

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi yang memanfaatkan teknik *Convolutional Neural Network* (CNN) untuk menentukan usia individu secara akurat berdasarkan citra.

4.1.1 Analisis Data

Data yang diperlukan untuk penelitian ini terdiri dari citra wajah yang selanjutnya dilakukan pengolahan citra. *Algoritma Convolutional Neural Network* (CNN) digunakan untuk memilih gambar pelatihan dan pengujian pada tahap pemrosesan data, dengan memanfaatkan bahasa pemrograman *Python*. Berikut beberapa tahapan yang akan dilaksanakan:

1. Menginput dataset citra wajah pelatihan dari rentang umur 1 sampai 90 tahun
2. Pre-processing citra
3. Pemodelan jaringan *Convolutional Neural Network* (CNN)
4. Melakukan training dan testing data menggunakan algoritma CNN

4.1.2 Preprocessing Citra

Sebelum dimanfaatkan sebagai masukan pelatihan, citra tersebut mengalami preprocessing untuk memudahkan algoritma *Convolutional Neural Network* (CNN) dalam melakukan pelatihan dan mengidentifikasi fitur-fitur citra masukan. Sebelum algoritma CNN dapat memproses gambar, dilakukan dua putaran *pre-processing* citra.

1. *Resizing*

Menyesuaikan ukuran piksel seluruh dataset

2. *Grayscaled* Citra

Mengganti citra RGB menjadi citra grayscale

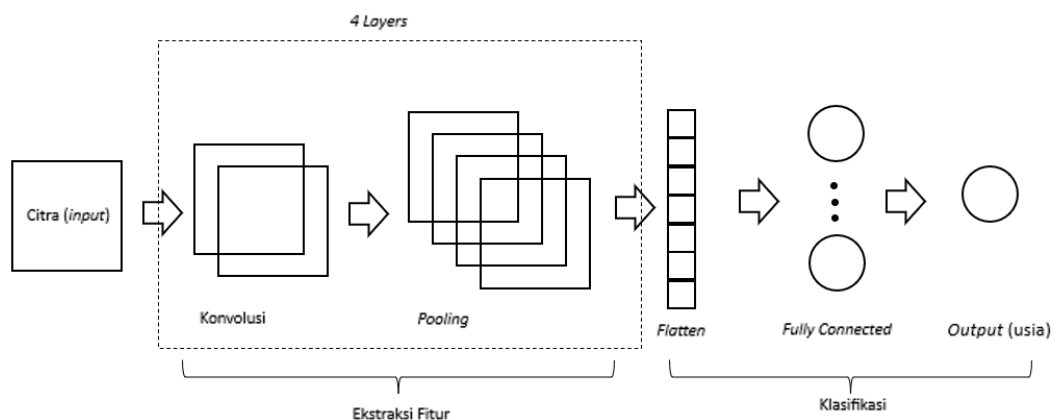
3. Augmentasi Citra

Memanipulasi citra tanpa kehilangan inti atau esensi dari data. Augmentasi yang dilakukan adalah *rescale*, *rotate*, *zoom* dan *flip*.

Seluruh proses *pre-processing* data melibatkan memasukkan semua file ke *TensorFlow* menggunakan *Python*, sehingga menghasilkan perolehan gambar yang disiapkan untuk komputasi menggunakan CNN.

4.1.3 Pemodelan Jaringan

Setelah tahap *preprocessing* citra selesai, data yang diperoleh akan dilakukan *training* dengan menggunakan metode CNN. Arsitektur model jaringan algoritma CNN berdampak signifikan terhadap keakuratan hasil model.



Gambar 4. 1 Pemodelan Jaringan CNN

Citra yang disediakan menggambarkan pemodelan jaringan selama proses pelatihan, sehingga menghasilkan model yang lugas namun sangat tepat. Citra yang digunakan pada penelitian ini memiliki dimensi 48x48 piksel. Konsep pemodelan jaringan CNN dapat dijelaskan sebagai berikut:

1. Tahap Konvolusi 1

Menerapkan kernel 3x3 dengan 48 filter. Tahap ini melibatkan penggabungan dua matriks berbeda untuk menghasilkan nilai matriks baru. Setelah tahap konvolusi selesai, fungsi aktivasi tambahan yang disebut *Rectified Linear Unit (RELU)* akan digabungkan. Tujuan dari fungsi RELU ini adalah untuk mengubah nilai negatif menjadi nol, sehingga menghilangkan nilai negatif pada matriks hasil konvolusi. Tahap konvolusi menggunakan nilai *padding 0*, yang memastikan bahwa ukuran konvolusi yang dihasilkan tetap tidak berubah pada 48x48.

2. Tahap *Pooling* 1

Operasi *pooling* digunakan pada tahap ini untuk memperkecil ukuran matriks. *Pooling layer* pengumpulan terdiri dari filter yang secara siklis akan mengubah ukuran nilai matriksnya pada wilayah *feature maps*. Langkah *pooling* akan menghasilkan nilai matriks baru sebagai hasil pemanfaatan aktivasi *maxpooling* pada penelitian ini. *Maxpooling* beroperasi dengan memilih nilai tertinggi dalam perpindahan karnalnya.

3. Tahap Konvolusi 2

Tahap ini dibangun berdasarkan hasil tahap *pooling* pertama dengan memproses matriks citra 24x24 menggunakan 48 filter, masing-masing dengan ukuran kernel 3x3. Saat ini, fungsi aktivasi RELU juga digunakan.

4. Tahap *Pooling* 2,

Tahapan proses yang terlibat dalam pengumpulan 2 identik dengan prosedur *pooling* yang pertama. Perbedaannya terletak pada besarnya hasil matriks. Operasi *pooling* kedua menghasilkan *output* dengan dimensi 24x24. Langkah konvolusi dan pengumpulan diulang empat kali, menghasilkan total ukuran *output* pada *pooling* terakhir 3x3.

5. *Flatening*

Tujuan dari tahap *flatten* adalah untuk mengubah *output* dari *pooling layer* menjadi vektor. Selanjutnya tahap selanjutnya *fully connected* akan melakukan proses pendeteksian citra dengan vektor *flatten* sebagai nilai input.

6. *Fully Connected*

Temuan yang *flatten* akan diproses menggunakan jaringan saraf *backpropagation* untuk membuat prediksi.

Tabel di bawah menampilkan pemodelan convolutional neural network yang digunakan.

Tabel 4. 1 Model *Convolutional Neural Network*

No.	Nama	Shape	Parameter
1	<i>Input</i>	48x48x3	0
2	<i>Conv_1</i>	48x48x32	896
3	<i>Maxpool_1</i>	24x24x32	0
4	<i>Conv_2</i>	24x24x64	18496
5	<i>Maxpool_2</i>	12x12x64	0
6	<i>Conv_3</i>	12x12x128	73856
8	<i>Maxpool_3</i>	6x6x128	0
9	<i>Conv_4</i>	6x6x256	295168
10	<i>Maxpool_4</i>	3x3x256	0
11	<i>Dense</i>	64	147520
13	<i>Output</i>	1	65
Total			536001

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

4.1.4 Proses Konvolusi

Untuk memahami proses konvolusi, peneliti akan mengekstraksi sampel dari matriks tertentu di dalam *input image*. Ukuran *input image* yang digunakan adalah 48x48 piksel. Oleh karena itu, peneliti secara selektif mengekstraksi sebagian matriks dari gambar masukan untuk dijadikan sampel yang mewakili prosedur konvolusi. Dalam contoh ilustrasi ini, jumlah piksel yang dihitung untuk citra wajah adalah 5×5 piksel. Berikut adalah contoh citra skala *grayscale* dengan nilai piksel yang ditentukan untuk gambar 4.2.

3	1	5	5	4
1	3	3	4	3
5	3	6	3	2
2	3	4	4	3
3	4	5	3	2

 $*$

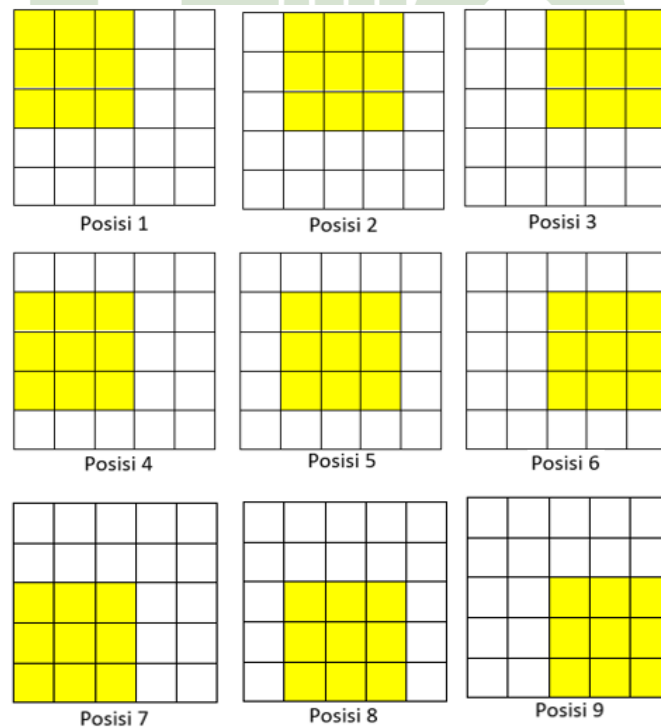
-1	-1	1
-1	1	-1
1	-1	1

 $=$

8	-5	-3
-6	1	-6
-1	-7	-6

Gambar 4. 2 Proses Konvolusi

Citra di atas menunjukkan sebuah karnel dengan *stride* 1 dan dimensi 3x3. *Stride* adalah parameter yang mengontrol jumlah perpindahan filter. Jika nilai *stride* disetel ke 1, maka operasi konvolusi akan dilakukan. Filter akan menerjemahkan 1 piksel secara horizontal dan kemudian vertikal. *Stride* yang digunakan pada gambar yang digambarkan adalah 1. *Stride* pendek menghasilkan informasi masukan yang lebih rumit, meskipun memerlukan peningkatan kebutuhan komputasi dibandingkan dengan *stride* yang lebih besar.



Gambar 4. 3 Pemetaan Proses Konvolusi

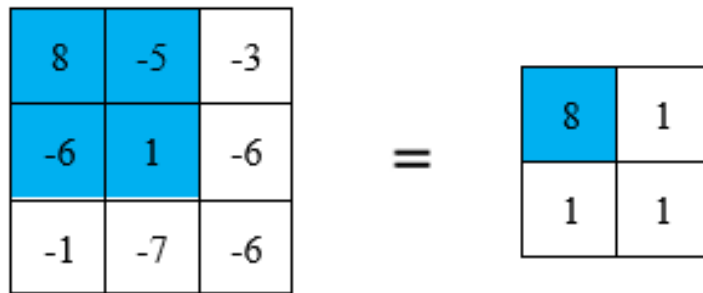
Citra 4.3 mengilustrasikan pemetaan komputasi *dot product* yang terlibat dalam proses konvolusi dengan ukuran kernel 3x3. Produk titik dihitung sebagai berikut:

$$\begin{aligned}
 \text{Posisi 1} &= (3*(-1)) + (1*(-1)) + (5*1) + (1*(-1)) + (3*1) + (3*(-1)) + \\
 &\quad (5*1) + (3*(-1)) + (6*1) = \mathbf{8} \\
 \text{Posisi 2} &= (1*(-1)) + (3*(-1)) + (3*1) + (5*(-1)) + (3*1) + (6*(-1)) + \\
 &\quad (5*1) + (4*(-1)) + (3*1) = \mathbf{-5} \\
 \text{Posisi 3} &= (5*(-1)) + (3*(-1)) + (6*1) + (5*(-1)) + (4*1) + (3*(-1)) + \\
 &\quad (4*1) + (3*(-1)) + (2*1) = \mathbf{-3} \\
 \text{Posisi 4} &= (1*(-1)) + (5*(-1)) + (2*1) + (3*(-1)) + (3*1) + (3*(-1)) + \\
 &\quad (3*1) + (6*(-1)) + (4*1) = \mathbf{-6} \\
 \text{Posisi 5} &= (3*(-1)) + (3*(-1)) + (3*1) + (3*(-1)) + (6*1) + (4*(-1)) + \\
 &\quad (4*1) + (3*(-1)) + (4*1) = \mathbf{1} \\
 \text{Posisi 6} &= (3*(-1)) + (6*(-1)) + (4*1) + (4*(-1)) + (3*1) + (4*(-1)) + \\
 &\quad (3*1) + (2*(-1)) + (3*1) = \mathbf{-6} \\
 \text{Posisi 7} &= (5*(-1)) + (2*(-1)) + (3*1) + (3*(-1)) + (3*1) + (4*(-1)) + \\
 &\quad (6*1) + (4*(-1)) + (5*1) = \mathbf{-1} \\
 \text{Posisi 8} &= (3*(-1)) + (3*(-1)) + (4*1) + (6*(-1)) + (4*1) + (5*(-1)) + \\
 &\quad (3*1) + (4*(-1)) + (3*1) = \mathbf{-7} \\
 \text{Posisi 9} &= (6*(-1)) + (4*(-1)) + (5*1) + (3*(-1)) + (4*1) + (3*(-1)) + \\
 &\quad (2*1) + (3*(-1)) + (2*1) = \mathbf{-6}
 \end{aligned}$$

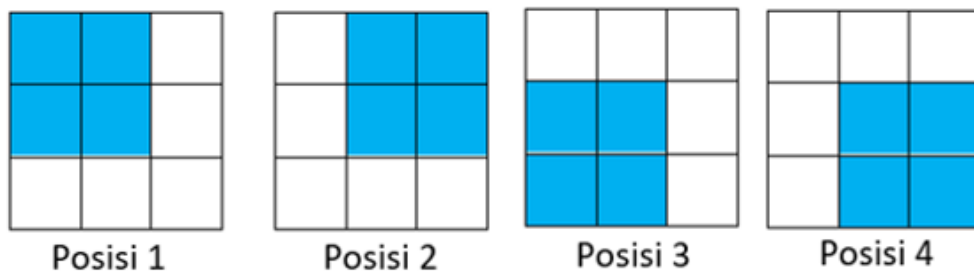
Setelah proses konvolusi selesai, langkah selanjutnya adalah proses *pooling layer*.

4.1.5 Proses Pooling

Pooling adalah teknik yang mengurangi jumlah langkah dalam matriks nilai dengan menggabungkan proses. Pendekatan *maxpooling* biasanya digunakan dalam operasi pengumpulan ini. Strategi ini umumnya digunakan dalam penelitian yang berkaitan dengan pembelajaran mendalam. Representasi visual dari proses *pooling* digambarkan di bawah ini:

Gambar 4. 4 Proses *Pooling*

Ukuran *pooling* yang digunakan seperti terlihat pada citra di atas adalah 2x2 dengan *stride* 1. Pergeseran karnel pada matriks input adalah satu. Selama proses *poolin*, *maxpooling* digunakan untuk mengidentifikasi dan mengekstrak nilai tertinggi. Nilai ini kemudian digeser berdasarkan *stride* dan ukuran yang ditentukan. Output yang dihasilkan diperoleh dengan memilih nilai maksimum dari matriks *feature map* yang dihasilkan selama proses konvolusi, seperti terlihat pada gambar di atas. Hasil dari operasi *maxpooling* adalah matriks 2x2.

Gambar 4. 5 Pemetaan Proses *Pooling*

Citra yang disediakan mengilustrasikan perhitungan *dot product* dalam operasi *maxpooling* menggunakan matriks berdimensi 2x2. Perhitungan *dot product* melibatkan penentuan nilai maksimum dalam setiap *range* lokasi.

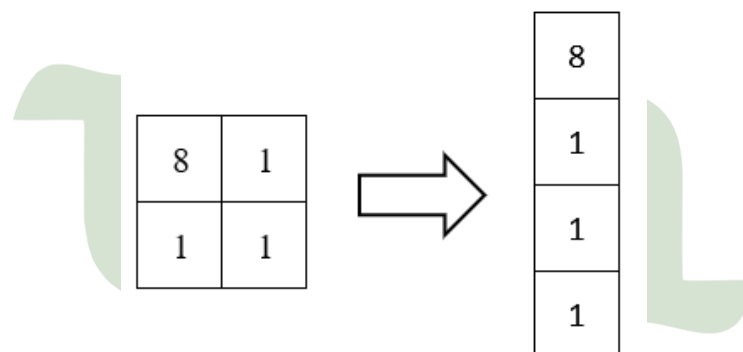
- Posisi 1 = 8, -5, -6, 1. Nilai tertinggi adalah **8**
 Posisi 2 = -5, -3, 1, -6. Nilai tertinggi adalah **1**
 Posisi 3 = -6, 1, -1, -7. Nilai tertinggi adalah **1**
 Posisi 4 = 1, -6, -7, -6. Nilai tertinggi adalah **1**

Prosedur *pooling* melibatkan perolehan hasil yang tetap dalam format *multidimensional array*. Karena hasilnya masih dalam format *array*, maka fungsi

flatten harus disertakan. Fungsi *flatten* mengubah hasil dari format *array* menjadi format vektor, yang kemudian dapat digunakan sebagai *input* untuk *fully connected layer*.

Hanya satu iterasi teknik konvolusi dan *pooling* yang dilakukan dalam sampel ini. Setelah tahap *pooling* selesai, tahap selanjutnya adalah *flatening*. *Flatening* mengacu pada proses mengubah matriks pada *pooling layer* menjadi satu kolom, yang direpresentasikan sebagai vektor. Selanjutnya, vektor ini akan dimasukkan ke dalam *input layer* di *Convolutional Neural Network* (CNN).

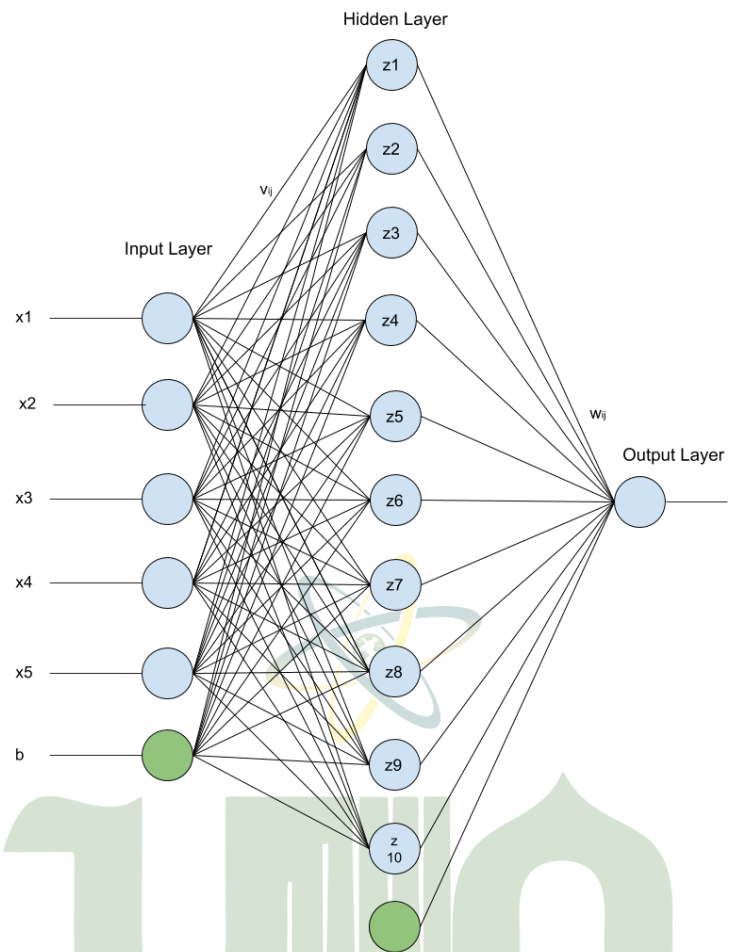
Setiap *layer pooling* akan diubah menjadi vektor. Jika suatu *pooling layer* berdimensi 3x3, maka hasil *flatening* adalah vektor yang terdiri dari 9 elemen. Operasi *pooling* dalam contoh ini menghasilkan matriks 2x2 sebagai hasilnya. Jika matriks-matriks tersebut di *flatening* maka vektor yang dihasilkan akan mempunyai 4 baris.



Gambar 4. 6 Proses *Flatening*

4.1.6 Proses *Fully Connected*

Lapisan *pooling* atau konvolusional menghasilkan hasil *maxpooling*, yang kemudian di *flatening*. Hasil yang *flatening* kemudian dimasukkan ke dalam *full connected layer* untuk pelatihan menggunakan metode backpropagation pada jaringan saraf tiruan. Program peneliti menghasilkan vektor pipih dengan panjang 9 garis setelah penggabungan terakhir. Panjang lebih padat yang digunakan adalah 64. Arsitektur jaringan saraf propagasi mundur yang digunakan dapat diamati pada gambar yang disediakan.



Gambar 4. 7 Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Keterangan :

x = *input* yang berupa vektor dari ekstraksi citra yang berjumlah 9

b = bias

y = *dense* yang berjumlah 64

output = nilai citra

Node output akan menghasilkan nilai citra sebagai keluaran, dan nilai ini akan disimpan untuk diakses selama pengujian, mencakup semua citra dari fase pelatihan. Selama tahap pengujian, nilai output keluaran akan dibandingkan dengan nilai gambar keseluruhan yang diperoleh selama tahap pelatihan untuk mengidentifikasi nilai yang paling mirip dan menetapkannya ke dalam kategori.

Adapun langkah pelatihan jaringan syaraf tiruan *backpropagation* secara

manual adalah sebagai berikut :

Langkah 1 : Inisialisasi bobot bias dan parameter lainnya.

Sebelum melakukan pelatihan, perlu dilakukan inisialisasi bobot dan nilai dari unit tersembunyi dan unit keluaran. Bobot diinisialisasi melalui penetapan nilai acak. Misal didapat bobot seperti tabel 4.2 (bobot dari layer masukan ke *layer* tersembunyi = v_{ij}), tabel 4.3 (bobot dari layer tersembunyi ke *layer* keluaran = w_{jk}) dengan nilai *learning rate* 0.8

Tabel 4. 2 Inisialisasi bobot dan bias menuju *layer* tersembunyi

	x_1	x_2	x_3	x_4	x_5	bias
z_1	-0.0085	-0.0080	-0.0083	-0.0084	-0.0083	-0.0214
z_2	-0.0109	-0.0103	-0.0107	-0.0108	-0.0107	-0.0276
z_3	0.0100	0.0095	0.0098	0.0099	0.0098	0.0254
z_4	-0.0111	-0.0105	-0.0109	-0.0110	-0.0109	-0.0281
z_5	-0.0036	-0.0034	-0.0035	-0.0035	-0.0035	-0.0090
z_6	0.0108	0.0102	0.0106	0.0107	0.0106	0.0274
z_7	0.0060	0.0056	0.0058	0.0059	0.0058	0.0151
z_8	-0.0013	-0.0012	-0.0012	-0.0013	-0.0012	-0.0032
z_9	-0.0123	-0.0116	-0.0121	-0.0122	-0.0121	-0.0311
z_{10}	-0.0125	-0.0118	-0.0123	-0.0124	-0.0123	-0.0316

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN
Tabel 4. 3 Inisialisasi bobot dan bias menuju *layer* keluaran

	y
z_1	1.5623
z_2	2.0221
z_3	-1.9105
z_4	2.0604
z_5	0.6414
z_6	-2.0591
z_7	-1.1454
z_8	0.2098
z_9	2.2833
z_{10}	2.3205

<i>bias</i>	-3.1809
-------------	---------

Langkah 2 :

Lakukan langkah 3-9 apabila kondisi yang diinginkan belum terpenuhi (mse = 0.0001 dan epoch = 1000)

Langkah 3 :

Untuk setiap pasangan data latih yang digunakan, maka lakukan langkah 4 -8:

Langkah 4 (Fase *Feedforwad*):

Hitung keluaran unit tersembunyi (z_j)

$$z_{net_j} = b_{ij} + \sum x_i v_{ij}$$

$$\begin{aligned} z_{net_1} &= b_{01} + x_1 v_{11} + x_2 v_{21} + x_3 v_{31} + x_4 v_{41} + x_5 v_{51} \\ &= -0.0214 + (0.3962 * -0.0085) + (0.3731 * -0.0080) \\ &\quad + (0.3885 * -0.0083) + (0.3923 * -0.0084) \\ &\quad + (0.3885 * -0.0083) = -0.0375 \end{aligned}$$

$$\begin{aligned} z_{net_2} &= b_{02} + x_1 v_{12} + x_2 v_{22} + x_3 v_{32} + x_4 v_{42} + x_5 v_{52} \\ &= -0.0276 + (0.3962 * -0.0109) + (0.3731 * -0.0103) \\ &\quad + (0.3885 * -0.0107) + (0.3923 * -0.0108) \\ &\quad + (0.3885 * -0.0107) = -0.0483 \end{aligned}$$

$$\begin{aligned} z_{net_3} &= b_{03} + x_1 v_{13} + x_2 v_{23} + x_3 v_{33} + x_4 v_{43} + x_5 v_{53} \\ &= 0.0254 + (0.3962 * 0.0100) + (0.3731 * 0.0095) \\ &\quad + (0.3885 * 0.0098) + (0.3923 * 0.0099) \\ &\quad + (0.3885 * 0.0098) = 0.0444 \end{aligned}$$

$$\begin{aligned}
 z_{net4} &= b_{04} + x_1v_{14} + x_2v_{24} + x_3v_{34} + x_4v_{44} + x_5v_{54} \\
 &= -0.0281 + (0.3962 * -0.0111) + (0.3731 * -0.0105) \\
 &\quad + (0.3885 * -0.0109) + (0.3923 * -0.0110) \\
 &\quad + (0.3885 * -0.0109) = -0.0492
 \end{aligned}$$

$$\begin{aligned}
 z_{net5} &= b_{05} + x_1v_{15} + x_2v_{25} + x_3v_{35} + x_4v_{45} + x_5v_{55} \\
 &= -0.0090 + (0.3962 * -0.0036) + (0.3731 * -0.0034) \\
 &\quad + (0.3885 * -0.0035) + (0.3923 * -0.0035) \\
 &\quad + (0.3885 * -0.0035) = -0.0158
 \end{aligned}$$

$$\begin{aligned}
 z_{net6} &= b_{06} + x_1v_{16} + x_2v_{26} + x_3v_{36} + x_4v_{46} + x_5v_{56} \\
 &= 0.0274 + (0.3962 * 0.0108) + (0.3731 * 0.0102) \\
 &\quad + (0.3885 * 0.0106) + (0.3923 * 0.0107) \\
 &\quad + (0.3885 * 0.0106) = 0.0479
 \end{aligned}$$

$$\begin{aligned}
 z_{net7} &= b_{07} + x_1v_{17} + x_2v_{27} + x_3v_{37} + x_4v_{47} + x_5v_{57} \\
 &= 0.0151 + (0.3962 * 0.0060) + (0.3731 * 0.0056) \\
 &\quad + (0.3885 * 0.0058) + (0.3923 * 0.0059) \\
 &\quad + (0.3885 * 0.0058) = 0.0264
 \end{aligned}$$

$$\begin{aligned}
 z_{net8} &= b_{08} + x_1v_{18} + x_2v_{28} + x_3v_{38} + x_4v_{48} + x_5v_{58} \\
 &= -0.0032 + (0.3962 * -0.0013) + (0.3731 * -0.0012) \\
 &\quad + (0.3885 * -0.0012) + (0.3923 * -0.0013) \\
 &\quad + (0.3885 * -0.0012) = -0.0056
 \end{aligned}$$

$$\begin{aligned}
 z_{net9} &= b_{09} + x_1v_{19} + x_2v_{29} + x_3v_{39} + x_4v_{49} + x_5v_{59} \\
 &= -0.0311 + (0.3962 * -0.0123) + (0.3731 * -0.0116)
 \end{aligned}$$

$$+ (0.3885 * -0.0121) + (0.3923 * -0.0122)$$

$$+ (0.3885 * -0.0121) = -0.0545$$

$$\begin{aligned} z_{net10} &= b_{010} + x_1v_{110} + x_2v_{210} + x_3v_{310} + x_4v_{410} + x_5v_{510} \\ &= -0.0316 + (0.3962 * -0.0125) + (0.3731 * -0.0118) \\ &\quad + (0.3885 * -0.0123) + (0.3923 * -0.0124) \\ &\quad + (0.3885 * -0.0123) = -0.0554 \end{aligned}$$

Menghitung aktivasi dengan fungsi sigmoid biner:

$$z_j = \frac{1}{1 + \exp^{-y_{net_k}}}$$

$$z_1 = \frac{1}{1 + \exp^{-(-0.0375)}} = 0.4906$$

$$z_2 = \frac{1}{1 + \exp^{-(-0.0483)}} = 0.4879$$

$$z_3 = \frac{1}{1 + \exp^{-(0.0444)}} = 0.5111$$

$$z_4 = \frac{1}{1 + \exp^{-(-0.0492)}} = 0.4877$$

$$z_5 = \frac{1}{1 + \exp^{-(-0.0158)}} = 0.4961$$

$$z_6 = \frac{1}{1 + \exp^{-(0.0479)}} = 0.5120$$

$$z_7 = \frac{1}{1 + \exp^{-(0.0264)}} = 0.5066$$

$$z_8 = \frac{1}{1 + \exp^{-(-0.0056)}} = 0.4986$$

$$z_9 = \frac{1}{1 + \exp^{-(-0.0545)}} = 0.4864$$

$$z_{10} = \frac{1}{1 + \exp^{-(-0.0554)}} = 0.4862$$

Langkah 5 :

Menghitung hasil pada unit output y_k

$$y_{net_k} = w_{k0} + \sum_j^p z_j w_{kj}$$

$$y_{net_1} = w_{k0} + z_1 w_{11} + z_2 w_{12} + z_3 w_{13} + z_4 w_{14} + z_5 w_{15} + z_6 w_{16}$$

$$+ z_7 w_{17} + z_8 w_{18} + z_9 w_{19} + z_{10} w_{110}$$

$$\begin{aligned} y_{net_1} &= -3.1809 + (0.4906 * 1.5623) + (0.4879 * 2.0221) \\ &\quad + (0.5111 * -1.9105) + (0.4877 * 2.0604) \\ &\quad + (0.4961 * 0.6414) + (0.5120 * -2.0591) \\ &\quad + (0.5066 * -1.1454) + (0.4986 * 0.2098) \\ &\quad + (0.4864 * 2.2833) + (0.4862 * 2.3205) = \\ &\quad -0.3723 \end{aligned}$$

Menghitung aktivasi dengan fungsi sigmoid biner:

$$Y_k = \frac{1}{1 + \exp^{-Y_{net_k}}}$$

$$Y_1 = \frac{1}{1 + \exp^{-(-0.3723)}} = 0.4080$$

Langkah 6 (Fase Backpropagation) :

Menghitung nilai faktor δ pada unit output berdasarkan nilai kesalahan di setiap unit keluaran y_k

$$\delta_k = (t_k - y_k)(1 - y_k)$$

$$\delta_k = (0.3654 - 0.4080) * 0.4080 * (1 - 0.4080)$$

$$\delta_k = -0.0103$$

Langkah 7 :

Menghitung faktor δ pada unit tersembunyi berdasarkan kesalahan pada unit tersembunyi z_j .

$$\delta_{net_j} = \sum_k^m \delta_k W_{kj}$$

$$\delta_{net_1} = \delta_k W_{11} = -0.0103 * 1.5623 = -0.0161$$

$$\delta_{net_2} = \delta_k W_{12} = -0.0103 * 2.0221 = -0.0208$$

$$\delta_{net_3} = \delta_k W_{13} = -0.0103 * -1.9105 = 0.0197$$

$$\delta_{net_4} = \delta_k W_{14} = -0.0103 * 2.0604 = -0.0212$$

$$\delta_{net_5} = \delta_k W_{15} = -0.0103 * 0.6414 = -0.0066$$

$$\delta_{net_6} = \delta_k W_{16} = -0.0103 * -2.0591 = 0.0212$$

$$\delta_{net_7} = \delta_k W_{17} = -0.0103 * -1.1454 = 0.0118$$

$$\delta_{net_8} = \delta_k W_{18} = -0.0103 * 0.2098 = -0.0022$$

$$\delta_{net_9} = \delta_k W_{19} = -0.0103 * 2.2833 = -0.0235$$

$$\delta_{net_{10}} = \delta_k W_{110} = -0.0103 * 2.3205 = -0.0239$$

Faktor δ untuk unit tersembunyi :

$$\delta_j = \delta_{net_j} z_j (1 - z_j)$$

$$\delta_{1j} = \delta_{net_1} z_1 (1 - z_1) = -0.0161 * 0.4906 * (1 - 0.4906) = -0.0040$$

$$\delta_{2j} = \delta_{net_2} z_2 (1 - z_2) = -0.0208 * 0.4879 * (1 - 0.4879) = -0.0052$$

$$\delta_{3j} = \delta_{net_3} z_3 (1 - z_3) = 0.0197 * 0.5111 * (1 - 0.5111) = 0.0049$$

$$\delta_{4j} = \delta_{net_4} z_4 (1 - z_4) = -0.0212 * 0.4877 * (1 - 0.4877) = -0.0053$$

$$\delta_{5j} = \delta_{net_5} z_5 (1 - z_5) = -0.0066 * 0.4961 * (1 - 0.4961) = -0.0016$$

$$\delta_{6j} = \delta_{net_6} z_6 (1 - z_6) = 0.0212 * 0.5120 * (1 - 0.5120) = 0.0053$$

$$\delta_{7j} = \delta_{net_7} z_7 (1 - z_7) = 0.0118 * 0.5066 * (1 - 0.5066) = 0.0029$$

$$\delta_{8j} = \delta_{net_8} z_8 (1 - z_8) = -0.0022 * 0.4986 * (1 - 0.4986) = -0.0005$$

$$\delta_{9j} = \delta_{net_9} z_9 (1 - z_9) = -0.0235 * 0.4864 * (1 - 0.4864) = -0.0059$$

$$\delta_{10j} = \delta_{net_{10}} z_{10} (1 - z_{10}) = -0.0239 * 0.4862 * (1 - 0.4862) = -0.0060$$

Langkah 8 :

Untuk menghitung suku perubahan bobot dan bias pada unit tersembunyi, dikarenakan perhitungan masih berada pada epoch 1, maka persamaan dibawah ini

$$\Delta v_{ij} = \alpha \psi_{1ij} = \alpha \delta_j x_i$$

$$\Delta v_{11} = \alpha \delta_1 x_1 = 0.8 * -0.0040 * 0.3962 = -0.0013$$

$$\Delta v_{12} = \alpha \delta_2 x_1 = 0.8 * -0.0052 * 0.3962 = -0.0016$$

$$\Delta v_{13} = \alpha \delta_3 x_1 = 0.8 * 0.0049 * 0.3962 = 0.0016$$

$$\Delta v_{14} = \alpha \delta_4 x_1 = 0.8 * -0.0053 * 0.3962 = -0.0017$$

$$\Delta v_{15} = \alpha \delta_5 x_1 = 0.8 * -0.0016 * 0.3962 = -0.0005$$

$$\Delta v_{16} = \alpha \delta_6 x_1 = 0.8 * 0.0053 * 0.3962 = 0.0017$$

$$\Delta v_{17} = \alpha \delta_7 x_1 = 0.8 * 0.0029 * 0.3962 = 0.0009$$

$$\Delta v_{18} = \alpha \delta_8 x_1 = 0.8 * -0.0005 * 0.3962 = -0.0002$$

$$\Delta v_{19} = \alpha \delta_9 x_1 = 0.8 * -0.0059 * 0.3962 = -0.0019$$

$$\Delta v_{110} = \alpha \delta_{10} x_1 = 0.8 * -0.0060 * 0.3962 = -0.0019$$

$$\Delta v_{21} = \alpha \delta_1 x_2 = 0.8 * -0.0040 * 0.3731 = -0.0012$$

$$\Delta v_{22} = \alpha \delta_2 x_2 = 0.8 * -0.0052 * 0.3731 = -0.0016$$

$$\Delta v_{23} = \alpha \delta_3 x_2 = 0.8 * 0.0049 * 0.3731 = 0.0015$$

$$\Delta v_{24} = \alpha \delta_4 x_2 = 0.8 * -0.0053 * 0.3731 = -0.0016$$

$$\Delta v_{25} = \alpha \delta_5 x_2 = 0.8 * -0.0016 * 0.3731 = -0.0005$$

$$\Delta v_{26} = \alpha \delta_6 x_2 = 0.8 * 0.0053 * 0.3731 = 0.0016$$

$$\Delta v_{27} = \alpha \delta_7 x_2 = 0.8 * 0.0029 * 0.3731 = 0.0009$$

$$\Delta v_{28} = \alpha \delta_8 x_2 = 0.8 * -0.0005 * 0.3731 = -0.0002$$

$$\Delta v_{29} = \alpha \delta_9 x_2 = 0.8 * -0.0059 * 0.3731 = -0.0018$$

$$\Delta v_{210} = \alpha \delta_{10} x_2 = 0.8 * -0.0060 * 0.3731 = -0.0018$$

$$\Delta v_{31} = \alpha \delta_1 x_3 = 0.8 * -0.0040 * 0.3885 = -0.0012$$

$$\Delta v_{32} = \alpha \delta_2 x_3 = 0.8 * -0.0052 * 0.3885 = -0.0016$$

$$\Delta v_{33} = \alpha \delta_3 x_3 = 0.8 * 0.0049 * 0.3885 = 0.0015$$

$$\Delta v_{34} = \alpha \delta_4 x_3 = 0.8 * -0.0053 * 0.3885 = -0.0016$$

$$\Delta v_{35} = \alpha \delta_5 x_3 = 0.8 * -0.0016 * 0.3885 = -0.0005$$

$$\Delta v_{36} = \alpha \delta_6 x_3 = 0.8 * 0.0053 * 0.3885 = 0.0016$$

$$\Delta v_{37} = \alpha \delta_7 x_3 = 0.8 * 0.0029 * 0.3885 = 0.0009$$

$$\Delta v_{38} = \alpha \delta_8 x_3 = 0.8 * -0.0005 * 0.3885 = -0.0002$$

$$\Delta v_{39} = \alpha \delta_9 x_3 = 0.8 * -0.0059 * 0.3885 = -0.0018$$

$$\Delta v_{310} = \alpha \delta_{10} x_3 = 0.8 * -0.0060 * 0.3885 = -0.0019$$

$$\Delta v_{41} = \alpha \delta_1 x_4 = 0.8 * -0.0040 * 0.3923 = -0.0013$$

$$\Delta v_{42} = \alpha \delta_2 x_4 = 0.8 * -0.0052 * 0.3923 = -0.0016$$

$$\Delta v_{43} = \alpha \delta_3 x_4 = 0.8 * 0.0049 * 0.3923 = 0.0015$$

$$\Delta v_{44} = \alpha \delta_4 x_4 = 0.8 * -0.0053 * 0.3923 = -0.0017$$

$$\Delta v_{45} = \alpha \delta_5 x_4 = 0.8 * -0.0016 * 0.3923 = -0.0005$$

$$\Delta v_{46} = \alpha \delta_6 x_4 = 0.8 * 0.0053 * 0.3923 = 0.0017$$

$$\Delta v_{47} = \alpha \delta_7 x_4 = 0.8 * 0.0029 * 0.3923 = 0.0009$$

$$\Delta v_{48} = \alpha \delta_8 x_4 = 0.8 * -0.0005 * 0.3923 = -0.0002$$

$$\Delta v_{49} = \alpha \delta_9 x_4 = 0.8 * -0.0059 * 0.3923 = -0.0018$$

$$\Delta v_{410} = \alpha \delta_{10} x_4 = 0.8 * -0.0060 * 0.3923 = -0.0019$$

$$\Delta v_{51} = \alpha \delta_1 x_5 = 0.8 * -0.0040 * 0.3885 = -0.0012$$

$$\Delta v_{52} = \alpha \delta_2 x_5 = 0.8 * -0.0052 * 0.3885 = -0.0016$$

$$\Delta v_{53} = \alpha \delta_3 x_5 = 0.8 * 0.0049 * 0.3885 = 0.0015$$

$$\Delta v_{54} = \alpha \delta_4 x_5 = 0.8 * -0.0053 * 0.3885 = -0.0016$$

$$\Delta v_{55} = \alpha \delta_5 x_5 = 0.8 * -0.0016 * 0.3885 = -0.0005$$

$$\Delta v_{56} = \alpha \delta_6 x_5 = 0.8 * 0.0053 * 0.3885 = 0.0016$$

$$\Delta v_{57} = \alpha \delta_7 x_5 = 0.8 * 0.0029 * 0.3885 = 0.0009$$

$$\Delta v_{58} = \alpha \delta_8 x_5 = 0.8 * -0.0005 * 0.3885 = -0.0002$$

$$\Delta v_{59} = \alpha \delta_9 x_5 = 0.8 * -0.0059 * 0.3885 = -0.0018$$

$$\Delta v_{510} = \alpha \delta_{10} x_5 = 0.8 * -0.0060 * 0.3885 = -0.0019$$

$$\Delta b_{1j} = \alpha \beta_{1ij} = \alpha \delta_j$$

$$\Delta b_{1_1} = \alpha \delta_1 = 0.8 * -0.0040 = -0.0032$$

$$\Delta b_{1_2} = \alpha \delta_2 = 0.8 * -0.0052 = -0.0042$$

$$\Delta b_{1_3} = \alpha \delta_3 = 0.8 * 0.0049 = 0.0039$$

$$\Delta b_{1_4} = \alpha \delta_4 = 0.8 * -0.0053 = -0.0042$$

$$\Delta b_{1_5} = \alpha \delta_5 = 0.8 * -0.0016 = -0.0013$$

$$\Delta b_{1_6} = \alpha \delta_6 = 0.8 * 0.0053 = 0.0042$$

$$\Delta b_{1_7} = \alpha \delta_7 = 0.8 * 0.0029 = 0.0024$$

$$\Delta b_{1_8} = \alpha \delta_8 = 0.8 * -0.0005 = -0.0004$$

$$\Delta b_{1_9} = \alpha \delta_9 = 0.8 * -0.0059 = -0.0047$$

$$\Delta b_{1_{10}} = \alpha \delta_{10} = 0.8 * -0.0060 = -0.0048$$

Untuk menghitung suku perubahan bobot dan bias pada unit tersembunyi, dikarenakan perhitungan masih berada pada epoch 1, maka persamaan yang digunakan menggunakan persamaan berikut.

$$\Delta w_{jk} = \alpha \psi_{2jk} = \alpha \delta_{kz_j}$$

$$\Delta w_{1k} = \alpha \delta_{kz_1} = 0.8 * -0.0313 * 0.4906 = -0.0040$$

$$\Delta w_{2k} = \alpha \delta_{kz_2} = 0.8 * -0.0313 * 0.4879 = -0.0040$$

$$\Delta w_{3k} = \alpha \delta_{kz_3} = 0.8 * -0.0313 * 0.5111 = -0.0042$$

$$\Delta w_{4k} = \alpha \delta_{kz_4} = 0.8 * -0.0313 * 0.4877 = -0.0040$$

$$\Delta w_{5k} = \alpha \delta_{kz_5} = 0.8 * -0.0313 * 0.4961 = -0.0041$$

$$\Delta w_{6k} = \alpha \delta_{kz_6} = 0.8 * -0.0313 * 0.5120 = -0.0042$$

$$\Delta w_{7k} = \alpha \delta_k z_7 = 0.8 * -0.0313 * 0.5066 = -0.0042$$

$$\Delta w_{8k} = \alpha \delta_k z_8 = 0.8 * -0.0313 * 0.4986 = -0.0041$$

$$\Delta w_{9k} = \alpha \delta_k z_9 = 0.8 * -0.0313 * 0.4864 = -0.0040$$

$$\Delta w_{10k} = \alpha \delta_k z_{10} = 0.8 * -0.0313 * 0.4862 = -0.0040$$

$$\Delta b_{2k} = \alpha \beta z_k = \alpha \delta_k$$

$$\Delta b_{2k} = \alpha \delta_k = 0.8 * -0.0313 = -0.0082$$

Dikarenakan perhitungan masih berada pada epoh 1, maka *perhitungan adaptive learning rate* tidak dapat dilakukan pada epoh 1, perhitungan ini dapat dilakukan diatas 1 epoh

Langkah 9 (Perubahan Bobot) :

Perubahan bobot dan bias menuju unit tersembunyi

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

Tabel 4. 4 Perubahan bobot dan bias pada unit tersembunyi

	x_1	x_2	x_3	x_4	x_5	bias
z_1	-0.0097	-0.0092	-0.0096	-0.0097	-0.0096	-0.0246
z_2	-0.0126	-0.0118	-0.0123	-0.0125	-0.0123	-0.0317
z_3	0.0116	0.0109	0.0114	0.0115	0.0114	0.0293
z_4	-0.0128	-0.0121	-0.0126	-0.0127	-0.0126	-0.0323
z_5	-0.0041	-0.0038	-0.0040	-0.0040	-0.0040	-0.0103
z_6	0.0125	0.0118	0.0123	0.0124	0.0123	0.0316
z_7	0.0069	0.0065	0.0068	0.0068	0.0068	0.0174
z_8	-0.0014	-0.0013	-0.0014	-0.0014	-0.0014	-0.0036
z_9	-0.0142	-0.0134	-0.0139	-0.0140	-0.0139	-0.0358
z_{10}	-0.0144	-0.0136	-0.0141	-0.0143	-0.0141	-0.0364

Perubahan bobot dan bias menuju unit keluaran

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

Tabel 4. 5 Perubahan bobot dan bias pada unit keluaran

	Y
z ₁	1.5582
z ₂	2.0181
z ₃	-1.9147
z ₄	2.0564
z ₅	0.6373
z ₆	-2.0634
z ₇	-1.1496
z ₈	0.2057
z ₉	2.2793
z ₁₀	2.3165
<i>bias</i>	-2.0634

Berikut adalah ilustrasi komputasi yang terlibat dalam proses prediksi menggunakan jaringan saraf tiruan *backpropagation*. Perhitungan ini secara eksklusif akan menggunakan sampel dari arsitektur yang terdiri dari 5 node input, 10 node pada hidden layer dan nilai.

Tabel 4. 6 Data Prediksi

No.	x1	x2	x3	x4	X5
1	2	5	3	4	1

Langkah pertama adalah melakukan normalisasi dengan rumus normalisasi.

$$\text{Normalisasi} = \frac{\text{Data Asli} - \text{Min}}{\text{Maks} - \text{Min}}$$

Tabel 4. 7 Data Prediksi Hasil Normalisasi

No.	x1_norm	x2_norm	x3_norm	x4_norm	x5_norm
1	0.3	0.9	0.5	0.7	0.1

Kemudian inialisasi bobot V_{ij} dan bobot W_{jk} dengan bilangan acak Setelah ditetapkan nilai bobot setiap node, hitung keluaran di unit hidden (Z_j).

$$Z_{netj} = X_0 \sum_{i=1}^n X_i V_{ij}$$

$$\begin{aligned} Z_{net1} &= -0.0246 + (0.3 * -0.0097) + (0.9 * -0.0092) + (0.5 * -0.0096) + (0.7 * -0.0097) \\ &\quad + (0.1 * -0.0096) = -0.0395 \end{aligned}$$

$$\begin{aligned} Z_{net2} &= -0.0317 + (0.3 * -0.0126) + (0.9 * -0.0118) + (0.5 * -0.0123) + (0.7 * -0.0125) \\ &\quad + (0.1 * -0.0123) = -0.0582 \end{aligned}$$

$$\begin{aligned} Z_{net3} &= 0.0293 + (0.3 * 0.0116) + (0.9 * 0.0109) + (0.5 * 0.0114) + (0.7 * 0.0115) + \\ &\quad (0.1 * 0.0114) = 0.0428 \end{aligned}$$

$$\begin{aligned} Z_{net4} &= -0.0323 + (0.3 * -0.0128) + (0.9 * -0.0121) + (0.5 * -0.0126) + (0.7 * -0.0127) \\ &\quad + (0.1 * -0.0126) = -0.0590 \end{aligned}$$

$$\begin{aligned} Z_{net5} &= -0.0103 + (0.3 * -0.0041) + (0.9 * -0.0038) + (0.5 * -0.0040) + (0.7 * -0.0040) \\ &\quad + (0.1 * -0.0040) = -0.0222 \end{aligned}$$

$$\begin{aligned} Z_{net6} &= 0.0316 + (0.3 * 0.0125) + (0.9 * 0.0118) + (0.5 * 0.0123) + (0.7 * 0.0124) + \\ &\quad (0.1 * 0.0123) = 0.0449 \end{aligned}$$

$$\begin{aligned} Z_{net7} &= 0.0174 + (0.3 * 0.0069) + (0.9 * 0.0065) + (0.5 * 0.0068) + (0.7 * 0.0068) + \\ &\quad (0.1 * 0.0068) = 0.0257 \end{aligned}$$

$$\begin{aligned} Z_{net8} &= -0.0036 + (0.3 * -0.0014) + (0.9 * -0.0013) + (0.5 * -0.0014) + (0.7 * -0.0014) \\ &\quad + (0.1 * -0.0014) = -0.0062 \end{aligned}$$

$$\begin{aligned} Z_{net9} &= -0.0358 + (0.3 * -0.0142) + (0.9 * -0.0134) + (0.5 * -0.0139) + (0.7 * -0.0140) \\ &\quad + (0.1 * -0.0139) = -0.0624 \end{aligned}$$

$$\begin{aligned} Z_{net10} &= -0.0364 + (0.3 * -0.0144) + (0.9 * -0.0136) + (0.5 * -0.0141) + (0.7 * -0.0143) \\ &\quad + (0.1 * -0.0141) = -0.0638 \end{aligned}$$

Setelah nilai keluaran di node hidden diperoleh, langkah selanjutnya hitungkeluaran di hidden layer menggunakan fungsi aktivasi

$$Z1 = \frac{1}{1 + \exp(-Z_{netj})}$$

$$Z1 = \frac{1}{1 + \exp^{-(-0.0395)}} = 0.4912$$

$$Z2 = \frac{1}{1 + \exp^{-(-0.0582)}} = 0.4854$$

$$Z3 = \frac{1}{1 + \exp^{-(-0.0428)}} = 0.5106$$

$$Z4 = \frac{1}{1 + \exp^{-(-0.0590)}} = 0.4852$$

$$Z5 = \frac{1}{1 + \exp^{-(-0.0222)}} = 0.4944$$

$$Z6 = \frac{1}{1 + \exp^{-(-0.0449)}} = 0.5112$$

$$Z7 = \frac{1}{1 + \exp^{-(-0.0257)}} = 0.5064$$

$$Z8 = \frac{1}{1 + \exp^{-(-0.0062)}} = 0.4984$$

$$Z9 = \frac{1}{1 + \exp^{-(-0.0624)}} = 0.4844$$

$$Z10 = \frac{1}{1 + \exp^{-(-0.0638)}} = 0.4840$$

Langkah selanjutnya adalah menghitung keluaran di semua jaringan hidden ke output layer (Y).

$$Y_{netk} = W_0 \sum_{i=1}^n Z_i W_{jk}$$

$$\begin{aligned} Y_{net} = & -2.0634 + (1.5582*0.4912) + (2.0181*0.4854) + (-1.9147*0.5106) + \\ & (2.0564*0.4852) + (0.6373*0.4944) + (-2.0634*0.5112) + (-1.1496*0.5064) \\ & + (0.2057*0.4984) + (2.2793*0.4844) + (2.3165*0.4840) = 0.7076 \end{aligned}$$

Setelah *output* diperoleh pada node output, keluaran pada *output layer* dapat dihitung menggunakan fungsi aktivasi. Fungsi aktivasi didefinisikan sebagai

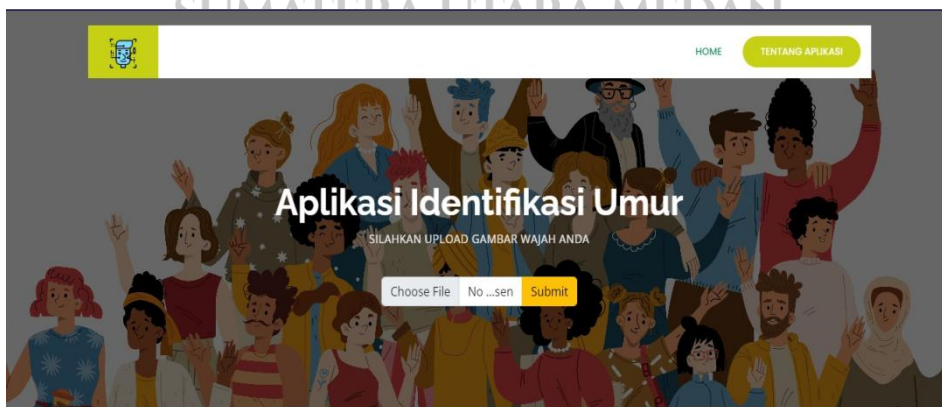
$$Y_k = \frac{1}{1 + \exp(-Y_{net_k})}$$

$$Y_1 = \frac{1}{1 + \exp^{-(0.7076)}} = 0.6698$$

Setelah nilai keluaran yang diaktifkan diterima di node keluaran, langkah terakhir adalah denormalisasi. Dengan melakukan denormalisasi nilai sebesar 0,6698 maka diperoleh nilai aktual yang akan menentukan kategori objek prediksi, apakah tergolong tua atau muda. Nilai tersebut selanjutnya akan dibandingkan dengan nilai yang diperoleh dari foto-foto yang dilatih sebelumnya untuk memastikan rentang usia individu yang tergambar dalam foto-foto yang diolah tersebut.

4.2 Hasil

Proses pelatihan dan pengujian dilakukan melalui desain jaringan saraf propagasi mundur yang telah ditentukan sebelumnya. Proses pelatihan dan pengujian memanfaatkan total 9780 titik data, dengan rasio data pelatihan dan pengujian sebesar 3:1. Secara khusus, 7335 titik data dialokasikan untuk pelatihan, sedangkan 2445 titik data dialokasikan untuk pengujian. Di bawah ini adalah pemaparan awal mengenai temuan penelitian ini.



Gambar 4. 8 Tampilan Dashboard

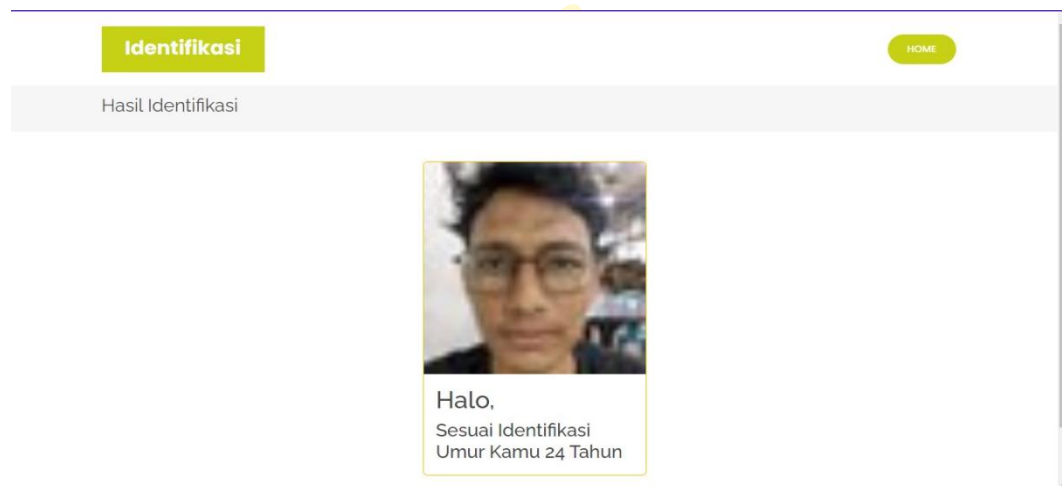
4.2.1 Hasil Model Training

Dataset *training* terdiri dari 7335 titik data. Setelah proses pemodelan jaringan CNN selesai, model akan menjalani pelatihan menggunakan data pelatihan yang telah disiapkan. *Input image* dapat digunakan untuk mencari pasangan citradengan membandingkannya dengan citralain di set pelatihan.

Selama prosedur pelatihan *TensorFlow*, peneliti mencapai akurasi 0,8147 dan kerugian 0,3756 dengan memanfaatkan *epoch* 500 dan *learning rate* 0,001.

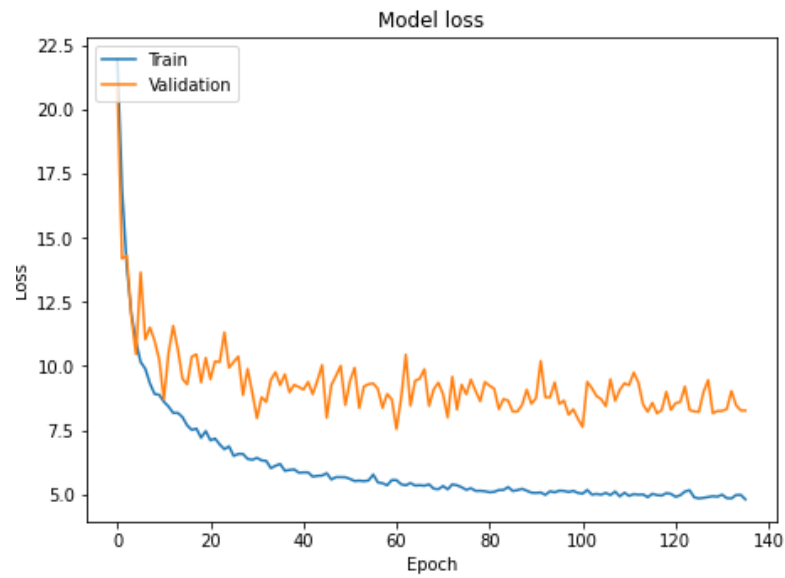
4.2.2 Hasil Data Testing

Proses *testing* menggunakan dataset yang terdiri dari 2445 titik data. Data berikut mewakili hasil dari satu sampel yang diperoleh dari sistem.

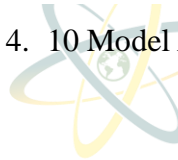


Gambar 4. 9 Tampilan Hasil Sistem

Tahap *testing* sistem ini melibatkan 140 *epoch*, dimana selama itu diperoleh *model loss*.



Gambar 4. 10 Model *Loss* Testing



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN