

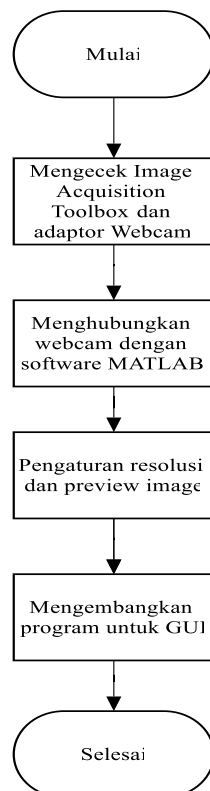
BAB IV

PROSES, HASIL, DAN PEMBAHASAN

A. Proses

1. Akuisisi Data

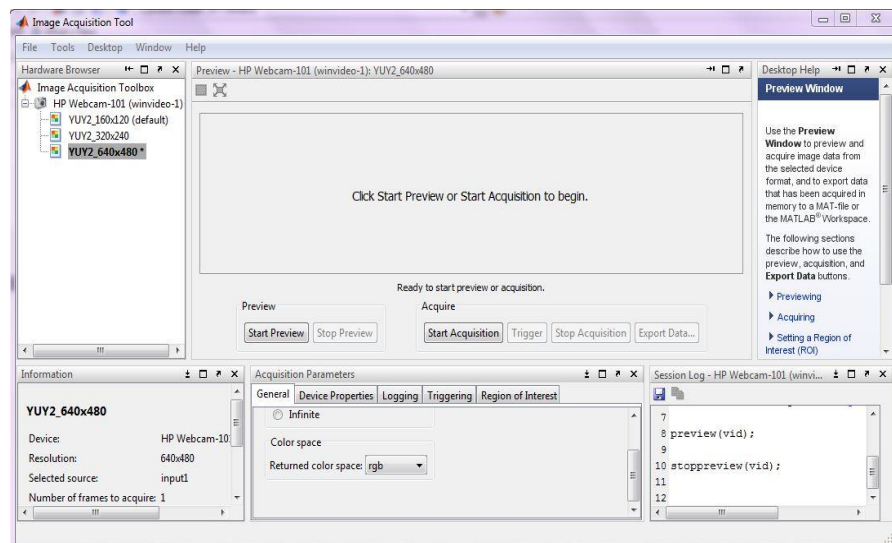
Tahap akuisisi data merupakan tahapan pengambilan citra yang dilakukan menggunakan kamera atau membuka citra yang ada pada folder di PC. Sebelum melakukan proses akuisisi data, *webcam* dihubungkan ke *software* MATLAB terlebih dahulu dengan langkah yang tertera pada gambar 26.



Gambar 1. Flowchart menghubungkan webcam dengan software MATLAB

Webcam dan *software* MATLAB dihubungkan dengan menggunakan *Image Acquisition Toolbox* dan adaptor *webcam* bernama *winvideo*. Hal tersebut bisa diketahui dengan cara mengetik sintaks *imaqhwinfo* pada *Command Window*.

Kemudian melakukan pengaturan resolusi dan *preview image* dengan mengetik sintaks *imaqtool* (MathWorks, 2012) pada *Command Window* sampai muncul tampilan seperti pada gambar 27. Apabila webcam dan software MATLAB telah terhubung, maka dilanjutkan dengan mengembangkan program seperti pada tabel 8.



Gambar 2. Tampilan *imaqtool*

Tabel 1. Program Akuisisi Data

Program Akuisisi Data		
No.	Program	Keterangan
1.	<pre>global VidObj VidObj = videoinput('winvideo',1,'YUY2_640x480'); %inisialisasi webcam handles.VidObj = VidObj; axes(handles.axes1) hImage= image(zeros(300,300,3),'Parent',handles.axes1); %tampilan capture dan ukuran preview(VidObj,hImage);</pre>	Tombol “Kamera On”
2.	<pre>global VidObj gambar = getsnapshot(VidObj); %capture image</pre>	Tombol “Capture”

	<pre> gambarRGB = ycbcr2rgb(gambar);%konversi citra ke RGB imshow(gambarRGB); handles.gambarRGB = gambarRGB; axes(handles.axes1) guidata(hObject, handles); </pre>	
3.	<pre> [filename,pathname] = uigetfile('*.jpg');%membuka file if ~isequal(filename,0) gambarRGB= imread(fullfile(pathname,fi lename)); axes(handles.axes1) imshow(gambarRGB) title('Citra RGB') else return end handles.gambarRGB = gambarRGB; guidata(hObject, handles); </pre>	Tombol “Buka”

2. Pre-processing

Pada tahap *pre-processing* akan dilakukan 2 proses pengolahan citra yakni *cropping* dan *resizing*.

a. Cropping

Proses *cropping* bertujuan untuk memotong citra yang akan digunakan dan membuang citra yang tidak digunakan. Citra akan dipotong menjadi bentuk persegi panjang dengan fungsi *imrect* (MathWorks, 2012). Ukuran serta posisi citra yang akan dipotong akan disesuaikan menurut citra itu sendiri karena ukuran mata pada setiap orang berbeda. Setelah penyesuaian posisi, maka citra akan di potong dengan fungsi *imcrop* (MathWorks, 2012). Adapun program pada tahap proses *cropping* terdapat pada tabel 9.

Tabel 2. Program *Cropping*

Program <i>Cropping</i>		
No	Program	Keterangan
1.	<pre> gambarRGB = handles.gambarRGB; [m,n,~] = size(gambarRGB); axes(handles.axes2) imshow(gambarRGB); h = imrect(gca,[n/2 m/2 0.2*n 0.2*m]);%membentuk persegi panjang wait(h); mask = createMask(h); [B,~] = bwboundaries(mask,'noholes'); k = 1:length(B) boundary = B{k}; axes(handles.axes1) imshow(gambarRGB); hold on plot(boundary(:,2), boundary(:,1), 'b', 'LineWidth', 2) hold off R = gambarRGB(:,:,1); G = gambarRGB(:,:,2); B = gambarRGB(:,:,3); R(~mask) = 0; G(~mask) = 0; B(~mask) = 0; RGB = cat(3,R,G,B);[row,col] = find(mask==1); gbrcrop = imcrop(RGB,[min(col) min(row) max(col)-min(col) max(row)-min(row)]);%memotong citra handles.gbrcrop = gbrcrop; axes(handles.axes2) imshow(gbrcrop); guidata(hObject, handles); </pre>	Tombol “Crop”
2.	<pre> gbrcrop = handles.gbrcrop; [name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image'); %menyimpan data pada folder if ~isequal (name_file_save, 0) imwrite(gbrcrop,fullfile(path_save , name_file_save)); else return end </pre>	Tombol “Simpan Hasil Crop”

b. *Resizing*

Citra hasil *cropping* memiliki besar ukuran piksel yang berbeda, sedangkan pada tahap ekstraksi ciri, citra yang digunakan harus memiliki besar ukuran piksel yang sama. Pada proses *resizing* ini lah besar ukuran piksel citra akan dirubah dan dibuat menjadi sama antara satu citra dengan citra lainnya dengan menggunakan fungsi *imresize* (MathWorks, 2012). Besar ukuran piksel citra ditentukan dengan mencoba satu persatu besar ukuran yang sesuai agar citra terhindar dari *blurring*.

Setelah mencoba satu persatu, ditemukan ukuran piksel yang tepat sebesar 255 x 144 *pixels* untuk diterapkan pada semua citra yang digunakan. Tabel 10 merupakan program yang digunakan pada tahap *resizing*.

Tabel 3. Program Tahap *Resizing*

Program Tahap <i>Resizing</i>		
No.	Program	Keterangan
1.	<pre> gbrcrop = handles.gbrcrop; rsz = imresize(gbrcrop,[144 255]);%resize dengan ukuran 255 x 144 axes(handles.axes7) imshow(rsz); handles.rsz = rsz; guidata(hObject, handles); </pre>	Tombol “Resize”
2.	<pre> rsz = handles.rsz; [name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image'); %menyimpan data pada folder if ~isequal (name_file_save, 0) imwrite (rsz,fullfile(path_save, name_file_save)); else return end </pre>	Tombol “Simpan Hasil Resize”

3. Tahap Ekstrasi Ciri

Pada tahap ekstrasi ciri akan dicari nilai ciri yang dipakai yakni *metric* dan *eccentricity* dari citra latih dan uji. Tahap ekstrasi ciri dilakukan dengan mengembangkan program pada *software* MATLAB. Tabel 11 merupakan program pada tahap ekstrasi ciri.

Tabel 4. Program Tahap Ekstrasi Ciri

Program Tahap Ekstrasi Ciri		
No.	Program	Keterangan
1.	<pre>gambarRGB = handles.gambarRGB; J = gambarRGB(:,:,1); K = im2bw(J,.6); L = imcomplement(K); str = strel('disk',5); M = imclose(L,str); N = imfill(M,'holes'); O = bwareaopen(N,2200); stats = regionprops(O,'Area','Perime ter','Eccentricity'); area = stats.Area; perimeter = stats.Perimeter; metric = 4*pi*area/(perimeter^2); eccentricity = stats.Eccentricity; ciri = cell(2,2); ciri{1,1} = 'Metric'; ciri{2,1} = 'Eccentricity'; ciri{1,2} = num2str(metric); ciri{2,2} = num2str(eccentricity); handles.ciri = ciri; guidata(hObject, handles) set(handles.uitable1,'Data', ciri,'RowName',1:2)</pre>	Tombol “Ekstrasi Ciri”

4. Tahap Klasifikasi

a. Pelatihan Jaringan

Setelah menemukan masing – masing nilai dari ciri yang digunakan, citra akan melalui proses pelatihan jaringan agar menemukan suatu pola. Pelatihan jaringan merupakan proses pembelajaran menggunakan Jaringan Syaraf Tiruan dengan model *supervised learning* dan metode *backpropagation* guna menemukan suatu pola dengan bobot terbaik. Pada proses pelatihan jaringan, harus diperhatikan tentang penggunaan arsitektur dan parameter jaringan. Arsitektur jaringan serta parameter jaringan yang digunakan sangat mempengaruhi hasil dari proses pelatihan jaringan. Arsitektur jaringan dan parameter jaringan dapat dicari dengan mengubah – ubah nilai *hidden layer*, *epoch*, dan *mse*. Hasil dari proses pelatihan jaringan yang berbentuk *net* akan disimpan dan dipanggil kembali saat pengujian jaringan. Tabel 12 merupakan program pelatihan jaringan yang digunakan.

Tabel 5. Program Pelatihan Jaringan

Program Pelatihan Jaringan
<pre> clc; clear; close all; image_folder = 'latih'; filenames = dir(fullfile(image_folder, '*.jpg')); total_images = numel(filenames); area = zeros(1,total_images); perimeter = zeros(1,total_images); metric = zeros(1,total_images); eccentricity = zeros(1,total_images); for n = 1:total_images full_name= fullfile(image_folder, filenames(n).name); I = imread(full_name); J = I(:,:,1); K = im2bw(J,.6); L = imcomplement(K); str = strel('disk',5); M = imclose(L,str); N = imfill(M,'holes'); O = bwareaopen(N,2200); </pre>

```

stats =
regionprops(0, 'Area', 'Perimeter', 'Eccentricity');
area(n) = stats.Area;
perimeter(n) = stats.Perimeter;
metric(n) = 4*pi*area(n)/(perimeter(n)^2);
eccentricity(n) = stats.Eccentricity;
end

input = [metric;eccentricity];
target = zeros(1,68);
target(:,1:34) = 1; %data memiliki ketinggian
dioptri ≥ 0,25D
target(:,35:68) = 0; %data memiliki ketinggian
dioptri = 0D

net = newff(input,target,[20
10],{'logsig','logsig'},'trainlm');
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;
net = train(net,input,target);
output = round(sim(net,input));
save netlatih %hasil pelatihan jaringan

[m,n] = find(output==target);
akurasi = sum(m)/total_images*100

```

b. Pengujian Jaringan

Pada pengujian jaringan, hasil pelatihan jaringan yang berupa *net* akan dipanggil kembali dengan menggunakan fungsi *load* untuk melakukan pengecekan terhadap data baru. Data baru tersebut akan mengenali pola yang sesuai dengan kategorinya. Pengujian jaringan dilakukan langsung pada GUI yang telah disiapkan.

5. Unjuk Kerja Keseluruhan

Pada tahap ini akan ditampilkan kemampuan sistem dalam mengidentifikasi mata miopia dan mata tidak miopia melalui GUI pada *software* MATLAB R2012a. Tabel 13 merupakan tabel program yang digunakan untuk melakukan pengujian unjuk kerja.

Tabel 6. Program Pengujian Unjuk Kerja

Program Pengujian Unjuk Kerja		
No.	Program	Keterangan
1.	<pre> rsz = handles.rsz; J = rsz(:,: ,1); K = im2bw(J, .6); L = imcomplement(K); str = strel('disk',5); M = imclose(L,str); N = imfill(M, 'holes'); O = bwareaopen(N,2200); stats = regionprops(O, 'Area', 'Perimete r', 'Eccentricity'); area = stats.Area; perimeter = stats.Perimeter; metric = 4*pi*area/(perimeter^2); eccentricity = stats.Eccentricity; input = [metric;eccentricity]; load netlatih output = round(sim(net,input)); if output == 1; hsl = 'Miopia'; else hsl = 'Tidak Miopia'; end set(handles.edit1, 'String', hsl); </pre>	Tombol “Hasil”

B. Hasil Pengujian

1. Akuisisi Data

Gambar 28 merupakan hasil GUI pada tahap akuisisi data dan tabel 14 merupakan tabel hasil akuisisi data 6 dari 40 citra uji. Pengambilan citra tidak sampai pada mengetahui tingkat dioptri dari gangguan miopia yang diderita, namun hanya sebatas kategori miopia atau tidak miopia.



Gambar 3. Hasil pengujian GUI Tahap Akusisi Data

Tabel 7. Hasil Pengujian Tahap Akusisi Data

No.	Nama Data	Data
1.	M17	
2.	M18	

3.	M19	
4.	N15	
5.	N17	
6.	N19	

2. *Pre – Processing*

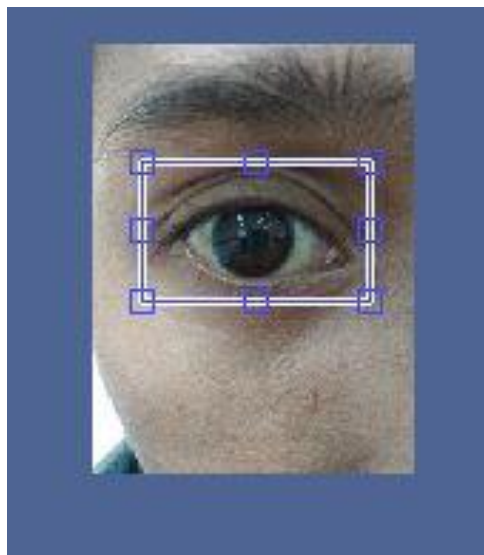
Gambar 29 merupakan hasil pengujian GUI *pre – processing*.



Gambar 4. Hasil pengujian GUI tahap *pre - processing*

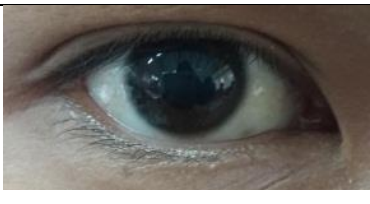
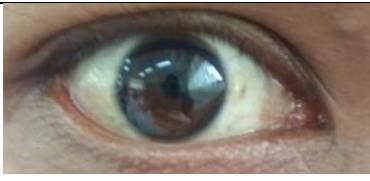
a. *Cropping*

Gambar 30 merupakan hasil pengujian GUI *cropping* dan tabel 15 merupakan tabel hasil tahap *cropping* 6 dari 40 data uji.



Gambar 5. Hasil Pengujian GUI tahap *cropping*

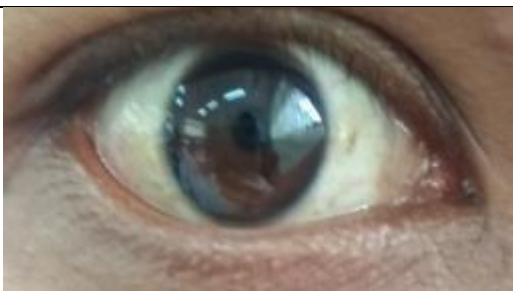
Tabel 8. Hasil Pengujian Tahap *Cropping*

No.	Nama Data	Hasil <i>Cropping</i>
1.	M17	
2.	M18	
3.	M19	
4.	N15	
5.	N17	
6.	N19	

b. *Resizing*

Tabel 16 merupakan tabel hasil pengujian tahap *resizing* 6 dari 40 data uji.

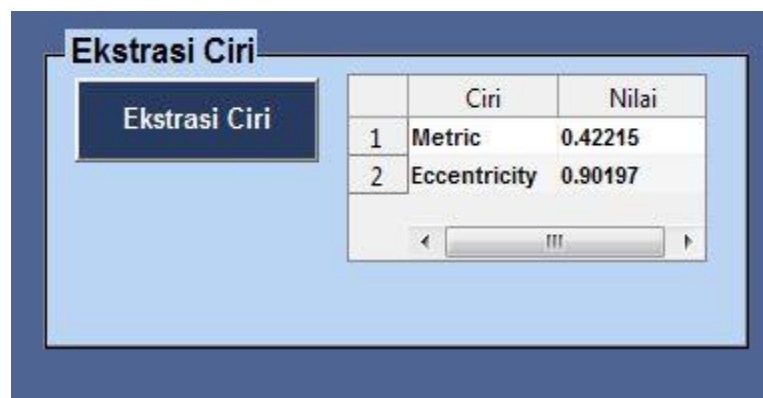
Tabel 9. Hasil Pengujian Tahap *Resizing*

No.	Nama Data	Hasil <i>Resizing</i>
1.	M17	
2.	M18	
3.	M19	
4.	N15	
5.	N17	

6.	N19	
----	-----	--

3. Ekstrasi Ciri

Gambar 31 merupakan hasil pengujian GUI pada tahap ekstrasi ciri dan tabel 17 merupakan tabel hasil pengujian tahap ekstrasi ciri.



Gambar 6. Hasil Pengujian GUI tahap ekstrasi ciri

Tabel 10. Hasil Pengujian Tahap Ekstrasi Ciri

No.	Nama Data	Nilai <i>Metric</i>	Nilai <i>Eccentricity</i>
1.	M1	0.58413	0.80978
2,	M2	0.73193	0.82529
3,	M3	0.3239	0.89387
4,	M4	0.26225	0.91279
5,	M5	0.73193	0.82529
6,	M6	0.46061	0.90631
7.	M7	0.72356	0.8258

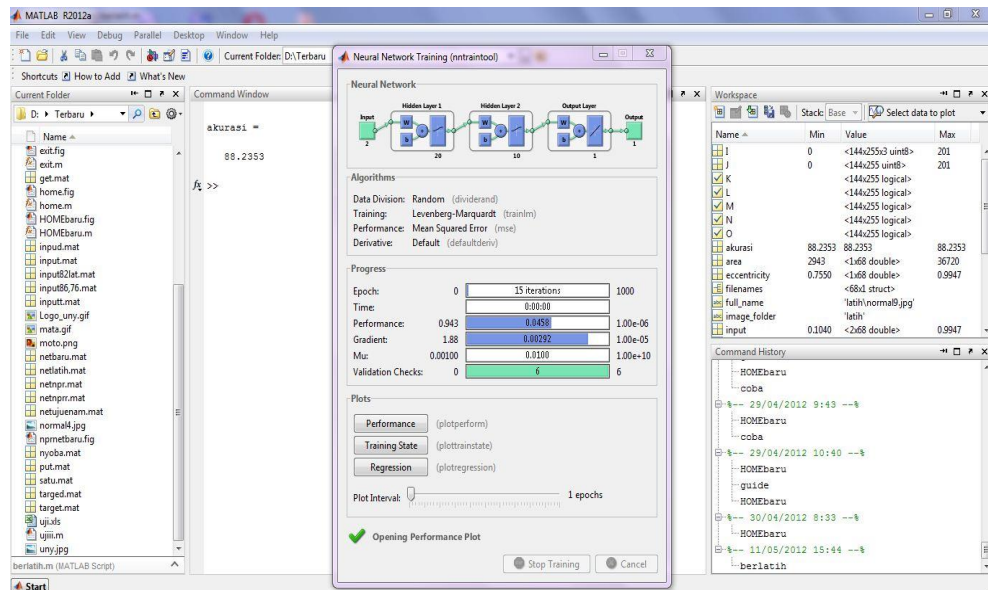
8.	M8	0.31565	0.8195
9.	M9	0.21319	0.845
10.	M10	0.4579	0.86402
11.	M11	0.33761	0.92255
12.	M12	0.33647	0.90072
13.	M13	0.29701	0.7832
14.	M14	0.20899	0.92949
15.	M15	0.42042	0.88428
16.	M16	0.73193	0.82529
17.	M17	0.73811	0.82533
18.	M18	0.24988	0.82083
19.	M19	0.63541	0.82865
20.	M20	0.62169	0.83328
21.	N1	0.74454	0.82458
22.	N2	0.73193	0.82529
23.	N3	0.71235	0.82464
24.	N4	0.62455	0.81151
25.	N5	0.7422	0.8253
26.	N6	0.28711	0.94361
27.	N7	0.73193	0.82529
28.	N8	0.5538	0.86663
29.	N9	0.6035	0.74515
30.	N10	0.57734	0.79258
31.	N11	0.4288	0.84041
32.	N12	0.47115	0.83564
33.	N13	0.41821	0.92865
34.	N14	0.59379	0.79633
35.	N15	0.17525	0.86851
36.	N16	0.15729	0.91044
37.	N17	0.30334	0.927

38.	N18	0.34573	0.90315
39.	N19	0.60491	0.83432
40.	N20	0.36335	0.85924

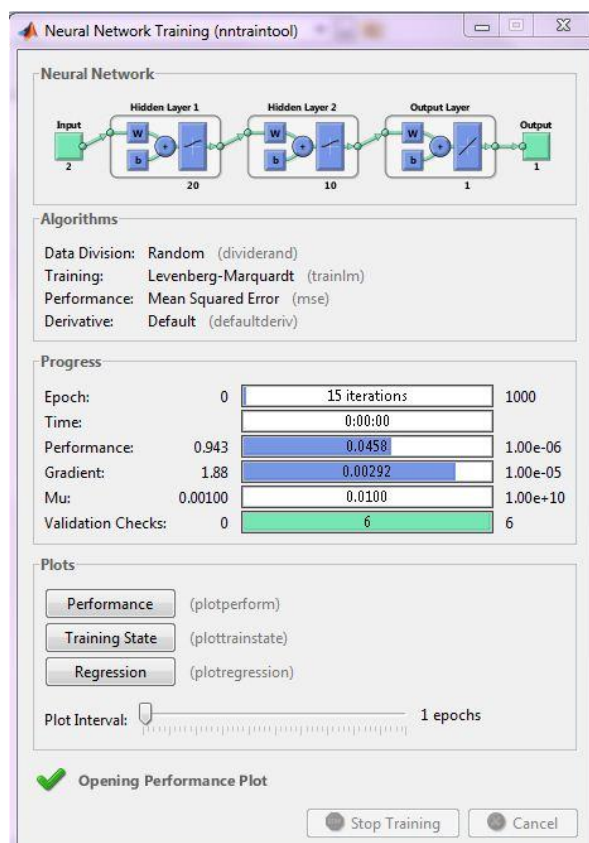
4. Klasifikasi

a. Pelatihan jaringan

Gambar 32 menunjukkan hasil akurasi pelatihan jaringan, gambar 33 merupakan gambar arsitektur dan parameter pelatihan jaringan. Tabel 18 merupakan tabel hasil pelatihan dari 68 data citra latih, dimana pada program pelatihan jaringan tertulis bahwa citra responden 1 sampai 34 merupakan citra data miopia dengan rincian memiliki tingkat miopia $\geq 0,25D$, nilai targetnya adalah 1. Sedangkan citra responden 35 sampai 68 merupakan citra data tidak miopia dengan rincian memiliki tingkat miopia = $0D$ nilai targetnya adalah 0. Selanjutnya, jika nilai *output* yang dihasilkan sama dengan nilai target maka hasil deteksi adalah sesuai. Sebaliknya, jika nilai *output* tidak sama dengan nilai target maka hasil deteksi tidak sesuai.



Gambar 7. Hasil proses pelatihan jaringan



Gambar 8. Arsitektur dan Parameter Pelatihan Jaringan

Tabel 11. Hasil Pelatihan Jaringan

No.	Nama Data Latih	Target	<i>Output</i>	Keterangan
1.	Data 1	1	1	Sesuai
2.	Data 2	1	1	Sesuai
3.	Data 3	1	1	Sesuai
4.	Data 4	1	1	Sesuai
5.	Data 5	1	1	Sesuai
6.	Data 6	1	1	Sesuai
7.	Data 7	1	0	Tidak Sesuai
8.	Data 8	1	1	Sesuai
9.	Data 9	1	1	Sesuai
10.	Data 10	1	0	Tidak Sesuai
11.	Data 11	1	1	Sesuai
12.	Data 12	1	1	Sesuai
13.	Data 13	1	1	Sesuai
14.	Data 14	1	1	Sesuai
15.	Data 15	1	1	Sesuai
16.	Data 16	1	1	Sesuai
17.	Data 17	1	1	Sesuai
18.	Data 18	1	1	Sesuai
19.	Data 19	1	1	Sesuai
20.	Data 20	1	1	Sesuai
21.	Data 21	1	1	Sesuai
22.	Data 22	1	1	Sesuai
23.	Data 23	1	1	Sesuai
24.	Data 24	1	1	Sesuai
25.	Data 25	1	1	Sesuai
26.	Data 26	1	1	Sesuai
27.	Data 27	1	1	Sesuai

28.	Data 28	1	1	Sesuai
29.	Data 29	1	1	Sesuai
30.	Data 30	1	1	Sesuai
31.	Data 31	1	1	Sesuai
32.	Data 32	1	1	Sesuai
33.	Data 33	1	1	Sesuai
34.	Data 34	1	1	Sesuai
35.	Data 35	0	0	Sesuai
36.	Data 36	0	0	Sesuai
37.	Data 37	0	0	Sesuai
38.	Data 38	0	0	Sesuai
39.	Data 39	0	0	Sesuai
40.	Data 40	0	0	Sesuai
41.	Data 41	0	1	Tidak Sesuai
42.	Data 42	0	1	Tidak Sesuai
43.	Data 43	0	0	Sesuai
44.	Data 44	0	0	Sesuai
45.	Data 45	0	1	Tidak Sesuai
46.	Data 46	0	0	Sesuai
47.	Data 47	0	1	Tidak Sesuai
48.	Data 48	0	0	Sesuai
49.	Data 49	0	0	Sesuai
50.	Data 50	0	0	Sesuai
51.	Data 51	0	0	Sesuai
52.	Data 52	0	0	Sesuai
53.	Data 53	0	1	Tidak Sesuai
54.	Data 54	0	0	Sesuai
55.	Data 55	0	0	Sesuai
56.	Data 56	0	0	Sesuai
57.	Data 57	0	0	Sesuai

58.	Data 58	0	0	Sesuai
59.	Data 59	0	0	Sesuai
60.	Data 60	0	0	Sesuai
61.	Data 61	0	0	Sesuai
62.	Data 62	0	0	Sesuai
63.	Data 63	0	0	Sesuai
64.	Data 64	0	0	Sesuai
65.	Data 65	0	1	Tidak Sesuai
66.	Data 66	0	0	Sesuai
67.	Data 67	0	0	Sesuai
68.	Data 68	0	0	Sesuai

Berdasarkan hasil tabel pelatihan jaringan disimpulkan bahwa terdapat 60 data yang outputnya sesuai dengan target yang diharapkan. Maka dapat diperoleh hasil akurasi dari pelatihan jaringan dari 68 data latih ialah sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah benar}}{\text{Jumlah data}} \times 100\% \\
 &= \frac{60}{68} \times 100\% \\
 &= 88,235\%
 \end{aligned}$$

b. Pengujian jaringan

Tabel 19 merupakan hasil pengujian terhadap data uji dari 40 citra uji. Berdasarkan tabel hasil tersebut disimpulkan bahwa terdapat 28 data uji sesuai dengan hasil medis responden yang sesungguhnya. Maka didapatkan hasil akurasi pengujian sebesar :

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

$$= \frac{28}{40} \times 100\%$$

$$= 70\%$$

5. Unjuk Kerja Keseluruhan

Gambar 34 merupakan tampilan GUI pada saat hasil deteksi berupa keputusan “Miopia”, gambar 35 merupakan tampilan GUI pada saat hasil deteksi berupa keputusan “Tidak Miopia”, serta tabel 19 merupakan tabel hasil unjuk kerja keseluruhan.



Gambar 9. Tampilan GUI dengan hasil identifikasi “Miopia”



Gambar 10. Tampilan GUI hasil identifikasi “Tidak Miopia”

Tabel 12. Pengujian Unjuk Kerja Keseluruhan

No.	Nama Data	Hasil deteksi sistem	Hasil Medis	Keterangan
1.	M1	Tidak Miopia	Miopia	Salah
2.	M2	Miopia	Miopia	Benar
3.	M3	Miopia	Miopia	Benar
4.	M4	Miopia	Miopia	Benar
5.	M5	Miopia	Miopia	Benar
6.	M6	Miopia	Miopia	Benar
7.	M7	Tidak Miopia	Miopia	Salah
8.	M8	Miopia	Miopia	Benar
9.	M9	Tidak Miopia	Miopia	Salah
10.	M10	Tidak Miopia	Miopia	Salah
11.	M11	Miopia	Miopia	Benar
12.	M12	Miopia	Miopia	Benar
13.	M13	Miopia	Miopia	Benar
14.	M14	Miopia	Miopia	Benar
15.	M15	Miopia	Miopia	Benar

16.	M16	Miopia	Miopia	Benar
17.	M17	Miopia	Miopia	Benar
18.	M18	Tidak Miopia	Miopia	Salah
19.	M19	Miopia	Miopia	Benar
20.	M20	Miopia	Miopia	Benar
21.	N1	Miopia	Tidak Miopia	Salah
22.	N2	Tidak Miopia	Tidak Miopia	Benar
23.	N3	Miopia	Tidak Miopia	Salah
24.	N4	Tidak Miopia	Tidak Miopia	Benar
25.	N5	Miopia	Tidak Miopia	Salah
26.	N6	Tidak Miopia	Tidak Miopia	Benar
27.	N7	Miopia	Tidak Miopia	Salah
28.	N8	Tidak Miopia	Tidak Miopia	Benar
29.	N9	Tidak Miopia	Tidak Miopia	Benar
30.	N10	Tidak Miopia	Tidak Miopia	Benar
31.	N11	Tidak Miopia	Tidak Miopia	Benar
32.	N12	Tidak Miopia	Tidak Miopia	Benar
33.	N13	Tidak Miopia	Tidak Miopia	Benar
34.	N14	Tidak Miopia	Tidak Miopia	Benar
35.	N15	Tidak Miopia	Tidak Miopia	Benar
36.	N16	Tidak Miopia	Tidak Miopia	Benar
37.	N17	Miopia	Tidak Miopia	Salah
38.	N18	Miopia	Tidak Miopia	Salah
39.	N19	Miopia	Tidak Miopia	Salah
40.	N20	Tidak Miopia	Tidak Miopia	Benar

C. Pembahasan

Proses pengembangan sistem Pengolahan Citra Digital Untuk Mendeteksi Gangguan Miopia Menggunakan Jaringan Syaraf Tiruan terdiri dari beberapa

tahap seperti akuisisi data, *pre – processing*, ekstraksi ciri, klasifikasi, dan identifikasi hasil.

Berikut pembahasan dari masing – masing tahapan pengembangan sistem Pengolahan Citra Digital Untuk Mendeteksi Gangguan Miopia Menggunakan Jaringan Syaraf Tiruan :

1. Akuisisi data

Setelah dilakukan pengujian pada tahap akuisisi data dihasilkan seperti pada tabel 14. Sistem telah mampu mengambil citra mata melalui *webcam* secara *realtime* maupun membuka citra mata pada folder yang sudah ada di PC. Citra diambil dengan *webcam* beresolusi 640 x 480 *pixels*. Resolusi webcam akan mempengaruhi kualitas citra yang dihasilkan. Citra yang diambil merupakan citra mata kanan atau mata kiri. Jumlah citra mata akan mempengaruhi hasil pembelajaran atau pelatihan jaringan. Semakin banyak citra yang digunakan, maka akan mempermudah jaringan dalam mengenali pola. Pada pengembangan sistem ini citra yang digunakan sebanyak 68 citra mata dengan rincian 34 citra mata miopia dan 34 citra mata tidak miopia untuk proses pembelajaran atau pelatihan jaringan, serta 40 citra mata dengan rincian 20 citra mata miopia dan 20 citra mata tidak miopia untuk pengujian.

2. *Pre – processing*

Pada tahap ini dilakukan pemotongan citra (*cropping*) dan perubahan ukuran piksel citra (*resizing*).

a. *Cropping*

Setelah proses pengujian pada tahap *cropping* atau pemotongan citra, citra telah berhasil dipotong dengan fungsi *imcrop* berbentuk persegi panjang menggunakan fungsi *imrect* sesuai dengan *software* MATLAB R2012a. Hasil *cropping* terdapat pada tabel 15. Jumlah citra mata yang berhasil dipotong ialah sebanyak 68 citra mata dengan rincian 34 citra mata miopia dan 34 citra mata tidak miopia untuk proses pembelajaran atau pelatihan jaringan, serta 40 citra mata dengan rincian 20 citra mata miopia dan 20 citra mata tidak miopia untuk pengujian.

b. Resizing

Setelah proses pengujian pada tahap *resizing*, citra mata berhasil diresize atau dirubah ukuran pikselnya menggunakan fungsi *imresize* menjadi 255 x 144 *pixels*. Hasil pada tahap *resizing* terdapat pada tabel 16. Jumlah citra mata yang berhasil *resize* ialah sebanyak 68 citra mata dengan rincian 34 citra mata miopia dan 34 citra mata tidak miopia untuk proses pembelajaran atau pelatihan jaringan, serta 40 citra mata dengan rincian 20 citra mata miopia dan 20 citra mata tidak miopia untuk pengujian.

3. Ekstrasi Ciri

Setelah proses pengujian pada tahap ekstrasi ciri dihasilkan nilai *metric* dan *eccentricity* dari 40 citra uji seperti pada tabel 17. Sistem berhasil menentukan nilai ciri yang dipakai yakni *metric* dan *eccentricity*. Ciri *metric* dan *eccentricity* merupakan contoh dari ciri bentuk. Selain ciri bentuk, juga terdapat ciri lain seperti ciri tekstur, ciri statistik, ciri warna, dan lainnya. Penggunaan dan jumlah ciri yang diekstrasi harus sesuai dengan objek yang akan dideteksi karena akan

mempengaruhi hasil deteksi. Proses ekstrasi ciri pada proyek akhir ini dilakukan dengan menggunakan program pada tabel 10. Penjelasan program yang dipakai dalam penentuan nilai *metric* dan *eccentricity* terdapat pada tabel 20.

Tabel 13. Penjelasan program ekstrasi ciri

Program	Penjelasan
<code>stats=regionprops(O,'Area','Perimeter','Eccentricity');</code>	Untuk menghitung wilayah
<code>area(n) = stats.Area;</code>	Untuk menentukan nilai Area
<code>perimeter(n) = stats.Perimeter;</code>	Untuk menentukan nilai Perimeter
<code>eccentricity(n) = stats.Eccentricity;</code>	Untuk menentukan nilai eccentricity
<code>Metric(n) = 4*pi*area(n) / perimeter(n) ^2;</code>	Untuk menentukan nilai metric
Nilai O = BW	Didapatkan dari fungsi <code>stats=regionprops(BW,properties)</code> .

4. Klasifikasi

Tahap klasifikasi dilakukan menggunakan Jaringan Syaraf Tiruan dengan model *supervised learning* dan metode *backpropagation*. *Supervised learning* merupakan proses pembelajaran atau pelatihan jaringan terawasi dengan input dan target yang telah ditentukan. Setelah melewati proses pelatihan jaringan, maka akan dilakukan proses pengujian jaringan.

a. Pelatihan Jaringan

Pelatihan jaringan dilakukan untuk melakukan pembelajaran terhadap 68 data latih yang terdiri dari 34 citra mata miopia dengan tingkat dioptri $\geq 0,25D$ dan 34 citra mata tidak miopia dengan tingkat dioptri $= 0D$. Pelatihan jaringan dilakukan dengan menggunakan Jaringan Syaraf Tiruan dengan metode

Backpropagation Levenberg-Marquardt. Proses pelatihan ini dilakukan agar citra latih dapat mengenali suatu pola sesuai kategorinya, yakni miopia atau tidak miopia. Pola tersebut tersusun berdasarkan arsitektur dan parameter jaringan yang digunakan. Arsitektur jaringan yang dipakai terdiri dari 2 masukan yang berupa nilai ciri *metric* dan *eccentricity*, 2 *hidden layer* dengan rincian 20 *neurons* pada *hidden layer* 1, dan 10 *neurons* pada *hidden layer* 2, serta 1 keluaran yang berupa keputusan hasil deteksi. Selain terdapat arsitektur jaringan, juga terdapat parameter lain yang telah diinputkan seperti *max epoch* sebesar 1000, dan *max goal* sebesar $1e-6$. Fungsi aktivasi berupa *sigmoid biner (logsig)* karena pada unit *output* hanya dirancang untuk menampilkan salah satu dari dua keputusan, yakni miopia atau tidak miopia. Penentuan arsitektur serta parameter jaringan dilakukan dengan cara mencari satu – persatu arsitektur yang tepat sampai menghasilkan akurasi terbaik. Hasil pelatihan jaringan dapat dilihat pada gambar 18.

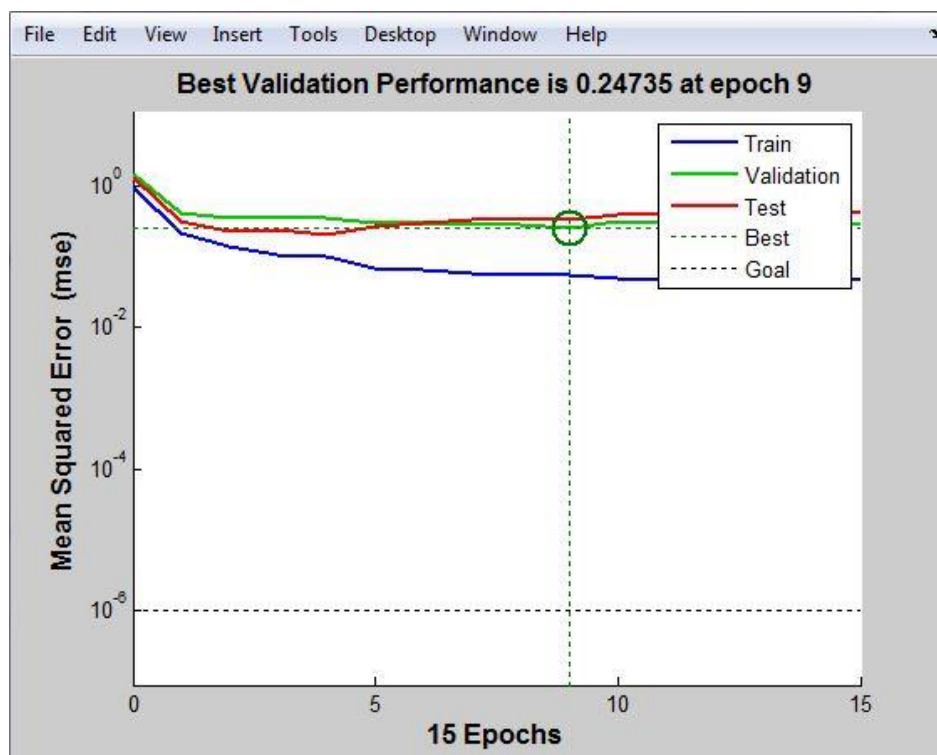
Pada gambar 33, ditampilkan gambar arsitektur jaringan yang dipakai serta parameter – parameter lainnya seperti, *epoch* yang berarti perulangan pembelajaran, *time* yang berarti waktu yang ditempuh untuk melakukan proses pembelajaran, *performance* sebagai gambaran kualitas pembelajaran, *gradient* yang menyatakan kemiringan antara iterasi satu dengan iterasi lainnya, serta *val fail* untuk mengetahui jumlah kesalahan yang terjadi saat proses validasi (Maulidin & Assafat, 2014) .

Berdasarkan gambar 33, proses pembelajaran terhenti pada *epoch* atau iterasi ke 15. Hal ini dikarenakan *val fail* telah lebih dulu melebihi angka *max fail* yang diatur dengan angka default. Pembelajaran yang terhenti pada perulangan

atau iterasi ke 15 menghasilkan nilai *performance* sebesar 0,0458. Apabila nilai *performance* mendekati nilai 0, maka kualitas *performance* akan semakin baik. Selain itu, dihasilkan pula nilai *gradient* sebesar 0,00292, dimana angka tersebut menyatakan kemiringan dari satu iterasi ke iterasi lainnya.

Selain menghasilkan arsitektur dan parameter – parameter lain, dihasilkan pula grafik – grafik dengan penjelasan sebagai berikut :

1) *Plot Performance*

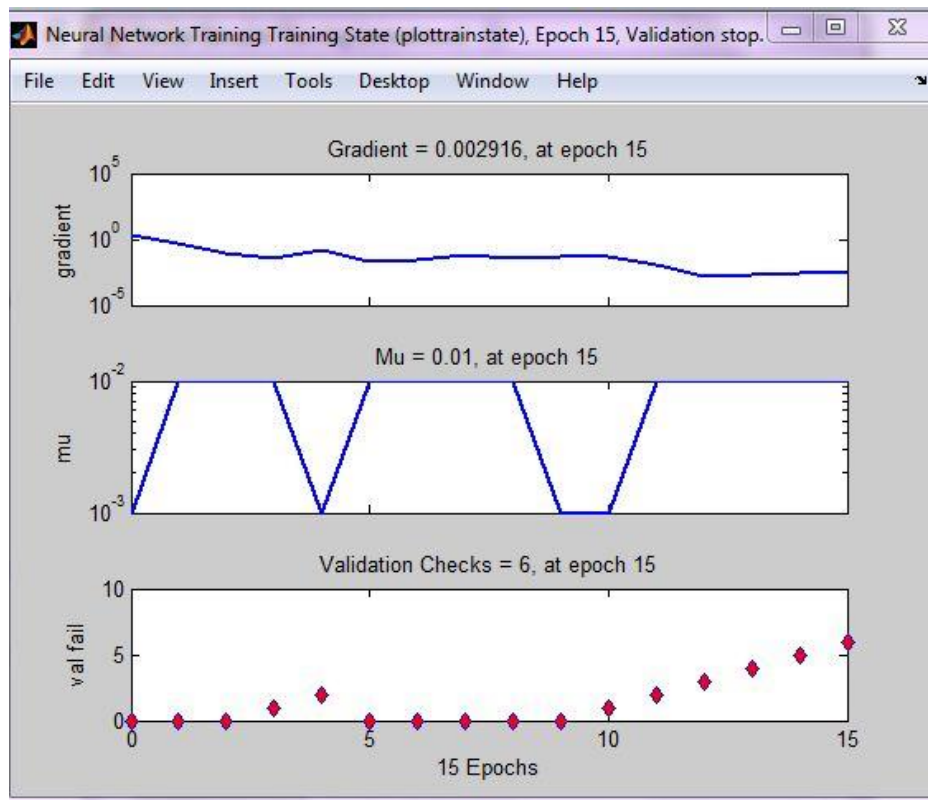


Gambar 11. *Plot Performance*

Grafik *plot performance* yang terdapat pada gambar 36, merupakan grafik yang menunjukkan hubungan MSE (*Mean Squarred Error*) dengan *epoch* pada masing – masing proses *train*, *validation*, dan *test*. Pada gambar 36, *best performance* pada tahap *validation* ialah sebesar 0,24735 yang terjadi saat *epoch* ke 9. Hal tersebut ditandai dengan posisi garis *Validation* yang berwarna hijau

yang berhimpitan dengan garis *Best* dan *Goal* yang berbentuk garis putus – putus pada posisi *epoch* diangka 9.

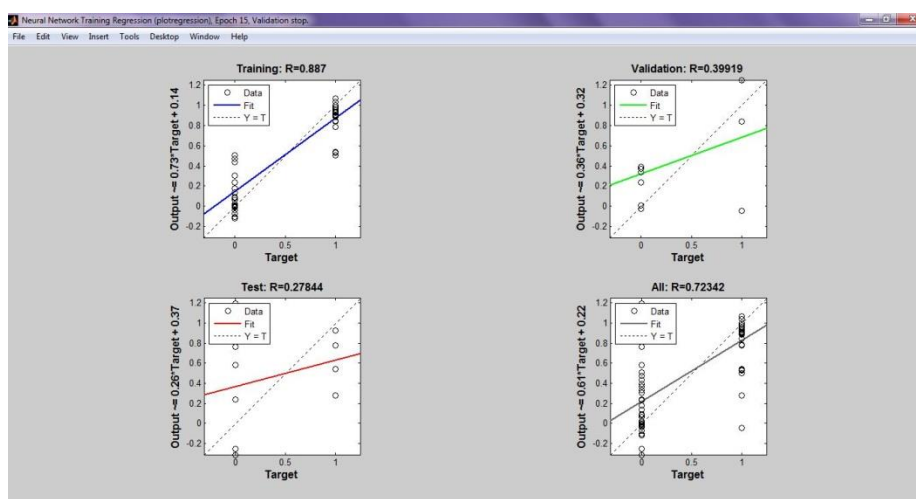
2) Plot Training State



Gambar 12. Plot Training State

Grafik *plot training state* yang terdapat pada gambar 37, merupakan grafik yang menunjukkan hubungan *epoch* dengan *gradient*, *Mu*, dan *Val fail*. Pada gambar 37, ditampilkan bahwa grafik *gradient* menyatakan kemiringan dari epoch 1 sampai 15 memiliki nilai sebesar 0,002916, nilai *Mu* memiliki default sebesar 0,001. Pada epoch 15 nilai *Mu* ialah sebesar 0,01. Serta *Val fail* yang menandai jumlah citra yang gagal tervalidasi pada saat epoch 15.

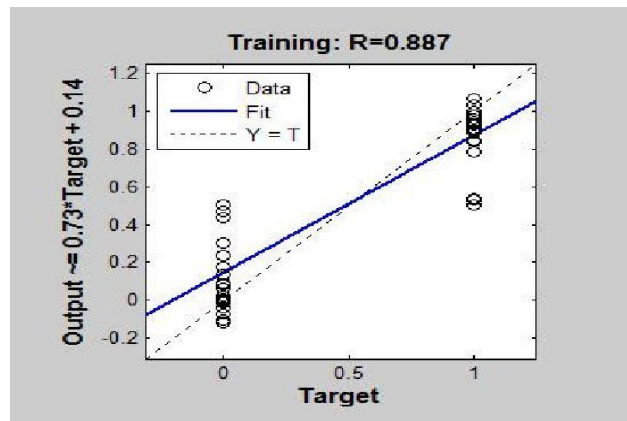
1) Plot Regression



Gambar 13. Grafik *Plot Regression*

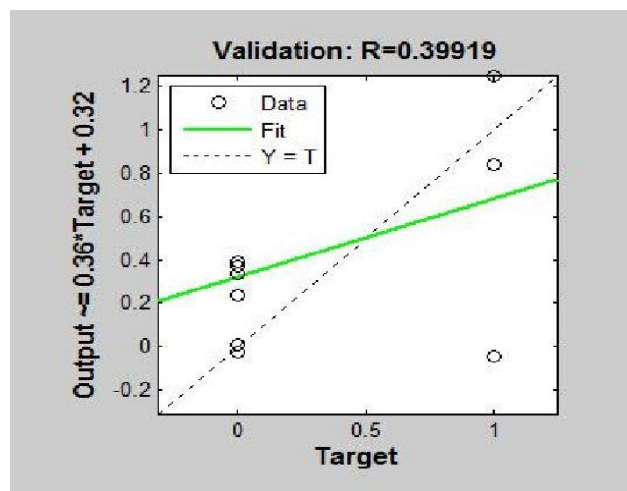
Grafik *plot regression* pada gambar 38 merupakan grafik yang menunjukkan besar nilai regresi pada saat proses *training*, *validation*, *test*, dan keseluruhan. Regresi merupakan nilai yang menunjukkan apakah *output* yang dihasilkan menyimpang dari target atau tidak. Nilai regresi yang mendekati 1 memiliki arti bahwa *output* yang dihasilkan tidak terlalu menyimpang dari target dan sebaliknya.

Pada gambar 39 ditunjukkan gambar grafik regresi dari proses training dengan simbol warna biru. Pada gambar terlihat bahwa nilai regresi pada proses *training* ialah sebesar 0,887 yang berarti *output* tidak terlalu menyimpang dengan target.



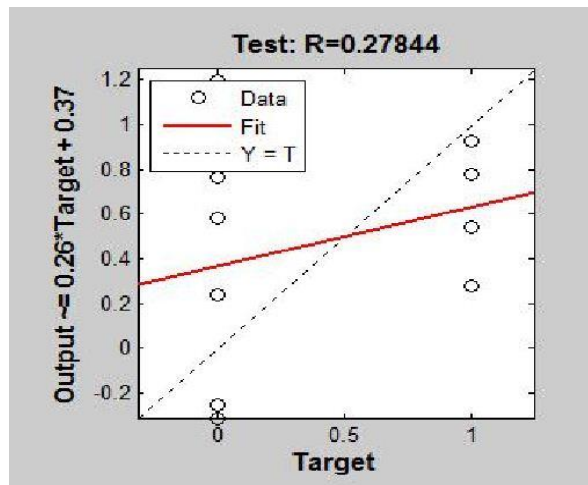
Gambar 14. Grafik regresi saat training

Pada gambar 40 ditunjukkan gambar grafik regresi pada proses *validation* dengan simbol garis hijau. Pada gambar terlihat bahwa nilai regresi pada proses *validation* ialah sebesar 0,39919.



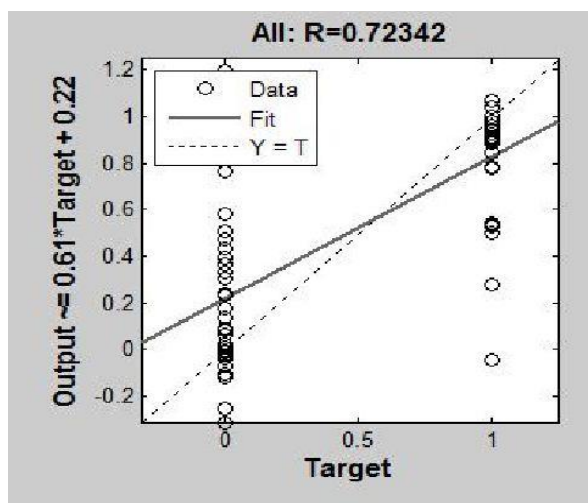
Gambar 15. Grafik regresi validation

Pada gambar 41 ditunjukkan gambar grafik regresi pada proses *test* dengan simbol garis merah. Pada gambar terlihat bahwa nilai regresi pada proses test ialah sebesar 0,27844.



Gambar 16. Grafik regresi *test*

Pada gambar 42 ditunjukkan gambar grafik regresi pada keseluruhan proses dengan simbol garis hitam. Pada gambar terlihat bahwa nilai regresi pada keseluruhan proses ialah sebesar 0,72342.



Gambar 17. Grafik regresi keseluruhan proses

b. Pengujian Jaringan

Pengujian jaringan dilakukan dengan cara memanggil hasil pelatihan dalam bentuk net pada program GUI. Proses pengujian dilakukan dengan cara mencocokkan kesesuaian hasil deteksi dari 40 citra uji dengan hasil medis

responden yang sesungguhnya melalui program GUI. Citra uji yang dicek juga harus berupa citra baru (bukan termasuk citra latih). Setelah dilakukan pengecekan, maka akan ditentukan hasil akurasi dengan rumus sebagai berikut :

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

Besar akurasi yang dihasilkan merupakan tingkat keberhasilan sistem untuk mendeteksi gangguan miopia atau tidak miopia.

Berdasarkan hasil pada tabel 19, dihasilkan akurasi keberhasilan sebesar 70%, dimana masih terdapat 30% kesalahan atau error. Kesalahan ini bisa disebabkan karena berbagai hal seperti data citra yg dilatihkan hanya berjumlah 68 citra, jumlah data tersebut dapat ditingkatkan lagi agar proses pengenalan pola oleh jaringan semakin baik. Berdasarkan hasil pengujian pada tabel 14, citra yang dihasilkan juga mengalami *blurring*. Hal tersebut terjadi karena resolusi *webcam* yang digunakan rendah, *webcam* yang digunakan juga belum memiliki fitur *auto focus*. Akurasi pengujian juga bergantung pada ciri yang diekstraksi karena ciri tersebut akan menjadi pembeda dari masing – masing kategori miopia maupun tidak miopia.

Secara keseluruhan, akurasi pengujian dapat ditingkatkan dengan cara sebagai berikut : (1) meningkatkan jumlah data latih agar proses pengenalan pola oleh jaringan semakin baik; (2) meningkatkan resolusi webcam agar citra yang dihasilkan tidak *blurring*; (3) memperbanyak ciri yang diekstraksi agar pembeda dari masing – masing kategori miopia dan tidak miopia semakin jelas

5. Unjuk Kerja Keseluruhan

Setelah melakukan pengujian terhadap tampilan GUI sistem dari tahap akuisisi data sampai identifikasi hasil seperti pada gambar 34 dan 35 serta melihat tabel hasil unjuk kerja seperti pada tabel 19, diketahui sistem telah mampu melakukan tahap akuisisi data hingga identifikasi hasil. Sistem juga telah mampu mendeteksi miopia (dengan batas terendah miopia sebesar 0,25D) dan tidak miopia dengan akurasi sebesar sebesar 70%.