;
```



## 2. Proses Save ke Folder

Setelah tahap diatas dilalui akan menghasilkan citra biner yang berukuran 160 x 120 pixel secara otomatis akan disimpan pada folder.

Syntax program yang digunakan adalah :

```
% --- Executes on button press in SAVE.
function SAVE_Callback(hObject, eventdata, handles)
% hObject    handle to SAVE (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global rgb D1 D2 D3 D4 D5 D6 D7 D8D9 D10
D11D12D13D15D16D17D18D19D20D21D22D23D24D25D26D2
7D28D29D30D31D32D33D34D35gambar a
b=1;
gambar=a+b;
%set(handles.edit8,'string',gambar);
if (gambar==1)
    a=gambar;
D1 = rgb;
else if (gambar==2)
    a=gambar;
D2 = rgb;
    else if (gambar==3)
    a=gambar;
    D3 = rgb;
    else if (gambar==4)
    a=gambar;
    D4 = rgb;
    else if (gambar==5)
    a=gambar;
    D5 = rgb;
    else if (gambar==6)
    a=gambar;
    D6 = rgb;
    else if (gambar==7)
    a=gambar;
    D7 = rgb;
    else if (gambar==8)
    a=gambar;
    D8 = rgb;
    else if (gambar==9)
```

```
a=gambar;
    D9 = rgb;

    else if (gambar==10)
a=0;
    D10 = rgb;
    gambar=0;
    else if (gambar==11)
a=0;
    D11 = rgb;
    gambar=0;

    else if (gambar==12)
a=0;
    D12 = rgb;
    gambar=0;
    else if (gambar==13)
a=0;
    D13 = rgb;
    gambar=0;
    else if (gambar==14)
a=0;
    D14 = rgb;
    gambar=0;
    else if (gambar==15)
a=0;
    D15 = rgb;
    gambar=0;
    else if (gambar==16)
a=0;
    D16 = rgb;
    gambar=0;
    else if (gambar==17)
a=0;
    D17 = rgb;
    gambar=0;
    else if (gambar==18)
a=0;
    D18 = rgb;
    gambar=0;
    else if (gambar==19)
a=0;
    D19 = rgb;
    gambar=0;
    else if (gambar==20)
a=0;
```

```
        D20 = rgb;
        gambar=0;
        else if (gambar==21)
a=0;
        D21 = rgb;
        gambar=0;
        else if (gambar==22)
a=0;
        D22 = rgb;
        gambar=0;
        else if (gambar==23)
a=0;
        D23 = rgb;
        gambar=0;
        else if (gambar==24)
a=0;
        D24 = rgb;
        gambar=0;
        else if (gambar==25)
a=0;
        D25 = rgb;
        gambar=0;
        else if (gambar==26)
a=0;
        D26 = rgb;
        gambar=0;
        else if (gambar==27)
a=0;
        D28 = rgb;
        gambar=0;
        else if (gambar==28)
a=0;
        D28 = rgb;
        gambar=0;
        else if (gambar==29)
a=0;
        D29 = rgb;
        gambar=0;
        else if (gambar==30)
a=0;
        D30 = rgb;
        gambar=0;
        else if (gambar==31)
a=0;
        D31 = rgb;
        gambar=0;
```



```

end
end
end
end
end
end
end

image1=uiputfile(...
{'*.bmp','file citra (*.bmp)';
'*.bmp','citra bmp (*.bmp)';...
'*. *','Semua File (*.*)'},...
'Menyimpan Citra');
imwrite(rgb,fullfile(image1),'BMP');

```

### 3. Proses Perubahan Citra RGB menjadi Grayscale

Pada tahap pra- proses kedua ini adalah merubah citra RGB hasil dari pengambilan gambar menjadi Grayscale, akan dapat diproses citra tersebut. Syntax program yang digunakan adalah :

```

% --- Executes on button press in IMANGE_PROCC.
function IMANGE_PROCC_Callback(hObject, eventdata,
handles)
% hObject handle to IMANGE_PROCC (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)
global rgb1 data2 BW1

I1 = rgb2gray(rgb1);
BW1 = im2bw(I1);%(I2,graythresh(I2));

axes(handles.axes1);imshow(BW1);
data2=bwarea(BW1);
%=====set(handles.edit8,'string',data2);

%I = rgb2gray(rgb);
%BW = im2bw(I);%(I2,graythresh(I2));
%axes(handles.axes1);imshow(BW);
%data1=bwarea(BW)
%set(handles.edit7,'string',data1);

```

#### 4. Proses Pendeteksian

Proses ini mendeteksi citra wajah dengan membandingkan data setpoint dengan citra yang tidak ada pada data set point.dengan programnya yaitu:

```

% --- Executes on button press in TRAINING.
function TRAINING_Callback(hObject, eventdata, handles)
% hObject    handle to TRAINING (see GCBO)
% eventdata  reserved - to be defined in future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Maxepoh TError Lr EpochSow D1 D2 D3 D4 D5 D6 D7
D8D9 D10
D11D12D13D15D16D17D18D19D20D21D22D23D24D25D26D2
7D28D29D30D31D32D33D34D35Y Bobotakhir_Bias_Lapisan P
data2
Maxepoh=str2num(get(handles.edit3, 'String'));
TError=str2num(get(handles.edit4, 'String'));
Lr=str2num(get(handles.edit5, 'String'));
EpochSow=str2num(get(handles.edit6, 'String'));
global Maxepoh TError Lr EpochSow D1 D2 D3 D4 D5 D6 D7 D8
D9 D10
P= [D1 D2 D3 D4 D5; D6 D7 D8 D9 D10; D11 D12 D13 D14
D15; D16 D17 D18
D19 D20; D21 D22 D23 D24 D25; D26 D27 D28 D29 D30; D31
D32 D33 D34 D35];
%[1 2 3 4 5;2 4 6 8 10]; %[0 1 2 1 10 12 -5 -8 -10 -15; 0 1 -1 6 3
-1 -2 2 -5 2];
T=[D1 D2 D3 D4 D5 ]; % 15916 16048 15852 16328 16361]; %0
0 1 1 2 2 -1 -1 -2 -
2];

%membangun jaringan saraf feedforward
net = newff(minmax(P),[5 1],{'tansig' 'purelin'},'trainrp');

%melihat bobotawal input, lapisan dan bias
Bobotawal_input      = net.IW{1,1}
Bobotawal_Bias_input = net.b{1,1}
Bobotawal_Lapisan    = net.LW{2,1}
Bobotawal_Bias_lapisan = net.b{2,1}

```

```

%set max epoch , goal, learning rate, show step
net.trainParam.epochs = Maxepoch;%1500;
net.trainParam.goal    = TError; %1e-3;
net.trainParam.lr      = Lr; %0.1;
net.trainParam.show    = EpochSow; %100;

```

```

%melakukan pembelajaran
net = train(net,P,T);

```

```

%melihat bobot2 awal input, lapisan dan , bias
Bobotakhir_input      = net.IW{1,1}
Bobotakhir_Bias_Input = net.b{1,1}
Bobotakhir_Lapisan    = net.LW{2,1}
Bobotakhir_Bias_Lapisan = net.b{2,1}

```

## 5. Proses testing

Fungsi dari tombol ini untuk mengenali gambar setiap user, apakah sesuai atau tidak. Dengan perintah programnya sebagai berikut :

```

% --- Executes on button press in TESTING.
function TESTING_Callback(hObject, eventdata, handles)
% hObject    handle to TESTING (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global D1 D6 data2 %rgb
%axes(handles.axes3); imshow(rgb);
data=data2-D1;
%set(handles.edit4,'string',data);
if(data>300)
    %set(handles.edit2,'string','tdk sama');
    data=data2-D6;
    if(data>300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOUSE');
    else if(data<-300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOUSE');
    else
        set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
        delete(instrfind);
        %clear all;
        s = serial('COM15');

```

```

set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'FlowC
ontrol','none','Terminator','CR/LF');
    fopen(s);
    fprintf(s,'%s','1');
%out = fscanf(s);
    fclose(s);
    delete(s);
    clear s;

    end
end
else if(data<-300)
    %set(handles.edit2,'string','Tidak Terdeteksi/CLOUSE');
    data=data2-D6;
    if(data>300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOUSE');
    else if(data<-300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOUSE');
    else
        set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
        delete(instrfind);
        %clear all;
        s = serial('COM15');

set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'FlowC
ontrol','none','Terminator','CR/LF');
    fopen(s);
    fprintf(s,'%s','1');
%out = fscanf(s);
    fclose(s);
    delete(s);
    clear s;

    end
end

else
    set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
    delete(instrfind);
    %clear all;
    s = serial('COM15');
set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'FlowC
ontrol','none','Terminator','CR/LF');
    fopen(s);

```



```

fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;
    end
end

```

## H. Proses Pembentukan Pola Jaringan Saraf Tiruan Propagasi Balik dan Pelatihan

1. Pembentukan pola jaringan syaraf tiruan propagasi balik,data input akan digunakan adalah saat pengambilan gambar secara langsung. Data input yang dipergunakan melalui tahapan pra-proses terlebih dahulu agar dapat menemukan ciri-ciri dari setiap citra.adapun programnya yaitu :

```

net = newff(minmax(P),[5 1],{'tansig' 'purelin'},'trainrp');

%melihat bobotawal input, lapidsan dan bias
Bobotawal_input      = net.IW{1,1}
Bobotawal_Bias_input = net.b{1,1}
Bobotawal_Lapisan    = net.LW{2,1}
Bobotawal_Bias_lapisan = net.b{2,1}

```

2. Mengubah ukuran matriks X yang berukuran M x N ,menjadi persamaan  $y = M \times N$  ,menjadi 160 x120 pixel = 19200 pixel dari setiap pengambilan citra wajah yaitu 7 orang dan 5 posisi wajah keseluruhan 35 gambar wajah.
3. Sebelum melakukan pelatihan,ditetapkan terlebih dahulu parameter yang akan digunakan. Dengan parameter set maksimum epoch (=

1500), kinerja tujuan  $1e-3$ , learning rate 0,1. Dengan programnya sebagai berikut :

```
net.trainParam.epochs = Maxepoch;%1500;
net.trainParam.goal    = TError; %1e-3;
net.trainParam.lr      = Lr; %0.1;
net.trainParam.show    = EpochShow; %100;
```

4. Perhitungan gradient dan perbaikan nilai bobot dilakukan setelah pengoperasian semua input data. Proses pelatihan yang saya lakukan dengan menggunakan program sebagai berikut :

```
net = train(net,P,T);
```

### **I. Pengujian sistem melalui lampu indikator**

Pada proses pengujian ini digunakan 35 citra wajah yang telah dilatih setiap wajah diuji agar dapat diketahui apakah benar-benar wajah terdeteksi atau tidak terdeteksi. Proses pengujian citra wajah dilakukan dengan pengambilan gambar langsung menggunakan webcam di proses melalui program matlab, keluaran dari PC menghasilkan sinyal pada indicator, lampu menyala saat program mengenal wajah pintu terbuka dan lampu akan mati ketika tidak mengenal wajah pintu akan tertutup.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pengujian dan analisa pada intinya adalah untuk mengetahui tingkat keberhasilan suatu sistem dengan metode yang sudah ditentukan. Masing masing metode mempunyai kelemahan dan kelebihan sendiri-sendiri. Sistem identifikasi wajah ini terdiri dari 2 sistem yakni sistem pendeteksi wajah dan sistem pengenalan wajah. Masing-masing sistem diuji dengan cara yang berbeda karena menggunakan metode yang berbeda pula.

#### **A. Pendeteksi Wajah**

Pada awal sistem sebelum pre-processing pada sistem pengenalan wajah ada sebuah sistem lain yakni sistem pendeteksi wajah dengan metode pattern matching atau pencocokan pola. Pendeteksi wajah ini yang membedakan bagian wajah dan bagian bukan wajah pada citra digital sehingga mempercepat proses pengenalan wajah, karena tidak perlu memperhitungkan komponen citra digital yang bukan wajah. Proses ini diawali dengan mengisi input wajah pembanding sebanyak 5 citra wajah dengan cara menekan *capture* inisial wajah.

#### **B. Proses Inisial dan input Wajah**

Masuk ke sistem pengenalan wajah, metode yang digunakan adalah

metode jaringan saraf tiruan atau neural network. Karena perbedaan metode ini maka uji coba yang dilakukan juga berbeda-beda. Pada proses pengujian sistem ini akan diuji sejauh mana kehandalan proses pengenalan wajah menggunakan neural network yang parameternya telah diujicoba, terlebih dahulu.






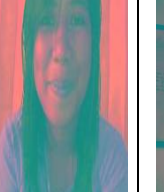


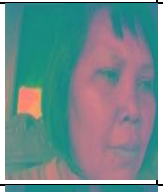



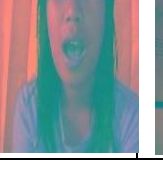

**Gambar 18.** Aplikasi pengenalan wajah

Pada gambar 18, pengenalan wajah akan tampil ketika ada penekanan pada tombol start yang berfungsi untuk memulai proses pengambilan citra, video webcam berfungsi untuk mengambil citra wajah didepannya dan aplikasi citra yang terdiri dari beberapa komponen untuk memproses data.

Langkah selanjutnya pengambilan data citra wajah dengan nama-nama inisial citra wajah masing-masing yaitu eta (citra wajah 1), saul (citra wajah 2), lino (citra wajah 3), hoa (citra wajah 4), mami (citra wajah 5), jeine

(citra wajah 6) dan veny (citra wajah 7). Masing-masing citra wajah diambil 5 sampel dengan image yang berbeda-beda, sehingga total image yang diambil berjumlah 35 image.

**Tabel 1** Citra wajah dan data referensi

CITRA WAJAH 1	CITRA WAJAH 2	CITRA WAJAH 3	CITRA WAJAH 4	CITRA WAJAH 5	CITRA WAJAH 6	CITRA WAJAH 7
						
						
						
						
						

Dari ketujuh image yang diambil, 6 diantaranya merupakan penghuni rumah (sebagai referensi anggota keluarga) yaitu citra wajah 1 sampai

dengan citra wajah 6. Sedangkan citra wajah 7 merupakan image untuk orang yang bukan keluarga.

### C. Proses Image- Neural Network

Proses pengenalan wajah diawali dengan menginputkan citra wajah. Pemasukkan citra wajah sebagai input dilakukan dengan cara menekan *button* input wajah. Sedangkan untuk image processing menggunakan fungsi *button* image-Neural Network. (yang diberi tanda merah).



**Gambar 19.** Pemasukan citra wajah sebagai input

Saat *button* input wajah ditekan maka akan muncul menu membuka gambar sehingga bisa dipilih citra wajah yang akan dimasukkan. Citra wajah yang dimasukkan akan ditampilkan dalam program untuk bentuk citra warna.

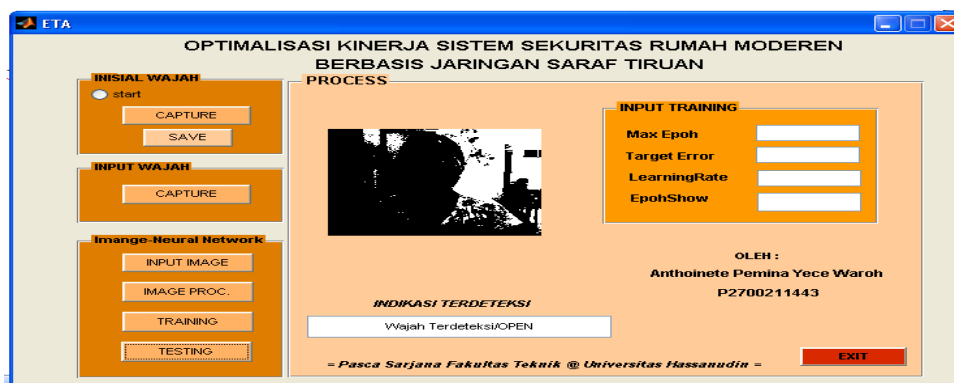
Citra wajah yang dimasukkan dalam bentuk citra warna akan diproses dengan mengkonversi citra wajah tersebut ke dalam bentuk citra *grayscale* dengan menekan *button* RGB-Gray, Gambar 19 menampilkan citra wajah

warna setelah pengambilan gambar dengan ukuran 160 x 120 pixel sama dengan 19200 pixel.



**Gambar 20.** Tampilan hasil Citra wajah warna yang sudah diproses dalam program

Citra wajah yang dimasukkan dalam bentuk citra warna akan diproses dengan mengkonversi citra wajah tersebut ke dalam bentuk citra *grayscale* dengan menekan *button RGB-Gray*, Gambar 20 menampilkan citra wajah warna setelah pengambilan gambar dengan ukuran 160 x 120 pixel sama dengan 19200 pixel.



**Gambar 21.** Citra wajah dalam bentuk citra *Grayscale*, dalam bentuk citra biner dan nilai biner

Kemudian citra wajah dalam bentuk citra Grayscale dikonversi kembali ke dalam bentuk citra biner (hitam-putih) dengan menekan *button* hitam-putih. Setelah diperoleh hasil citra biner (hitam putih) lalu dilakukan proses konversi ke nilai citra yang ditampilkan dalam program dengan melihat nilai pencitraan. Dalam pengujian penginputan wajah pada system dilakukan untuk ke 7 citra wajah dan diperlihatkan pada tabel 2.

**Tabel 2** karakter objek wajah

No.	Citra Input	Posisi Wajah	No.	Citra Input	Posisi Wajah
1	Eta	Biasa	16	Hoa	Biasa
2	Eta	Tersenyum	17	Hoa	Tersenyum
3	Eta	Serong kiri	18	Hoa	Serong kiri
4	Eta	Serong kanan	19	Hoa	Serong kanan
5	Eta	Mulut terbuka	20	Hoa	Mulut terbuka
6	Saul	Biasa	21	Mami	Biasa
7	Saul	Tersenyum	22	Mami	Tersenyum
8	Saul	Serong kiri	23	Mami	Serong kiri
9	Saul	Serong kanan	24	Mami	Serong kanan
10	Saul	Mulut terbuka	25	Mami	Mulut terbuka
11	Lino	Biasa	26	Jeine	Biasa



No.	Citra Input	Posisi Wajah	No.	Citra Input	Posisi Wajah
12	Lino	Tersenyum	27	Jeine	Tersenyum
13	Lino	Serong kiri	28	Jeine	Serong kiri
14	Lino	Serong kanan	29	Jeine	Serong kanan
15	Lino	Mulut terbuka	30	Jeine	Mulut terbuka

No.	Citra Input	Posisi Wajah	No.	Citra Input	Posisi Wajah
31	Veny	Biasa	34	Veny	Serong kanan
32	Veny	Tersenyum	35	Veny	Mulut terbuka
33	Veny	Serong kiri			

Objek wajah dari penelitian ini yaitu inisial eta, inisial saul, inisial lino, inisial hoa, inisial mami, inisial lino, inisial jeine dan inisial veny.

#### D. Pengujian kondisi citra biner

Pada pengujian kondisi output sistem akan mengaktif Solenoid untuk membuka anak kunci pintu (disimulasikan dengan lampu indikator) .

1. Inisial eta



**Gambar 22.** Wajah dikenal Sinyal yang keluar pada PC Lampu Hidup.

Pada gambar 21 adalah salah satu contoh wajah inisial eta pada tampilan PC dan lampu indicator dalam keadaan menyala menunjukkan program dapat mengenal wajah inisial eta dan untuk inisial saul, inisial lino, inisial hoa, inisial mami, inisial lino, inisial jeine diperlihatkan pada lampiran.

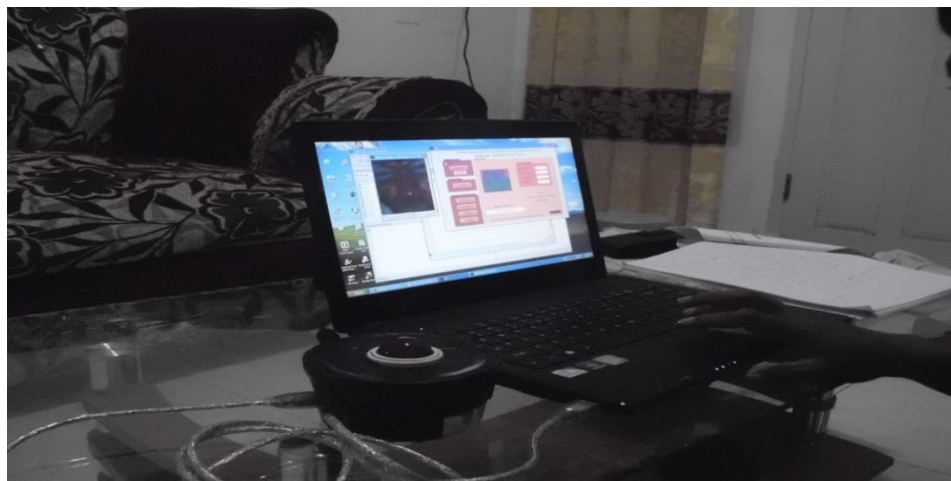
## 2. Inisial felix



**Gambar 23.** Citra wajah dalam bentuk citra *Grayscale*, citra biner dan nilai biner

Kemudian citra wajah dalam bentuk citra Grayscale dikonversi kembali ke dalam bentuk citra biner (hitam-putih) dengan menekan *button* hitam-putih.

Setelah diperoleh hasil citra biner (hitam putih) lalu dilakukan proses konversi ke nilai citra yang ditampilkan dalam program dengan melihat nilai pencitraan.



**Gambar 24.** Wajah Tidak dikenal Sinyal yang keluar pada PC Lampu Mati.

Pada gambar 23 adalah salah satu contoh wajah inisial felix pada tampilan PC dan lampu indikator dalam keadaan mati menunjukkan program tidak mengenal wajah inisial felix dan untuk inisial asrid, inisial indri, inisial rose, inisial lino, inisial cici, priskila ada pada lampiran.






## **1. Pengujian kondisi output berdasarkan input wajah**

### **a. Objek Wajah dengan inisial Nama Eta**

Proses pengujian keakuratan citra wajah menggunakan neural network (algoritma JST), dilakukan dengan cara menguji satu persatu

dari kelima citra wajah dari dua wajah ini adalah perwakilan dari tujuh wajah, yang lainnya ada pada lampiran inisil eta, saul, lino, mami, jeyne, dan veny. wajah berbeda sesuai dengan pengisian input wajah dengan kondisi output yang di hasilkan benar-benar teruji ketika citra wajah terdeteksi maka lampu akan menyala dan jika wajah tidak terdeteksi maka lampu akan mati, sesuai dengan data pada tabel 3 dan tabel 4.

**Tabel 3** hasil pengujian citra wajah pertama

Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
	Biasa	Menyala	Terdeteksi
	Tersenyum	Menyala	Terdeteksi
	Serong kiri	Menyala	Terdeteksi
	Serong kanan	Menyala	Terdeteksi
	Mulut terbuka	Menyala	Terdeteksi






Setelah pengisian proses awal telah dilakukan yaitu mengisi data dan menyimpan data dalam M-file dari kelima posisi wajah yaitu, wajah

biasa,wajah tersenyum,wajah serong kiri,wajah serong kanan,wajah mulut terbuka,kemudian tekan capture input wajah untuk menguji apakah wajah dikenal dan dicropping dan diproses pada histogram citra ,setelah itu tekan image prosesing ini akan merubah citra wajah warna menjadi bentuk grayscale citra biner dan nilai biner, kemudian tekan testing untuk menguji program yaitu indikasi terdeteksi/open dan lampu menyala,lihat pada lampiran. Pengujian saya lakukan dengan wajah menggunakan topi,wajah menggunakan kaca mata hitam,wajah menggunakan masker dan semuanya terdeteksi. Proses pengujian inisial saul,lino,hoa,mami (ibu saya),jeine,veny hasil pengujiannya terlampir pada lampiran.

b. Objek Wajah dengan inisial Nama marson

Target ketiga inisial marson diuji citra wajahnya yaitu pertama menekan komponen captur input maka akan tampil pada histogram citra (proses) akan tampil citra asli dengan posisi wajah asli (warna) tentunya dengan kelima posisi citra wajah pada tabel 4.3 satu persatu di uji kemudian tekan komponen input image ini membandingkan data yang tersimpan dengan citra wajah yang ada pada M-file dalam hal ini mencocokkan wajah,selanjutnya menekan image prosesing citra wajah dalam grayscale,proses terakhir yaitu mengetes program apakah terdeteksi atau tidak,setelah menekan komponen testing tampil indikasi tidak terdeteksi (wajah tidak dikenal pada PC) lampu mati /close dan ketuju wajah terlampir pada lampiran.

**Tabel 4** hasil pengujian citra wajah

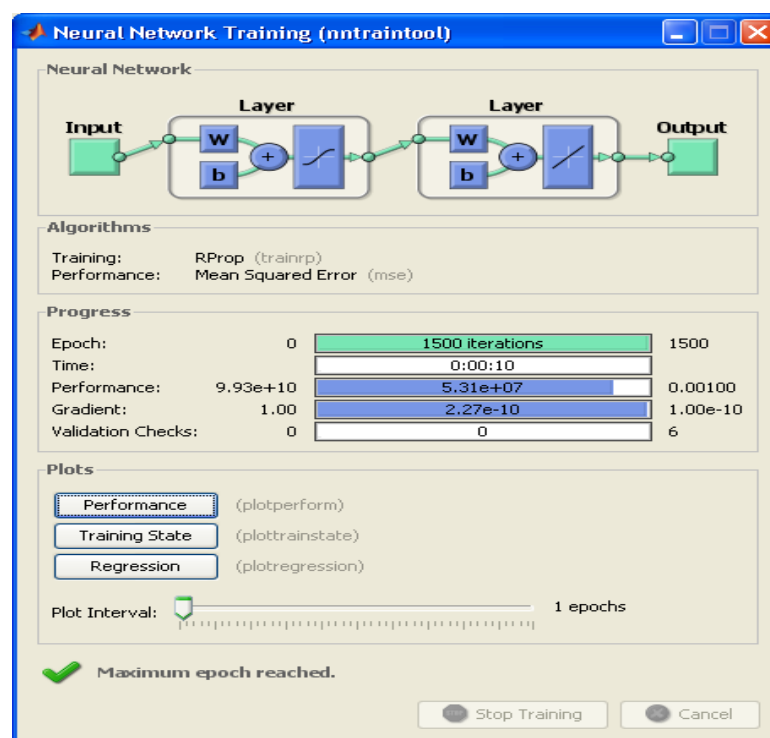
Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
	Biasa	Mati	Tidak Terdeteksi
	Tersenyum	Mati	Tidak Terdeteksi
	Serong kiri	Mati	Tidak Terdeteksi
	Serong kanan	Mati	Tidak Terdeteksi
	Mulut terbuka	Mati	Tidak Terdeteksi

Proses pengujian inisial bela,priskila,meisi,felix,asrid ,indri terlampir pada lembar lampiran.

## E. Neural Network

Untuk proses jaringan saraf tiruan langkah awal yang dilakukan setelah proses Image Processing adalah mengisi input training yaitu variable MaxEpoch = 1500, Target error =  $1e-3$ , Learningrate = 0.1, dan EpochShow adalah 200.

Setelah dilakukan pengisian parameter pelatihan JST maka akan dimasukan nilai target yaitu nilai pencitraan dari tiap-tiap citra wajah yang menjadi objek proses.



**Gambar 25.** Tampilan hasil training JST

Hasil pengujian data training Jaringan Saraf Tiruan dengan maksimum Epoch 1500 iterasi, waktu tempuh iterasi 10 detik untuk proses program untuk mengenali wajah sampai indikator lampu menyala. Performance

sebesar  $5.31 \times 10^7$ . nilai ini diperoleh 70% dari nilai maksimum performance, dengan nilai gradien  $2,27 \times 10^{-10}$ .

### 1. Pengujian Data Training sesuai wajah Input

Hasil pengujian data training ini bertujuan untuk mengetahui apakah sistem berjalan dengan sesuai program atau tidak dan apakah sistem ini benar-benar dapat mengoptimalkan kinerja sistem sekuritas rumah atau tidak, untuk hasilnya dapat dilihat pada tabel 5 sebagai berikut :

**Tabel 5** Hasil uji data testing sesuai input wajah untuk inisial eta

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
1	Wajah		Lampu	
ETA1	ETA	Biasa	Menyala	Terdeteksi
ETA2	ETA	Biasa	Menyala	Terdeteksi
ETA3	ETA	Biasa	Menyala	Terdeteksi
ETA4	ETA	Biasa	Menyala	Terdeteksi
ETA5	ETA	Biasa	Menyala	Terdeteksi
ETA6	ETA	Biasa	Menyala	Terdeteksi
ETA7	ETA	Biasa	Menyala	Terdeteksi
ETA8	ETA	Biasa	Menyala	Terdeteksi
ETA9	ETA	Biasa	Menyala	Terdeteksi
ETA10	ETA	Biasa	Menyala	Tidak terdeteksi
ETA1	ETA	Tersenyum	Menyala	Terdeteksi



Inisial Wajah 1	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
ETA2	ETA	Tersenyum	Menyala	Terdeteksi
ETA3	ETA	Tersenyum	Menyala	Terdeteksi
ETA4	ETA	Tersenyum	Menyala	Terdeteksi
ETA5	ETA	Tersenyum	Menyala	Terdeteksi
ETA6	ETA	Tersenyum	Menyala	Terdeteksi
ETA7	ETA	Tersenyum	Menyala	Terdeteksi
ETA8	ETA	Tersenyum	Menyala	Terdeteksi
ETA9	ETA	Tersenyum	Menyala	Terdeteksi
ETA10	ETA	Tersenyum	Menyala	Terdeteksi
ETA1	ETA	Serong Kiri	Menyala	Terdeteksi
ETA2	ETA	Serong Kiri	Menyala	Terdeteksi
ETA3	ETA	Serong Kiri	Menyala	Terdeteksi
ETA4	ETA	Serong Kiri	Menyala	Terdeteksi
ETA5	ETA	Serong Kiri	Menyala	Terdeteksi
ETA6	ETA	Serong Kiri	Menyala	Terdeteksi
ETA7	ETA	Serong Kiri	Menyala	Terdeteksi
ETA8	ETA	Serong Kiri	Menyala	Terdeteksi
ETA9	ETA	Serong Kiri	Menyala	Terdeteksi
ETA10	ETA	Serong Kiri	Menyala	Terdeteksi
ETA1	ETA	Serong Kanan	Menyala	Tidak Terdeteksi

Inisial Wajah 1	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
ETA2	ETA	Serong Kanan	Menyala	Terdeteksi
ETA3	ETA	Serong Kanan	Menyala	Terdeteksi
ETA4	ETA	Serong Kanan	Menyala	Terdeteksi
ETA5	ETA	Serong Kanan	Menyala	Terdeteksi
ETA6	ETA	Serong Kanan	Menyala	Terdeteksi
ETA7	ETA	Serong Kanan	Menyala	Terdeteksi
ETA8	ETA	Serong Kanan	Menyala	Terdeteksi
ETA9	ETA	Serong Kanan	Menyala	Terdeteksi
ETA10	ETA	Serong Kanan	Menyala	Terdeteksi
ETA1	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA2	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA3	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA4	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA5	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA6	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA7	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA8	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA9	ETA	Mulut Terbuka	Menyala	Terdeteksi
ETA10	ETA	Mulut Terbuka	Menyala	Terdeteksi

Hasil pengujian pada tabel 5 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\text{➤ } Ep = \left( \frac{jk}{Tp} \right) \times 100\%$$

$$= \left( \frac{2}{50} \right) \times 100\%$$

$$= 4\%$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 4\% \\ &= 96\% \end{aligned}$$

Dari hasil pengujian pada tabel 5 diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 2 kali kesalahan pada saat pengujian dilakukan. Hasil pengujian untuk input wajah dengan inisial lino diperlihatkan pada table 6.

**Tabel 6** Hasil uji data testing sesuai input wajah untuk inisial lino

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
LINO1	LINO	Biasa	Menyala	Terdeteksi
LINO2	LINO	Biasa	Menyala	Terdeteksi
LINO3	LINO	Biasa	Menyala	Terdeteksi
LINO4	LINO	Biasa	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
3	Wajah		Lampu	
LINO5	LINO	Biasa	Menyala	Terdeteksi
LINO6	LINO	Biasa	Menyala	Terdeteksi
LINO7	LINO	Biasa	Menyala	Terdeteksi
LINO8	LINO	Biasa	Menyala	Terdeteksi
LINO9	LINO	Biasa	Menyala	Terdeteksi
LINO10	LINO	Biasa	Menyala	Terdeteksi
LINO1	LINO	Tersenyum	Menyala	Terdeteksi
LINO2	LINO	Tersenyum	Menyala	Terdeteksi
LINO3	LINO	Tersenyum	Menyala	Terdeteksi
LINO4	LINO	Tersenyum	Menyala	Terdeteksi
LINO5	LINO	Tersenyum	Menyala	Terdeteksi
LINO6	LINO	Tersenyum	Menyala	Terdeteksi
LINO7	LINO	Tersenyum	Menyala	Terdeteksi
LINO8	LINO	Tersenyum	Menyala	Terdeteksi
LINO9	LINO	Tersenyum	Menyala	Terdeteksi
LINO10	LINO	Tersenyum	Menyala	Terdeteksi
LINO1	LINO	Serong Kiri	Menyala	Terdeteksi
LINO2	LINO	Serong Kiri	Menyala	Terdeteksi
LINO3	LINO	Serong Kiri	Menyala	Terdeteksi
LINO4	LINO	Serong Kiri	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
3	Wajah		Lampu	
LINO5	LINO	Serong Kiri	Menyala	Terdeteksi
LINO6	LINO	Serong Kiri	Menyala	Tidak Terdeteksi
LINO7	LINO	Serong Kiri	Menyala	Terdeteksi
LINO8	LINO	Serong Kiri	Menyala	Terdeteksi
LINO9	LINO	Serong Kiri	Menyala	Terdeteksi
LINO10	LINO	Serong Kiri	Menyala	Terdeteksi
LINO1	LINO	Serong Kanan	Menyala	Terdeteksi
LINO2	LINO	Serong Kanan	Menyala	Terdeteksi
LINO3	LINO	Serong Kanan	Menyala	Terdeteksi
LINO4	LINO	Serong Kanan	Menyala	Terdeteksi
LINO5	LINO	Serong Kanan	Menyala	Terdeteksi
LINO6	LINO	Serong Kanan	Menyala	Terdeteksi
LINO7	LINO	Serong Kanan	Menyala	Terdeteksi
LINO8	LINO	Serong Kanan	Menyala	Terdeteksi
LINO9	LINO	Serong Kanan	Menyala	Terdeteksi
LINO10	LINO	Serong Kanan	Mati	Terdeteksi
LINO1	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO2	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO3	LINO	Mulut Terbuka	Menyala	Tidak Terdeteksi
LINO4	LINO	Mulut Terbuka	Menyala	Terdeteksi

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
3				
LINO5	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO6	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO7	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO8	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO9	LINO	Mulut Terbuka	Menyala	Terdeteksi
LINO10	LINO	Mulut Terbuka	Menyala	Terdeteksi

Hasil pengujian pada tabel 6 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned}
 \text{➤ } Ep &= \left( \frac{jk}{Tp} \right) \times 100\% \\
 &= \left( \frac{2}{50} \right) \times 100\% \\
 &= 4\%
 \end{aligned}$$

$$\begin{aligned}
 \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\
 &= 100\% - 4\% \\
 &= 96\%
 \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Dari hasil pengujian pada tabel 6 maka diperoleh hasil keakuratan data sebesar 99% , kondisi ini diperoleh dengan hasil 2 kali salah dalam

pengujian pengujian. Hasil pengujian untuk input wajah dengan inisial eta diperlihatkan pada table 7.

**Tabel 7** Hasil uji data testing sesuai input wajah untuk inisial eta

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
4	Wajah		Lampu	
HOA1	HOA	Biasa	Menyala	Terdeteksi
HOA 2	HOA	Biasa	Menyala	Terdeteksi
HOA 3	HOA	Biasa	Menyala	Terdeteksi
HOA 4	HOA	Biasa	Menyala	Terdeteksi
HOA 5	HOA	Biasa	Menyala	Terdeteksi
HOA 6	HOA	Biasa	Menyala	Terdeteksi
HOA 7	HOA	Biasa	Menyala	Terdeteksi
HOA 8	HOA	Biasa	Menyala	Terdeteksi
HOA 9	HOA	Biasa	Menyala	Terdeteksi
HOA 10	HOA	Biasa	Menyala	Terdeteksi
HOA 1	HOA	Tersenyum	Menyala	Terdeteksi
HOA 2	HOA	Tersenyum	Menyala	Terdeteksi
HOA 3	HOA	Tersenyum	Menyala	Terdeteksi
HOA 4	HOA	Tersenyum	Menyala	Terdeteksi
HOA 5	HOA	Tersenyum	Menyala	Terdeteksi
HOA 6	HOA	Tersenyum	Menyala	Terdeteksi
HOA 7	HOA	Tersenyum	Menyala	Terdeteksi
HOA 8	HOA	Tersenyum	Menyala	Terdeteksi
HOA 9	HOA	Tersenyum	Menyala	Terdeteksi

Inisial Wajah 4	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
HOA 10	HOA	Tersenyum	Menyala	Terdeteksi
HOA 1	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 2	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 3	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 4	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 5	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 6	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 7	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 8	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 9	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 10	HOA	Serong Kiri	Menyala	Terdeteksi
HOA 1	HOA	Serong Kanan	Menyala	Tidak Terdeteksi
HOA 2	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 3	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 4	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 5	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 6	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 7	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 8	HOA	Serong Kanan	Menyala	Terdeteksi



Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
4				
HOA 9	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 10	HOA	Serong Kanan	Menyala	Terdeteksi
HOA 1	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 2	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 3	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 4	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 5	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 6	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 7	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 8	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 9	HOA	Mulut Terbuka	Menyala	Terdeteksi
HOA 10	HOA	Mulut Terbuka	Menyala	Tidak terdeteksi

Hasil pengujian pada tabel 7 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned}
 \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\
 &= \left( \frac{2}{50} \right) \times 100\% \\
 &= 4\%
 \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$\begin{aligned}
 \text{➤ \% Keakuratan} &= 100\% - E_p \\
 &= 100\% - 4\% \\
 &= 96\%
 \end{aligned}$$

Dari hasil pengujian pada tabel 7 Kesalahan 4 % dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 2 kali salah pengujian. Hasil pengujian untuk input wajah dengan inisial mami diperlihatkan pada table 8.

**Tabel 8** Hasil uji data testing sesuai input wajah untuk inisial mami

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
5				
MAMI1	MAMI	Biasa	Menyala	Terdeteksi
MAMI 2	MAMI	Biasa	Menyala	Terdeteksi
MAMI 3	MAMI	Biasa	Menyala	Terdeteksi
MAMI 4	MAMI	Biasa	Menyala	Terdeteksi
MAMI 5	MAMI	Biasa	Menyala	Terdeteksi
MAMI 6	MAMI	Biasa	Menyala	Terdeteksi
MAMI 7	MAMI	Biasa	Menyala	Terdeteksi
MAMI 8	MAMI	Biasa	Menyala	Terdeteksi
MAMI 9	MAMI	Biasa	Menyala	Terdeteksi
MAMI 10	MAMI	Biasa	Menyala	Terdeteksi
MAMI 1	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 2	MAMI	Tersenyum	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
5	Wajah			
MAMI 3	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 4	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 5	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 6	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 7	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 8	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 9	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 10	MAMI	Tersenyum	Menyala	Terdeteksi
MAMI 1	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 2	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 3	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 4	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 5	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 6	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 7	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 8	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 9	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 10	MAMI	Serong Kiri	Menyala	Terdeteksi
MAMI 1	MAMI	Serong Kanan	Menyala	Tidak Terdeteksi
MAMI 2	MAMI	Serong Kanan	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
5	Wajah		Lampu	
MAMI 3	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 4	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 5	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 6	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 7	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 8	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 9	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 10	MAMI	Serong Kanan	Menyala	Terdeteksi
MAMI 1	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 2	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 3	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 4	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 5	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 6	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 7	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 8	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 9	MAMI	Mulut Terbuka	Menyala	Terdeteksi
MAMI 10	MAMI	Mulut Terbuka	Menyala	Terdeteksi

Hasil pengujian pada tabel 8 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\text{➤ } Ep = \left( \frac{jk}{tp} \right) \times 100\%$$

$$= \left( \frac{2}{50} \right) \times 100\%$$

$$= 4\%$$

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 4\% \\ &= 96\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

Dari hasil pengujian pada tabel 8 Kesalahan 4 % dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 2 kali salah pengujian. Hasil pengujian untuk input wajah dengan inisial jeine diperlihatkan pada table 9.

**Tabel 9** Hasil uji data testing sesuai input wajah untuk inisial jeine

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
JEINE 1	JEINE	Biasa	Menyala	Terdeteksi
JEINE 2	JEINE	Biasa	Menyala	Terdeteksi
JEINE 3	JEINE	Biasa	Menyala	Terdeteksi
JEINE 4	JEINE	Biasa	Menyala	Terdeteksi
JEINE 5	JEINE	Biasa	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
6	Wajah			
JEINE 6	JEINE	Biasa	Menyala	Terdeteksi
JEINE 7	JEINE	Biasa	Menyala	Terdeteksi
JEINE 8	JEINE	Biasa	Menyala	Terdeteksi
JEINE 9	JEINE	Biasa	Menyala	Terdeteksi
JEINE 10	JEINE	Biasa	Menyala	Terdeteksi
JEINE 1	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 2	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 3	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 4	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 5	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 6	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 7	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 8	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 9	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 10	JEINE	Tersenyum	Menyala	Terdeteksi
JEINE 1	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 2	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 3	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 4	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 5	JEINE	Serong Kiri	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
6	Wajah			
JEINE 6	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 7	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 8	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 9	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 10	JEINE	Serong Kiri	Menyala	Terdeteksi
JEINE 1	JEINE	Serong Kanan	Menyala	Tidak Terdeteksi
JEINE 2	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 3	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 4	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 5	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 6	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 7	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 8	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 9	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 10	JEINE	Serong Kanan	Menyala	Terdeteksi
JEINE 1	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 2	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 3	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 4	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 5	JEINE	Mulut Terbuka	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
6	Wajah			
JEINE 6	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 7	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 8	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 9	JEINE	Mulut Terbuka	Menyala	Terdeteksi
JEINE 10	JEINE	Mulut Terbuka	Menyala	Terdeteksi

Hasil pengujian pada tabel 9 diatas,dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah,dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\text{➤ } Ep = \left( \frac{jk}{tp} \right) \times 100\%$$

$$= \left( \frac{2}{50} \right) \times 100\%$$

$$= 4\%$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi

kesalahan (error)

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 4\% \\ &= 96\% \end{aligned}$$

Dari hasil pengujian pada tabel 9 Kesalahan 4 % dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 2 kali salah pengujian. Hasil pengujian untuk input wajah dengan inisial veny diperlihatkan pada table 10.



**Tabel 10** Hasil uji data testing sesuai input wajah untuk inisialisasi veny

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
7	VENY	Biasa	Menyala	Terdeteksi
VENY 1	VENY	Biasa	Menyala	Terdeteksi
VENY 2	VENY	Biasa	Menyala	Terdeteksi
VENY 3	VENY	Biasa	Menyala	Terdeteksi
VENY 4	VENY	Biasa	Menyala	Terdeteksi
VENY 5	VENY	Biasa	Menyala	Terdeteksi
VENY 6	VENY	Biasa	Menyala	Terdeteksi
VENY 7	VENY	Biasa	Menyala	Terdeteksi
VENY 8	VENY	Biasa	Menyala	Terdeteksi
VENY 9	VENY	Biasa	Menyala	Terdeteksi
VENY 10	VENY	Biasa	Menyala	Terdeteksi
VENY 1	VENY	Tersenyum	Menyala	Terdeteksi
VENY 2	VENY	Tersenyum	Menyala	Terdeteksi
VENY 3	VENY	Tersenyum	Menyala	Terdeteksi
VENY 4	VENY	Tersenyum	Menyala	Terdeteksi
VENY 5	VENY	Tersenyum	Menyala	Terdeteksi
VENY 6	VENY	Tersenyum	Menyala	Terdeteksi
VENY 7	VENY	Tersenyum	Menyala	Terdeteksi
VENY 8	VENY	Tersenyum	Menyala	Terdeteksi
VENY 9	VENY	Tersenyum	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
7	Wajah			
VENY 10	VENY	Tersenyum	Menyala	Terdeteksi
VENY 1	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 2	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 3	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 4	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 5	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 6	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 7	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 8	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 9	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 10	VENY	Serong Kiri	Menyala	Terdeteksi
VENY 1	VENY	Serong Kanan	Menyala	Tidak Terdeteksi
VENY 2	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 3	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 4	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 5	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 6	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 7	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 8	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 9	VENY	Serong Kanan	Menyala	Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
7	Wajah			
VENY 10	VENY	Serong Kanan	Menyala	Terdeteksi
VENY 1	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 2	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 3	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 4	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 5	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 6	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 7	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 8	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 9	VENY	Mulut Terbuka	Menyala	Terdeteksi
VENY 10	VENY	Mulut Terbuka	Menyala	Terdeteksi

Hasil pengujian pada tabel 10 diatas,dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah,dimana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\text{➤ } Ep = \left( \frac{jk}{tp} \right) \times 100\%$$

$$= \left( \frac{2}{50} \right) \times 100\%$$

$$= 4\%$$

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 4\% \\ &= 96\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

Dari hasil pengujian pada tabel 10 Kesalahan 4 % dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 2 kali salah pengujian. Hasil pengujian untuk input wajah dengan inisial marson diperlihatkan pada table 11.

**Tabel 11** Hasil uji data testing tidak sesuai input wajah marson

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
MARSON1	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON2	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON3	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON4	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON5	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON6	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON7	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON8	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON9	SAUL	Biasa	Mati	Tidak Terdeteksi
MARSON10	SAUL	Biasa	Mati	Tidak Terdeteksi

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
MARSON1	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON2	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON3	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON4	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON5	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON6	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON7	SAUL	Tersenyum	Mati	Tidak Terdeteksi
MARSON 8	ETA	Tersenyum	Mati	Tidak Terdeteksi
MARSON 9	ETA	Tersenyum	Mati	Tidak Terdeteksi
MARSON 10	ETA	Tersenyum	Mati	Tidak Terdeteksi
MARSON 1	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 2	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 3	ETA	Serong Kiri	Mati	Terdeteksi
MARSON 4	ETA	Serong Kiri	Mati	Tidak Terdeteksi

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
MARSON 5	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 6	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 7	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 8	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 9	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 10	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MARSON 1	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 2	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 3	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 4	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 5	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 6	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 7	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 8	ETA	Serong Kanan	Mati	Tidak Terdeteksi

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
MARSON 9	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 10	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MARSON 1	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 2	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 3	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 4	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 5	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 6	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 7	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 8	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 9	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MARSON 10	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 11 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned} \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\ &= \left( \frac{2}{50} \right) \times 100\% \\ &= 4\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 4\% \\ &= 96\% \end{aligned}$$

Dari hasil pengujian pada tabel 11 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian. Hasil pengujian untuk input wajah dengan inisial bela diperlihatkan pada table 12.

**Tabel 12** Hasil uji data testing tidak sesuai input wajah inisial bela

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
BELA1	ETA	Biasa	Mati	Tidak Terdeteksi
BELA2	ETA	Biasa	Mati	Tidak Terdeteksi
BELA3	ETA	Biasa	Mati	Tidak Terdeteksi
BELA4	ETA	Biasa	Mati	Tidak Terdeteksi
BELA5	ETA	Biasa	Mati	Tidak Terdeteksi



Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
BELA6	ETA	Biasa	Mati	Tidak Terdeteksi
BELA7	ETA	Biasa	Mati	Tidak Terdeteksi
BELA8	ETA	Biasa	Mati	Tidak Terdeteksi
BELA9	ETA	Biasa	Mati	Tidak Terdeteksi
BELA10	ETA	Biasa	Mati	Tidak Terdeteksi
BELA1	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA2	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA3	ETA	Tersenyum	Mati	Terdeteksi
BELA4	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA5	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA6	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA7	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA8	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA9	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA10	ETA	Tersenyum	Mati	Tidak Terdeteksi
BELA1	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA2	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA3	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA4	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA5	ETA	Serong Kiri	Mati	Tidak Terdeteksi

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
BELA6	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA7	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA8	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA9	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA10	ETA	Serong Kiri	Mati	Tidak Terdeteksi
BELA1	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA2	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA3	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA4	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA5	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA6	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA7	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA8	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA9	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA10	ETA	Serong Kanan	Mati	Tidak Terdeteksi
BELA1	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA2	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA3	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA4	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA5	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi

Inisial Wajah I	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
BELA6	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA7	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA8	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA9	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
BELA10	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 12 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned}
 \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\
 &= \left( \frac{2}{50} \right) \times 100\% \\
 &= 4\%
 \end{aligned}$$

Dimana:

jk = jumlah kesalahan  
pengujian

TP = Total pengukuran

Ep = jumlah persentasi  
kesalahan (error)

$$\begin{aligned}
 \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\
 &= 100\% - 4\% \\
 &= 96\%
 \end{aligned}$$

Dari hasil pengujian pada tabel 12 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali

salah pengujian. Hasil pengujian untuk input wajah dengan inisial seren diperlihatkan pada table 13.

**Tabel 13** Hasil uji data testing tidak sesuai input wajah untuk orang yang berinisial seren

Inisial Wajah 2	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
PRISKILA1	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA2	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA3	LINO	Biasa	Mati	Terdeteksi
PRISKILA4	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA5	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA6	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA7	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA8	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA9	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA10	LINO	Biasa	Mati	Tidak Terdeteksi
PRISKILA1	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA2	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA3	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA4	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA5	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA6	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA7	LINO	Tersenyum	Mati	Tidak Terdeteksi

Inisial Wajah 2	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
PRISKILA8	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA9	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA10	LINO	Tersenyum	Mati	Tidak Terdeteksi
PRISKILA1	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA2	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA3	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA4	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA5	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA6	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA7	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA8	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA9	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA10	LINO	Serong Kiri	Mati	Tidak Terdeteksi
PRISKILA1	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA2	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA3	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA4	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA5	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA6	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA7	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA8	LINO	Serong Kanan	Mati	Tidak Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
2	Wajah		Lampu	
PRISKILA9	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA10	LINO	Serong Kanan	Mati	Tidak Terdeteksi
PRISKILA1	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA2	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA3	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA4	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA5	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA6	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA7	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA8	LINO	Mulut Terbuka	Lampu menyala	Terdeteksi
PRISKILA9	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi
PRISKILA10	LINO	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 13 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned} \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\ &= \left( \frac{1}{50} \right) \times 100\% \\ &= 2\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$\begin{aligned}
 \text{➤ \% Keakuratan} &= 100\% - E_p \\
 &= 100\% - 2\% \\
 &= 98\%
 \end{aligned}$$

Dari hasil pengujian pada tabel 13 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

**Tabel 14** Hasil uji data testing wajahnya inisial meisi mirip inisial eta.

Inisial Wajah : 3	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
MEISI1	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 2	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 3	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 4	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 5	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 6	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 7	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 8	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 9	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 10	ETA	Biasa	Mati	Tidak Terdeteksi
MEISI 1	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 2	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 3	ETA	Tersenyum	Mati	Terdeteksi

Inisial Wajah :	Input	Posisi Wajah	Keadaan	Hasil Pengujian
3	Wajah		Lampu	
MEISI 4	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 5	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 6	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 7	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 8	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 9	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 10	ETA	Tersenyum	Mati	Tidak Terdeteksi
MEISI 1	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 2	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 3	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 4	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 5	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 6	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 7	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 8	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 9	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 10	ETA	Serong Kiri	Mati	Tidak Terdeteksi
MEISI 1	ETA	Serong Kanan	Lampu menyala	Terdeteksi
MEISI 2	ETA	Serong Kanan	Mati	Tidak Terdeteksi



Inisial Wajah :	Input	Posisi Wajah	Keadaan	Hasil Pengujian
3	Wajah		Lampu	
MEISI 3	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 4	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 5	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 6	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 7	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 8	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 9	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 10	ETA	Serong Kanan	Mati	Tidak Terdeteksi
MEISI 1	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 2	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 3	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 4	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 5	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 6	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 7	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 8	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 9	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi
MEISI 10	ETA	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 14 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned} \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\ &= \left( \frac{1}{50} \right) \times 100\% \\ &= 2\% \end{aligned}$$

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

Dari hasil pengujian pada tabel 14 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

**Tabel 15** Hasil uji data testing tidak sesuai input wajah untuk inisial felix

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
FELIX1	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 2	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 3	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 4	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 5	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 6	SAUL	Biasa	Mati	Tidak Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
4	Wajah		Lampu	
FELIX 7	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 8	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 9	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 10	SAUL	Biasa	Mati	Tidak Terdeteksi
FELIX 1	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 2	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 3	SAUL	Tersenyum	Mati	Terdeteksi
FELIX 4	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 5	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 6	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 7	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 8	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 9	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 10	SAUL	Tersenyum	Mati	Tidak Terdeteksi
FELIX 1	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 2	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 3	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 4	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 5	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 6	SAUL	Serong Kiri	Mati	Tidak Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
4	Wajah		Lampu	
FELIX 7	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 8	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 9	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 10	SAUL	Serong Kiri	Mati	Tidak Terdeteksi
FELIX 1	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 2	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 3	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 4	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 5	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 6	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 7	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 8	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 9	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 10	SAUL	Serong Kanan	Mati	Tidak Terdeteksi
FELIX 1	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 2	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 3	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 4	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 5	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 6	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
4	Wajah			
FELIX 7	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 8	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 9	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi
FELIX 10	SAUL	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 15 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned}
 \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\
 &= \left( \frac{1}{50} \right) \times 100\% \\
 &= 2\%
 \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$\begin{aligned}
 \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\
 &= 100\% - 2\% \\
 &= 98\%
 \end{aligned}$$

Dari hasil pengujian pada tabel 15 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

**Tabel 16** Hasil uji data testing tidak sesuai input wajah untuk inisial asrid

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
5	Wajah		Lampu	
ASRID1	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID2	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID3	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID4	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID5	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID6	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID7	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID8	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID9	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID10	HOA	Biasa	Mati	Tidak Terdeteksi
ASRID1	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID2	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID3	HOA	Tersenyum	Mati	Terdeteksi
ASRID4	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID5	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID6	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID7	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID8	HOA	Tersenyum	Mati	Tidak Terdeteksi

ASRID9	HOA	Tersenyum	Mati	Tidak Terdeteksi
Inisial Wajah :	Input	Posisi Wajah	Keadaan	Hasil Pengujian
5	Wajah		Lampu	
ASRID10	HOA	Tersenyum	Mati	Tidak Terdeteksi
ASRID1	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID2	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID3	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID4	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID5	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID6	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID7	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID8	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID9	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID10	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ASRID1	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID2	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID3	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID4	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID5	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID6	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID7	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID8	HOA	Serong Kanan	Mati	Tidak Terdeteksi

ASRID9	HOA	Serong Kanan	Mati	Tidak Terdeteksi
Inisial Wajah : 5	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
ASRID10	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ASRID1	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID2	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID3	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID4	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID5	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID6	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID7	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID8	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID9	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ASRID10	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi

$$\begin{aligned} \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\ &= \left( \frac{1}{50} \right) \times 100\% \\ &= 2\% \end{aligned}$$

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)



Dari hasil pengujian pada tabel 16 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

**Tabel 17** Hasil uji data testing tidak sesuai input wajah untuk inisial asrid

Inisial Wajah 6	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
INDRI1	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI2	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI3	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI4	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI5	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI6	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI7	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI8	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI9	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI10	HOA	Biasa	Mati	Tidak Terdeteksi
INDRI11	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI12	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI13	HOA	Tersenyum	Mati	Terdeteksi
INDRI14	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI15	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI16	HOA	Tersenyum	Mati	Tidak Terdeteksi

Inisial Wajah 6	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
INDRI7	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI8	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI9	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI10	HOA	Tersenyum	Mati	Tidak Terdeteksi
INDRI11	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI12	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI13	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI14	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI15	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI16	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI17	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI18	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI19	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI10	HOA	Serong Kiri	Mati	Tidak Terdeteksi
INDRI11	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI12	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI13	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI14	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI15	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI16	HOA	Serong Kanan	Mati	Tidak Terdeteksi

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
6				
INDRI7	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI8	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI9	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI10	HOA	Serong Kanan	Mati	Tidak Terdeteksi
INDRI11	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI12	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI13	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI14	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI15	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI16	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI17	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI18	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI19	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
INDRI10	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 17 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\begin{aligned} \text{➤ } Ep &= \left( \frac{jk}{tp} \right) \times 100\% \\ &= \left( \frac{1}{50} \right) \times 100\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

$$= 2\%$$

$$\begin{aligned} \text{➤ \% Keakuratan} &= 100\% - E_p \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Dari hasil pengujian pada tabel 17 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 96% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

**Tabel 18** Hasil uji data testing tidak sesuai input wajah

Inisial Wajah	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
7				
ROSYE1	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE2	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE3	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE4	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE5	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE6	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE7	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE8	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE9	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE10	HOA	Biasa	Mati	Tidak Terdeteksi
ROSYE1	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE2	HOA	Tersenyum	Mati	Tidak Terdeteksi

Inisial Wajah 7	Input Wajah	Posisi Wajah	Keadaan Lampu	Hasil Pengujian
ROSYE3	HOA	Tersenyum	Mati	Terdeteksi
ROSYE4	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE5	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE6	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE7	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE8	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE9	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE10	HOA	Tersenyum	Mati	Tidak Terdeteksi
ROSYE1	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE2	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE3	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE4	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE5	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE6	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE7	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE8	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE9	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE10	HOA	Serong Kiri	Mati	Tidak Terdeteksi
ROSYE1	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE2	HOA	Serong Kanan	Mati	Tidak Terdeteksi

Inisial Wajah	Input	Posisi Wajah	Keadaan	Hasil Pengujian
7	Wajah		Lampu	
ROSYE3	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE4	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE5	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE6	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE7	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE8	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE9	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE10	HOA	Serong Kanan	Mati	Tidak Terdeteksi
ROSYE1	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE2	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE3	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE4	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE5	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE6	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE7	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE8	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE9	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi
ROSYE10	HOA	Mulut Terbuka	Mati	Tidak Terdeteksi

Hasil pengujian pada tabel 18 diatas, dapat diketahui bahwa keakuratan sistem untuk citra wajah yang sesuai input wajah, di mana hasil ekstraksi cirinya sebagai pola data referensi (citra latih) adalah :

$$\text{➤ } Ep = \left( \frac{jk}{tp} \right) \times 100\%$$

$$= \left( \frac{2}{50} \right) \times 100\%$$

$$= 2\%$$

$$\begin{aligned} \text{➤ } \% \text{ Keakuratan} &= 100\% - Ep \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Dimana:

jk = jumlah kesalahan pengujian

TP = Total pengukuran

Ep = jumlah persentasi kesalahan (error)

Dari hasil pengujian pada tabel 18 Kesalahan 2% dan diperoleh hasil keakuratan data sebesar 98% , kondisi ini diperoleh dengan hasil 1 kali salah pengujian.

## F. Kelemahan Sistem yang dibangun

Analisis kelemahan sistem rumah cerdas berbasis JST merupakan parameter untuk mengetahui kekurangan dari perancangan sistem. Parameter tersebut yaitu data training yang sesuai dan data training yang tidak sesuai citra wajah. Dengan pembuatan simulasi ini, uji coba yang dilakukan menggunakan wajah manusia dengan bentuk 5 posisi wajah , jenis kelamin yang berbeda dan usia yang berbeda. Hasil percobaan ini membuktikan sistem dapat mengenalinya dengan persentase 96%, jumlah kesalahannya mencapai 4% . Dengan capaian kesalahan ini tentunya

menunjukkan teknis sistem sekuritas rumah cerdas berbasis JST masih ada kelemahannya, antara lain yaitu Proses pencahayaan pada histogram citranya tidak dapat diatur intensitasnya pada perintah programnya sehingga saat pengujian sistem dilakukan dimana situasi lampu rumah modern gelap atau turun tegangan dari PLN maka program tidak dapat mengenal wajah (tidak terdeteksi ) sehingga pemilik rumah tidak dapat masuk kedalam rumah,karena pintu tertutup.



## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Setelah melakukan penelitian dalam tesis ini, maka dapat di tarik kesimpulan sebagai berikut:

1. Dapat menghasilkan sebuah sistem sekuritas rumah modern berbasis jaringan saraf tiruan (JST) dengan output berupa isyarat lampu indikator dan pintu rumah, dengan program yang dapat mendeteksi citra wajah manusia pada saat memasuki rumah modern.
2. Pengenalan wajah melalui webcam dengan algoritma back propagation menggunakan program matlab ternyata dapat mengakses data citra dan dapat mengoptimalkan kerja sistem sekuritas akses masuk rumah moderen.
3. Hasil keakuratan sistem untuk citra wajah yang sesuai dengan input wajah, diperoleh hasil keakuratan data sebesar 96% , dimana kondisi ini diperoleh dengan kesalahan pengujian data sebanyak 1 kali untuk 50 kali pengujian. Untuk citra wajah yang tidak sesuai, hasil pengujian yang diperoleh menunjukkan bahwa keakuratan data sebesar 96% untuk 50 kali pengujian.

## **B. Saran**

Dalam pembuatan sistem ini begitu banyak kekurangannya untuk itu saran dan petunjuk dapat dilakukan untuk menyempurnakan sistem antara lain :

1. Sistem yang dibuat dengan metode jaringan saraf tiruan dapat diaplikasikan pada metode lain. Membuat program untuk mendeteksi pola wajah yang sama (kembar yang identik) dan mendeteksi pola wajah kembar dengan jenis kelamin berbeda.
2. Sebaiknya menggunakan kamera digital ISO 200 dan ISO 400 tergantung situasi desekitar kamera.

## DAFTAR PUSTAKA

- Abdul Fadlil,Ikhson Hidayat,Sunardi, *Sistem pengenalan wajah manusia real-time menggunakan algoritma jaringan saraf tiruan* , Journal
- Jimmy E Purwanto, (2003). *Prototype mengenal wajah melalui webcam dengan menggunakan algoritma principal component analysis (PCA) dan Linier discriminant analisis (LDA)*, Journal
- Agus Buono,Amad Ridha,Harief Batian, *Sistem pengenalan wajah real-time ruang eigen dengan segmentasi berdasarkan warna kulit*, Journal
- Booch,G. Rumbaugh,J. Jacobson,I. (1999). *The Unified Modeling Language - User Guide*. Addison Wesley.
- Kendall, K. and Kendall, J., (2005) *Systems Analysis and Design*, 6th Ed. Prentice Hall.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dan USDP*. ANDI.
- Oetomo, Budi Sutedjo Dharma, (2006), *Perencanaan dan Pembangunan Sistem Informasi*,ANDI, Yogyakarta.
- Alan Budi (2007) *Pengolahan Citra (on line)* (elib,unikom.ac.id/files/disk 1/126/jbtunikom pp-gdl-SL-2007-alanbuduha-babii-pdf)
- Pressman, R. S. (2010). *Software Engineering : A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill.
- Sobottka,K dan Pifers,I. (1996). *Looking for face and facial features in color Image*.<http://citescer.lcs.mit.edu/sobottka96looking.html>.(19 meri 2013)
- Kusumadewi,S. (2004). *Membangun Jaringan Saraf Tiruan Menggunakan Matlab & Exel link*.Yogyakarta Garah Ilmu.
- Fausett,L.(1994). *Fundamental of Neural Network.Archtectures,Algorithm, and Application*.New jersey :Prentice\_Hall.
- Brenda Palit (2011). *Pendeteksi wajah dengan pengaruh cahaya*, Skripsi, UNSRAT. Manado

## LAMPIRAN 1

### Listing Program Pengenalan Pola Wajah Manusia.

```

function varargout = ETA(varargin)
% ETA M-file for ETA.fig
%     ETA, by itself, creates a new ETA or raises the existing
%     singleton*.
%
%     H = ETA returns the handle to a new ETA or the handle to
%     the existing singleton*.
%
%     ETA('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in ETA.M with the given input
arguments.
%
%     ETA('Property','Value',...) creates a new ETA or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before ETA_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to ETA_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ETA

% Last Modified by GUIDE v2.5 04-May-2008 22:39:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ETA_OpeningFcn, ...
                  'gui_OutputFcn',  @ETA_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before ETA is made visible.
function ETA_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ETA (see VARARGIN)

% Choose default command line output for ETA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes ETA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ETA_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
%axes(handles.axes5); imshow('picture0015.jpg');
%axes(handles.axes6); imshow('picture0017.jpg');

global vid
vid = videoinput('winvideo');
start(vid);
%vidRes = get(vid, 'VideoResolution');
%imWidth = vidRes(1);
%imHeight = vidRes(2);
%nBands = get(vid, 'NumberOfBands');
%hImage = image( zeros(imHeight, imWidth, nBands) );
preview(vid);%(vid, hImage);
%axes(handles.axes1); imshow(vid);

% --- Executes on button press in CAPTURE1.
function CAPTURE1_Callback(hObject, eventdata, handles)
% hObject    handle to CAPTURE1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global vid rgb
%pause(30);

```

```

rgb = getsnapshot(vid);
axes(handles.axes1);imshow(rgb);

% --- Executes on button press in SAVE.
function SAVE_Callback(hObject, eventdata, handles)
% hObject      handle to SAVE (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global rgb D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16
D17 D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31 32
D3 D34 D35
gambar a
b=1;
gambar=a+b;
%set(handles.edit8,'string',gambar);
if (gambar==1)
    a=gambar;
D1 = rgb;
    else if (gambar==2)B
    a=gambar;
    D2 = rgb;
        else if (gambar==3)
    a=gambar;
        D3 = rgb;
            else if (gambar==4)
    a=gambar;
                D4 = rgb;
                    else if (gambar==5)
    a=gambar;
                        D5 = rgb;
                            else if (gambar==6)
    a=gambar;
                                D6 = rgb;
                                    else if (gambar==7)
    a=gambar;
                                        D7 = rgb;
                                            else if (gambar==8)
    a=gambar;
                                                D8 = rgb;
                                                    else if (gambar==9)
    a=gambar;
                                                        D9 = rgb;

                                                    else if (gambar==10)
    a=0;
                                                        D10 = rgb;
                                                        gambar=0;

                                                    end
                                                end
                                        end
                                end
                        end
                end
        end
end
end

```

```

        end
        end
        end
    end
end

image1=uiinputfile(...
{'*.bmp','file citra (*.bmp)';
'*.bmp','citra bmp (*.bmp)';...
'*.','Semua File (*.*)'},...
'Menyimpan Citra');
imwrite(rgb,fullfile(image1),'BMP');

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
selection = questdlg(['Tutup ' get(handles.figure1,'Name')
'?'],...

```

```

['Tutup ' get(handles.figure1,'Name') '...'],...
'Ya','Tidak','Ya');
if strcmp(selection,'Tidak')
return;
end
delete(handles.figure1)

% --- Executes on button press in IMANGE_PROCC.
function IMANGE_PROCC_Callback(hObject, eventdata, handles)
% hObject    handle to IMANGE_PROCC (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global  rgb1 data2 BW1

I1 = rgb2gray(rgb1);
BW1 = im2bw(I1);%(I2,graythresh(I2));

axes(handles.axes1);imshow(BW1);
data2=bwarea(BW1);
%=====set(handles.edit8,'string',data2);

%I = rgb2gray(rgb);
%BW = im2bw(I);%(I2,graythresh(I2));
%axes(handles.axes1);imshow(BW);
%data1=bwarea(BW)
%set(handles.edit7,'string',data1);

% --- Executes on button press in TRAINING.
function TRAINING_Callback(hObject, eventdata, handles)
% hObject    handle to TRAINING (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Maxepoh TError Lr EpochSow D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11
D12 D13 D14 D15 D16 D17 D17 D18 D19 D20 D21 D22 D23 D24 D25 D26
D27 D28 D29 D30 D31 32 D3 D34 D35
Y Bobotakhir_Bias_Lapisan P data2
Maxepoh=str2num(get(handles.edit3, 'String'));
TError=str2num(get(handles.edit4, 'String'));
Lr=str2num(get(handles.edit5, 'String'));
EpochSow=str2num(get(handles.edit6, 'String'));

global Maxepoh TError Lr EpochSow D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11
D12 D13 D14 D15 D16 D17 D17 D18 D19 D20 D21 D22 D23 D24 D25 D26
D27 D28 D29 D30 D31 32 D3 D34 D35
P= [D1 D2 D3 D4 D5; D6 D7 D8 D9 D10];      %[1 2 3 4 5;2 4 6 8 10];
%[0 1 2 1 10 12 -5 -8 -10 -15; 0 1 -1 6 3 -1 -2 2 -5 2];
T=[D1 D2 D3 D4 D5 ]; % 15916 16048 15852 16328 16361];      %0 0 1 1
2 2 -1 -1 -2 -2];

```



```

%membangun jaringan saraf feedforward
net = newff(minmax(P),[5 1],{'tansig' 'purelin'},'trainrp');

%melihat bobotawal input, lapidsan dan bias
Bobotawal_input      = net.IW{1,1}
Bobotawal_Bias_input = net.b{1,1}
Bobotawal_Lapisan    = net.LW{2,1}
Bobotawal_Bias_lapisan = net.b{2,1}

%set max epoh , goal, learning rate, show step
net.trainParam.epochs = Maxepoh;%1500;
net.trainParam.goal    = TError; %1e-3;
net.trainParam.lr      = Lr; %0.1;
net.trainParam.show    = EpochSow; %100;

%melakukan pembelajaran
net = train(net,P,T);

%melihat bobot2 awal input, lapisan dan , bias
Bobotakhir_input      = net.IW{1,1}
Bobotakhir_Bias_Input = net.b{1,1}
Bobotakhir_Lapisan    = net.LW{2,1}
Bobotakhir_Bias_Lapisan = net.b{2,1}

%Melakukan Simulasi
Y = sim(net,P)

%gambar Grafik
pause;
subplot(211)
plot(P(1,:),T,'bo',P(1,:),Y,'r*');
title('Perbandingan antara target (0) dan Output Jaringan (*)');
xlabel('Input Pertama');
ylabel('Target atau Output');
grid;

subplot(212)
plot(P(2,:),T,'bo',P(2,:),Y,'r*');
title('Perbandingan antara target (0) dan Output Jaringan (*)');
xlabel('Input Kedua');
ylabel('Target atau Output');
grid;
set(handles.edit2,'string',Y);

% --- Executes on button press in TESTING.
function TESTING_Callback(hObject, eventdata, handles)
% hObject      handle to TESTING (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global D1 D6 data2 %rgb
%axes(handles.axes3); imshow(rgb);

```

```

data=data2-D1;
%set(handles.edit4,'string',data);
if(data>300)
    %set(handles.edit2,'string','tdk sama');
    data=data2-D6;
    if(data>300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOSE');
    else if(data<-300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOSE');
    else
        set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
        delete(instrfind);
        %clear all;
        s = serial('COM15');

set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'F
lowControl','none','Terminator','CR/LF');
fopen(s);
fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;

        end
    end
else if(data<-300)
    %set(handles.edit2,'string','Tidak Terdeteksi/CLOSE');
    data=data2-D6;
    if(data>300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOSE');
    else if(data<-300)
        set(handles.edit2,'string','Tidak Terdeteksi/CLOSE');
    else
        set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
        delete(instrfind);
        %clear all;
        s = serial('COM15');

set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'F
lowControl','none','Terminator','CR/LF');
fopen(s);
fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;

        end
    end

else
    set(handles.edit2,'string','Wajah Terdeteksi/OPEN');
delete(instrfind);
%clear all;

```

```

s = serial('COM15');
set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'FlowControl','none','Terminator','CR/LF');
fopen(s);
fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;
    end
end

% --- Executes on button press in CAPTURE2.
function CAPTURE2_Callback(hObject, eventdata, handles)
% hObject    handle to CAPTURE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global vid rgb1 %gambar2
%image2=uigetfile('*.jpg');
%gambar2=imread(image2);
%axes(handles.axes2);imshow(gambar2);

rgb1 = getsnapshot(vid);
axes(handles.axes1);imshow(rgb1);

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%       str2double(get(hObject,'String')) returns contents of
edit8 as a double

% --- Executes during object creation, after setting all
properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in START.
function START_Callback(hObject, eventdata, handles)
% hObject    handle to START (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of START
global a
a=0;

% --- Executes on button press in INPUT_IMANGE.
function INPUT_IMANGE_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to INPUT_IMAGE (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
global D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17
D17 D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31 32 D3
D34 D35
image1=uigetfile('a.bmp');
E1=imread(image1);
I1 = rgb2gray(E1);
%axes(handles.axes3);imshow(D1);
citra1=im2bw(I1);
D1=bwarea(citra1);

image2=uigetfile('b.bmp');
E2=imread(image2);
I2 = rgb2gray(E2);
%axes(handles.axes3);imshow(D1);
citra2=im2bw(I2);
D2=bwarea(citra2);

image3=uigetfile('c.bmp');
E3=imread(image3);
I3 = rgb2gray(E3);
%axes(handles.axes3);imshow(D1);
citra3=im2bw(I3);
D3=bwarea(citra3);

image4=uigetfile('d.bmp');
E4=imread(image4);
I4 = rgb2gray(E4);
%axes(handles.axes3);imshow(D1);
citra4=im2bw(I4);
D4=bwarea(citra4);

image5=uigetfile('e.bmp');
E5=imread(image5);
I5 = rgb2gray(E5);
%axes(handles.axes3);imshow(D1);
citra5=im2bw(I5);
D5=bwarea(citra5);

image6=uigetfile('f.bmp');
E6=imread(image6);
I6 = rgb2gray(E6);
%axes(handles.axes3);imshow(D1);
citra6=im2bw(I6);
D6=bwarea(citra6);

image7=uigetfile('g.bmp');
E7=imread(image7);
I7 = rgb2gray(E7);
%axes(handles.axes3);imshow(D1);
citra7=im2bw(I7);
D7=bwarea(citra7);

```

```

image8=uigetfile('h.bmp');
E8=imread(image8);
I8 = rgb2gray(E8);
%axes(handles.axes3);imshow(D1);
citra8=im2bw(I8);
D8=bwarea(citra8);

image9=uigetfile('i.bmp');
E9=imread(image9);
I9 = rgb2gray(E9);
%axes(handles.axes3);imshow(D1);
citra9=im2bw(I9);
D9=bwarea(citra9);

image10=uigetfile('j.bmp');
E10=imread(image10);
I10 = rgb2gray(E10);
%axes(handles.axes3);imshow(D1);
citra10=im2bw(I10);
D10=bwarea(citra10);

%=====rgb=D3;
%I = rgb2gray(rgb);
%BW = im2bw(I);%(I2,graythresh(I2));
%axes(handles.axes1);imshow(BW);
%=====data1=rgb %bwarea(BW)
%=====set(handles.edit7,'string',data1);
%=====set(handles.edit3,'string',D8);

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%          str2double(get(hObject,'String')) returns contents of
edit9 as a double

% --- Executes during object creation, after setting all
properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```



```
end
```

## 2.SIMULATOR M- FILE

```
function varargout = simulationEta(varargin)
% SIMULATIONETA M-file for simulationEta.fig
%     SIMULATIONETA, by itself, creates a new SIMULATIONETA or
%     raises the existing
%     singleton*.
%
%     H = SIMULATIONETA returns the handle to a new SIMULATIONETA
%     or the handle to
%     the existing singleton*.
%
%     SIMULATIONETA('CALLBACK',hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in SIMULATIONETA.M with the given
%     input arguments.
%
%     SIMULATIONETA('Property','Value',...) creates a new
%     SIMULATIONETA or raises the
%     existing singleton*. Starting from the left, property
%     value pairs are
%     applied to the GUI before simulationEta_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to simulationEta_OpeningFcn
%     via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help simulationEta

% Last Modified by GUIDE v2.5 05-May-2008 22:38:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @simulationEta_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @simulationEta_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before simulationEta is made visible.
function simulationEta_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to simulationEta (see
VARARGIN)

% Choose default command line output for simulationEta
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes simulationEta wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = simulationEta_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text

```

```

%         str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject handle to Start (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
global D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17
D17 D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31 D32
D33 D34 D35

image1=uigetfile('a.bmp');
E1=imread(image1);
I1 = rgb2gray(E1);
axes(handles.axes1);imshow(E1);
citra1=im2bw(I1);
D1=bwarea(citra1);
set(handles.edit1,'string',D1);

image2=uigetfile('b.bmp');
E2=imread(image2);
I2 = rgb2gray(E2);
axes(handles.axes2);imshow(E2);
citra2=im2bw(I2);
D2=bwarea(citra2);
set(handles.edit2,'string',D2);

image3=uigetfile('c.bmp');
E3=imread(image3);
I3 = rgb2gray(E3);

```

```
axes(handles.axes3);imshow(E3);
citra3=im2bw(I3);
D3=bwarea(citra3);
set(handles.edit3,'string',D3);

image4=uigetfile('d.bmp');
E4=imread(image4);
I4 = rgb2gray(E4);
axes(handles.axes4);imshow(E4);
citra4=im2bw(I4);
D4=bwarea(citra4);
set(handles.edit4,'string',D4);

image5=uigetfile('e.bmp');
E5=imread(image5);
I5 = rgb2gray(E5);
axes(handles.axes5);imshow(E5);
citra5=im2bw(I5);
D5=bwarea(citra5);
set(handles.edit5,'string',D5);

image6=uigetfile('f.bmp');
E6=imread(image6);
I6 = rgb2gray(E6);
axes(handles.axes6);imshow(E6);
citra6=im2bw(I6);
D6=bwarea(citra6);
set(handles.edit7,'string',D6);

image7=uigetfile('g.bmp');
E7=imread(image7);
I7 = rgb2gray(E7);
axes(handles.axes7);imshow(E7);
citra7=im2bw(I7);
D7=bwarea(citra7);
set(handles.edit8,'string',D7);

image8=uigetfile('h.bmp');
E8=imread(image8);
I8 = rgb2gray(E8);
axes(handles.axes8);imshow(E8);
citra8=im2bw(I8);
D8=bwarea(citra8);
set(handles.edit9,'string',D8);

image9=uigetfile('i.bmp');
E9=imread(image9);
I9 = rgb2gray(E9);
axes(handles.axes9);imshow(E9);
citra9=im2bw(I9);
D9=bwarea(citra9);
set(handles.edit10,'string',D9);

image10=uigetfile('j.bmp');
E10=imread(image10);
```

```

I10 = rgb2gray(E10);
axes(handles.axes10);imshow(E10);
citra10=im2bw(I10);
D10=bwarea(citra10);
set(handles.edit11,'string',D10);

% --- Executes on button press in compailer.
function compailer_Callback(hObject, eventdata, handles)
% hObject    handle to compailer (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global D1 D2 %D3  D4 D5%rgb
%axes(handles.axes3); imshow(rgb);
data=D1-D2;
%set(handles.edit4,'string',data);
if(data>400)
    set(handles.edit6,'string','tdk sama');
    delete(instrfind);
%clear all;
s = serial('COM15');
set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'F
lowControl','none','Terminator','CR/LF');
fopen(s);
fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;
else if(data<-400)
    set(handles.edit6,'string','tdk tdk sama');
    delete(instrfind);
%clear all;
s = serial('COM15');
set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'F
lowControl','none','Terminator','CR/LF');
fopen(s);
fprintf(s,'%s','1');
%out = fscanf(s);
fclose(s);
delete(s);
clear s;
else
    set(handles.edit6,'string','sama');
%delete(instrfind);
%clear all;
%s = serial('COM7');
%set(s,'Baudrate',9600,'DataBits',8,'Parity','none','StopBits',1,'
FlowControl','none','Terminator','CR/LF');
%fopen(s);
%fprintf(s,'%s','a');
%fclose(s);
%delete(s);
%clear s;
    end
end
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%          str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%          str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of
edit8 as a double

% --- Executes during object creation, after setting all
properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of
edit9 as a double

% --- Executes during object creation, after setting all
properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%       str2double(get(hObject,'String')) returns contents of
edit10 as a double

% --- Executes during object creation, after setting all
properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%       str2double(get(hObject,'String')) returns contents of
edit11 as a double

% --- Executes during object creation, after setting all
properties.

```

```
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

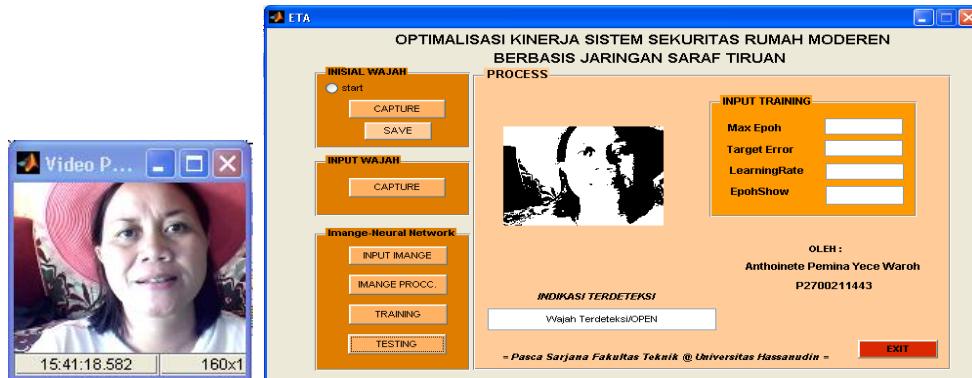
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

LAMPIRAN 2

Pengambilan citra wajah pada video dan diproses menghasilkan output indentikasih terdeteksi

a. Inisial Eta





The screenshot shows a video capture window on the left with a woman's face. The main application window, titled "ETA", displays the following content:

- INITIAL WAJAH**: start, CAPTURE, SAVE
- INPUT WAJAH**: CAPTURE
- Image-Neural Network**: INPUT IMAGE, IMAGE PROCC., TRAINING, TESTING
- INPUT TRAINING**: Max Epoch, Target Error, LearningRate, EpochShow
- INDIKASI TERDETEKSI**: Wajah Terdeteksi: OPEN
- OLEH**: Anthoinete Pemina Yece Waroh, P2700211443
- EXIT** button
- Footer**: - Pasca Sarjana Fakultas Teknik @ Universitas Hassanudin -

The screenshot shows a video capture window with a woman wearing sunglasses. The main application window displays:

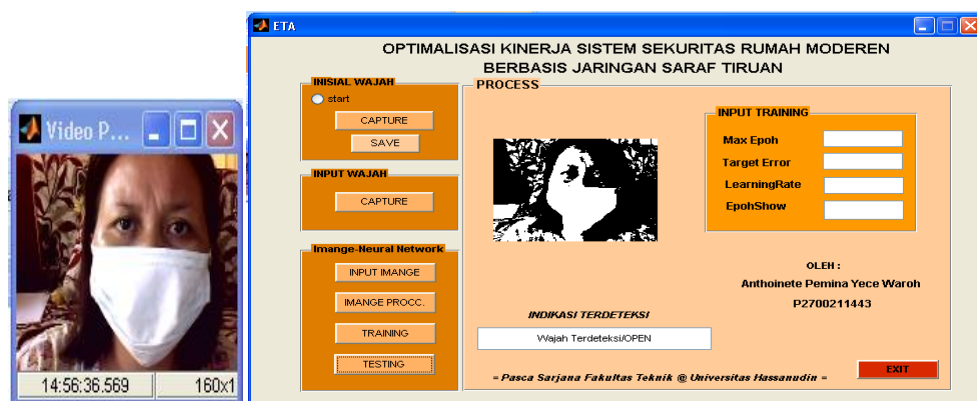
- INITIAL WAJAH**: start, CAPTURE, SAVE
- INPUT WAJAH**: CAPTURE
- Image-Neural Network**: INPUT IMAGE, IMAGE PROCC., TRAINING, TESTING
- INPUT TRAINING**: Max Epoch, Target Error, LearningRate, EpochShow
- INDIKASI TERDETEKSI**: Wajah Terdeteksi: OPEN
- OLEH**: Anthoinete Pemina Yece Waroh, P2700211443
- EXIT** button
- Footer**: - Pasca Sarjana Fakultas Teknik @ Universitas Hassanudin -

The screenshot shows a video capture window with a woman wearing sunglasses. The main application window displays:

- INITIAL WAJAH**: start, CAPTURE, SAVE
- INPUT WAJAH**: CAPTURE
- Image-Neural Network**: INPUT IMAGE, IMAGE PROCC., TRAINING, TESTING
- INPUT TRAINING**: Max Epoch, Target Error, LearningRate, EpochShow
- INDIKASI TERDETEKSI**: Wajah Terdeteksi: OPEN
- OLEH**: Anthoinete Pemina Yece Waroh, P2700211443
- EXIT** button
- Footer**: - Pasca Sarjana Fakultas Teknik @ Universitas Hassanudin -

The screenshot shows a video capture window with a woman wearing sunglasses. The main application window displays:

- INITIAL WAJAH**: start, CAPTURE, SAVE
- INPUT WAJAH**: CAPTURE
- Image-Neural Network**: INPUT IMAGE, IMAGE PROCC., TRAINING, TESTING
- INPUT TRAINING**: Max Epoch, Target Error, LearningRate, EpochShow
- INDIKASI TERDETEKSI**: Wajah Terdeteksi: OPEN
- OLEH**: Anthoinete Pemina Yece Waroh, P2700211443
- EXIT** button
- Footer**: - Pasca Sarjana Fakultas Teknik @ Universitas Hassanudin -







**b. Inisial Saul**







c. Inisial hoa



d. Inisial mami



### LAMPIRAN 3

#### Hasil pengujian wajah yang mirip

- Proses input



- Proses mengirim dan menerima gambar pada PC



- Tampilan out put wajah mirip yang tidak terdeteksi



- Inisial priskila tidak terdeteksi



- Inisial feliks tidak terdeteksi



- Inisial bela tidak terdeteksi



➤ Inisial rosye



➤ Inisial asrid



➤ Inisial indri



➤ Hasil Pelatihan data training

Learning Rate	: 0.1
Maksimum Epoch	: 1500
Target Error	: 0.001
Epok show	: 100
Erorr performansi	: 0.000789

➤ Bobot akhir koneksi dari lapisan input

Input	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>	Z <sub>4</sub>	Z <sub>5</sub>
X <sub>1</sub>	0.0075	-0.0008	-0.0008	-0.0622	-0.0065
X <sub>2</sub>	-0.0085	0.0020	0.0018	0.0017	0.0025
X <sub>3</sub>	0.0057	0.0013	0.0032	0.0013	0.0011
X <sub>4</sub>	0.0023	0.0016	0.0042	0.0031	0.0021
X <sub>5</sub>	0.0065	-0.0051	-0.0041	-0.0045	-0.0064
X <sub>6</sub>	0.0032	0.0028	0.0028	0.0029	0.0028
X <sub>7</sub>	0.0024	0.0019	0.0015	0.0019	0.0017
X <sub>8</sub>	0.0045	0.0035	0.0035	0.0035	0.0035
X <sub>9</sub>	0.0024	0.0052	0.0052	0.0052	0.0052
X <sub>10</sub>	0.0032	0.0029	0.0007	0.0010	-0.0065
X <sub>n</sub>	...	...	...	...	...

➤ Matriks Bobot koreksi dari lapisan output

Bobot	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
Z <sub>1</sub>	-0.2008	-0.1911	-0,1749
Z <sub>2</sub>	-0,0138	-0.0511	-0.0128
Z <sub>3</sub>	0.1852	-0.0959	-0.1586
Z <sub>4</sub>	-0.0611	-0.009	-0.0243
Z <sub>5</sub>	0.1581	0.1086	0.3112



- **Gambar Sistem mendeteksi wajah yang tidak sama pintu tertutup**



- **Gambar Sistem mendeteksi wajah yang sama pintu membuka**

