



## IMPLEMENTASI NEURAL NETWORK UNTUK MENDETEKSI PENYAKIT DIABETES PADA RETINOPATI

Asep Arwan Sulaeman<sup>1</sup>, M Najamuddin Dwi M<sup>2</sup> Supapto<sup>3</sup> Asep Suprianto<sup>4</sup> Suratman<sup>5</sup>

Program Studi Teknik Informatika Fakultas Teknik Universitas Pelita Bangsa

<sup>1</sup>aseparwan@pelitabangsa.ac.id, <sup>2</sup>najamuddin.dwi@pelitabangsa.ac.id, <sup>3</sup>suprapto@pelitabangsa.ac.id  
<sup>4</sup>asep.supriyanto@pelitabangsa.ac.id, <sup>5</sup>Suratman@pelitabangsa.ac.id

### Abstrak

Retinopati Diabetes (*Diabetic Retinopathy*) adalah penyakit yang dapat terjadi pada mata karena gangguan oleh penyakit diabetes. Gejala yang terjadi pada retinopati diabetes yaitu munculnya pembengkakan pada pembuluh darah. Penyakit ini menyebabkan saluran darah menjadi tersumbat dan mudah bocor sehingga menyebabkan titik-titik kemerahan pada retina. Proses deteksi untuk penyakit ini sudah bisa dilakukan dengan fundus camera. Namun untuk pemeriksaan tersebut masih menggunakan cara manual dengan mengamati langsung dan mengevaluasi foto fundus camera dan dokter sebagai pengambil keputusannya.

Tujuan dari penelitian ini yaitu dapat mendeteksi gejala penyakit ini lebih dini dengan cara menerapkan metode yang dapat menghasilkan akurasi yang tinggi dengan bantuan komputer melalui metode *neural network*. Data penelitian diperoleh penulis dari situs pengkodean Kaggle (<https://www.kaggle.com>) dimana setiap gambar mempunyai klasifikasi, yaitu: mata normal (*without DR*), mata dengan DR ringan, mata dengan DR sedang, mata dengan DR berat, dan mata dengan DR sangat berat. Data tersebut selanjutnya diproses dengan menggunakan *tools* MATLAB untuk *preprocessing* gambar dan Tensorflow yang digunakan untuk mencari bobot akurasi. Hasil dari penelitian menunjukkan bahwa dengan 1200 kali eksperimen dihasilkan bobot akurasi rata-rata sebesar 88.61% dengan standard deviasi sebesar 6.41% dengan *range* bobot akurasi tertinggi sebesar 93.01% dan bobot akurasi terendah sebesar 50.18%.

Kata kunci: retinopati diabetes, *image processing*, *neural network*

### Abstract

*Diabetic Retinopathy (Diabetic Retinopathy) is a disease that can occur in the eye due to interference by diabetes. Symptoms that occur in diabetic retinopathy are the appearance of swelling in the blood vessels. This disease causes blood vessels to become clogged and leak easily, causing reddish spots on the retina. The detection process for this disease can already be done with a fundus camera. However, this examination still uses the manual method by directly observing and evaluating fundus camera photos and the doctor as the decision maker.*

*The aim of this research is to be able to detect the symptoms of this disease earlier by applying a method that can produce high accuracy with the help of a computer through the neural network method. The research data was obtained by the authors from the Kaggle coding site (<https://www.kaggle.com>) where each image has a classification, namely: normal eyes (*without DR*), eyes with mild DR, eyes with moderate*

*DR, eyes with severe DR, and eyes with very heavy DR. The data is then processed using MATLAB tools for image preprocessing and Tensorflow which is used to find accuracy weights. The results of the study show that 1200 experiments resulted in an average accuracy weight of 88.61% with a standard deviation of 6.41% with a range of the highest accuracy weight of 93.01% and the lowest accuracy weight of 50.18%.*

*Keywords: diabetic retinopathy, image processing, neural network*

### 1. Pendahuluan

*Diabetic Retinopathy (DR) adalah komplikasi mikro vascular oleh penyakit diabetes. Gejala penyakit ini dapat menyebabkan penglihatan mata menurun dan bahkan dapat menyebabkan kebutaan. Retinopati diabetes menurut AAO (American Academy Ophthalmology) dan InaDRS terbagi menjadi 5 klasifikasi, yaitu: mata normal (*without DR*), mata dengan DR ringan (*Mild DR*), mata dengan DR sedang*

(Moderate DR), mata dengan DR berat (Severe DR), dan mata dengan DR sangat berat (Proliferative DR) [1]. Pada umumnya dokter spesialis mata mendeteksi tanda-tanda tersebut dengan mengamati langsung dari mata dengan melakukan pemeriksaan fundus. Pemeriksaan tersebut memiliki sensitivitas dan spesifitas yang lebih superior untuk mendiagnosis retinopati diabetes dibandingkan dengan oftalmoskopi langsung dan tidak langsung yang dilakukan oleh oftalmologis. Kebutuhan akan metode penyingkapan DR yang komprehensif dan otomatis telah lama dikenal, dan upaya sebelumnya telah menghasilkan kemajuan yang baik dengan menggunakan klasifikasi citra, pengenalan pola, dan machine learning. Pembelajaran mesin (*machine learning*) dapat didefinisikan sebagai proses dimana sistem dapat meningkatkan performa dari pengalaman. Terdapat tiga jenis *machine learning* yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Salah satu *task* dalam *supervised learning* adalah klasifikasi, dimana ini menjadi fokus permasalahan dari penelitian ini. Oleh karena itu berdasarkan uraian diatas, peneliti ingin mengusulkan untuk menguji metode *neural network* untuk menentukan tingkat kesehatan seseorang jika dilihat dari sudut pandang penyakit retinopati diabetes berdasarkan citra fundus kamera dengan menggunakan algoritma untuk mendeteksi penyakit diabetes yaitu algoritma *neural network* dari data situs pengkodean Kaggle untuk mengetahui bobot akurasi yang dihasilkan.

Berdasarkan hasil penelitian yang sudah dilakukan ada beberapa hasil akurasi yang didapatkan dalam penelitian sejenis dengan menggunakan beberapa metode yang berbeda. Berikut data yang diambil dari beberapa jurnal yang terindeks scopus yaitu Navoneel Chakrabarty [2] telah melakukan penelitian yang sejenis menggunakan *Convolutional Neural Network* dengan data *sample* sebanyak 24 *sample* sebagai data *training* dan 6 *sample* sebagai data *testing* dengan hasil akurasi sebesar 91,67%. K. R. Ananthapadmanaban dan G. Pathiban [3] telah meneliti dengan menggunakan *naïve bayes* dan *support vector machine* untuk memprediksi deteksi dini retinopati diabetes dengan hasil 83,37% untuk *naïve bayes* dan 64,91% untuk *support vector machine*. P. V. Rao, Gayathri. R, dan Sunitha. R [4] telah meneliti dengan menggunakan dua metode yang berbeda, yaitu *naïve bayes* dengan hasil akurasi sebesar 89,6 % dan *multilayer perceptron back propagation (MLP-BP) Artificial Neural Network (ANN)* sebesar 90,6%. Cut Fiarni dkk [5] menggunakan metode *naïve bayes* untuk mengklasifikasikan retinopati diabetes dengan menggunakan data sebanyak 158 dataset dan menghasilkan akurasi sebesar 68%.

## 2. Landasan Pemikiran

### 2.1 Pengolahan Citra Digital

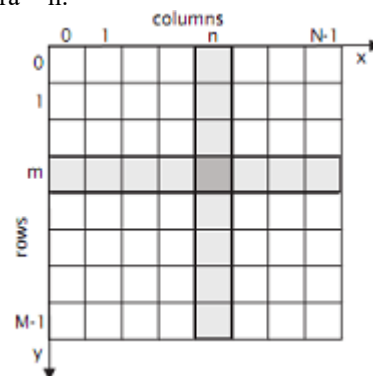
Pada dasarnya, indera yang dimiliki manusia adalah peralatan sensor alami yang paling canggih. Mata kita dapat mengenali bentuk obyek tertentu yang dapat direkam otak dalam mengidentifikasi ciri-ciri

dari obyek yang dilihat. Pada dasarnya proses tersebut adalah proses pembelajaran. Semakin sering kita melihat obyek yang kita lihat, maka semakin cepat kita mengenali obyek tersebut [6]. Citra digital merupakan suatu fungsi yang dinyatakan dalam dua dimensi  $f(x,y)$  dengan  $x$  maupun  $y$  adalah posisi koordinat dan  $f$  merupakan amplitudo pada posisi  $(x,y)$  yang dikenal sebagai intensitas atau *grey scale* [6]. Nilai dari *grayscale* berbentuk diskrit mulai dari 0 sampai dengan 255. Citra yang didapat oleh kamera dan sudah dikuantisasi ke bentuk nilai diskrit disebut sebagai citra digital. Citra digital tersusun oleh sejumlah nilai keabuan yang disebut sebagai piksel (*pixel*). Untuk melakukan pemrosesan citra digital, maka citra analog harus dikonversi terlebih dahulu kedalam citra digital. Suatu citra digital dapat dinyatakan dengan persamaan 1.1 sebagai berikut.

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix}$$

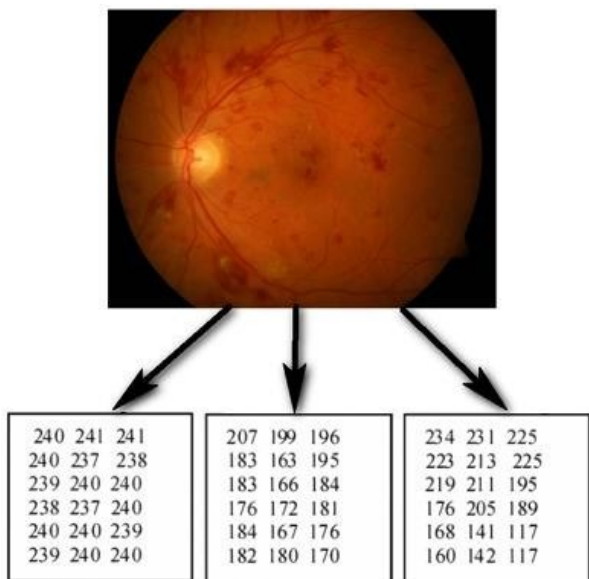
Gambar 1. Persamaan 1.1 [7]

Dari persamaan 1.1, citra digital dapat dinyatakan sebagai matrik dengan tingkat citra =  $m$  dan lebar citra =  $n$ .



Gambar 2. Representasi Citra Digital dalam 2 dimensi [7]

Salah satu aplikasi untuk pengolahan citra digital yaitu *color image* dimana gambar berwarna yang memiliki warna tertentu, warna tersebut adalah merah (*red*), hijau (*green*), dan biru (*blue*). Jika masing-masing warna memiliki range 0-255, maka totalnya adalah  $255^3=16.581.375$  (16K) variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap piksel, gambar tersebut juga disebut gambar bit warna. *Color image* ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau, dan biru untuk setiap pikselnya, seperti yang ditunjukkan pada gambar berikut.

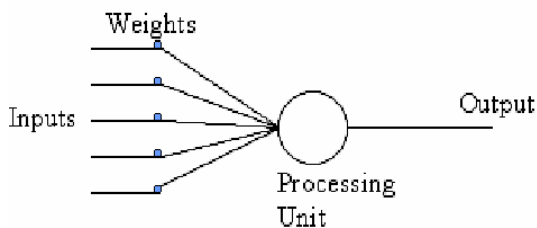


Gambar 3. Representasi Citra dalam 3 dimensi.

2.2 Neural Network

Neural network atau jaringan syaraf tiruan adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk menstimulasikan proses pembelajaran pada otak manusia tersebut. Neural network berusaha meniru arsitektur atau struktur cara kerja otak manusia yang berguna dan mampu menggantikan pekerjaan manusia di beberapa sektor. Contoh pekerjaan yang diharapkan mampu di selesaikan oleh neural network seperti mengenali pola, prediksi, klasifikasi, dan pendekatan fungsi [8]. Artificial Neural Network (ANN) sering kali digunakan sebagai tools yang sukses dalam menangani banyak masalah seperti pattern recognition, berbagai masalah klasifikasi, berbagai masalah regresi, dan persamaan differensial [8]. Jaringan syaraf tiruan memiliki 3 lapisan, yaitu lapisan input, lapisan tersembunyi atau hidden layer dan lapisan output [8].

Berikut merupakan gambar bentuk dari dasar sebuah neuron.



Gambar 4. Bentuk Dasar Neuron.

3. Metode Penelitian

Berikut langkah atau proses penelitian ini dengan menggunakan neural network dengan menggunakan Tensorflow untuk mendapatkan akurasi yang maksimal dari dataset tersebut.



Gambar 5. Langkah Penelitian

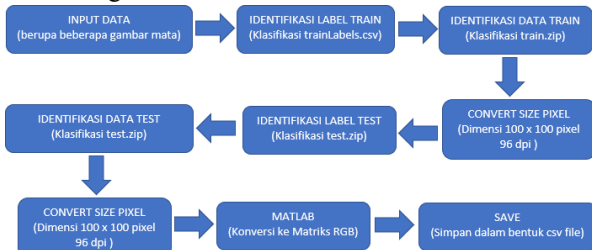
3.1. Input

Input yang digunakan berupa beberapa gambar mata berupa dataset klasifikasi yang didapat dari situs pengkodean Kaggle. Data tersebut terdiri dari:

- a. train.zip, sebagai data training (pelatihan) dimana sample yang digunakan sebanyak 35.126 sample untuk mata kiri dan mata kanan.
- b. test.zip, sebagai data testing sebanyak 53.576 sample berupa gambar mata kiri dan mata kanan.
- c. sample.zip, sebagai data sample sebanyak 5 sample berupa gambar mata kiri dan 5 sample berupa gambar mata kanan (data ini tidak digunakan karena tidak disebutkan untuk klasifikasi retinopati diabetes).
- d. sampleSubmissin.csv, sebagai data sample dengan format excel yang berisi label dari gambar mata kiri dan mata kanan dengan level 0 (mata normal) sebanyak 53.576 sample (data ini tidak digunakan karena hanya level 0 saja yang disebutkan).
- e. trainLabels.csv, sebagai data label dengan format excel yang berisi label dari gambar mata kiri dan mata kanan dengan level sebagai berikut:
  - 1) 0 (mata normal/ without DR) sebanyak 25810 sample.
  - 2) 1 (mata dengan DR ringan/ Mild DR) sebanyak 2443 sample.
  - 3) 2 (mata dengan DR sedang/ Moderate DR) sebanyak 5292 sample
  - 4) 3 (mata dengan DR berat/ Severe DR) sebanyak 873 sample
  - 5) 4 (mata dengan DR sangat berat/ Proliferative DR) sebanyak 708 sample.

### 3.2. Preprocessing Data

Pada tahap ini dilakukan beberapa proses pemilahan dengan cara manual dan menggunakan tools yang dibuat dan juga *tools* dari MATLAB yang bertujuan untuk meningkatkan kualitas citra pada gambar yang akan dianalisa sehingga dapat memudahkan untuk diinterpretasikan. Tahapan yang dilakukan dalam preprocessing ini dapat digambarkan dalam diagram berikut.



Gambar 6. Alur preprocessing

### 3.3. Classifier

Pada tahap ini, data yang akan diproses sebagai berikut:

No.	Klasifikasi DR	Mata Kiri	Mata Kanan
1	Mata Normal ( <i>without</i> DR)	12.871 sample	12.939 sample
2	Mata DR ringan ( <i>Mild</i> DR)	1.212 sample	1.231 sample
3	Mata DR sedang ( <i>Moderate</i> DR)	2.702 sample	2.590 sample
4	Mata DR berat ( <i>severe</i> DR)	425 sample	448 sample
5	Mata DR sangat berat ( <i>Proliferative</i> DR)	353 sample	355 sample

Gambar 7. Data Klasifikasi DR

Dari tabel diatas didapatkan kesimpulan bahwa data untuk mata dengan DR sangat berat (*Proliferative* DR) untuk mata kiri yang paling sedikit sampelnya sebanyak 353 sample. Peneliti membagi sama berat datanya untuk semua klasifikasi mata sehingga yang bisa digunakan untuk penelitian ini sebanyak 353 *sample* dataset (mengacu pada jumlah *sample* data yang paling sedikit pada *Proliferative* DR pada mata kiri) dengan perincian untuk pengambilan data *training* sebanyak 300 *sample* dan data *testing* sebanyak 53 *sample* dalam penelitian ini sebagai berikut.

No.	Klasifikasi DR	Mata Kiri	Mata Kanan
1	Mata Normal ( <i>without</i> DR)	300	300
2	Mata DR ringan ( <i>Mild</i> DR)	300	300
3	Mata DR sedang ( <i>Moderate</i> DR)	300	300
4	Mata DR berat ( <i>severe</i> DR)	300	300
5	Mata DR sangat berat ( <i>Proliferative</i> DR)	300	300
Total Data ( <i>sample</i> )		1.500	1.500

Gambar 8. Data Training Klasifikasi DR

No.	Klasifikasi DR	Mata Kiri	Mata Kanan
1	Mata Normal ( <i>without</i> DR)	53	53
2	Mata DR ringan ( <i>Mild</i> DR)	53	53
3	Mata DR sedang ( <i>Moderate</i> DR)	53	53
4	Mata DR berat ( <i>severe</i> DR)	53	53
5	Mata DR sangat berat ( <i>Proliferative</i> DR)	53	53
Total Data ( <i>sample</i> )		265	265

Gambar 9. Data Testing Klasifikasi DR

Total data yang akan digunakan dalam penelitian ini akan mengambil data sebanyak 3.530 *sample* untuk mata kiri dan mata kanan untuk semua klasifikasi retinopati diabetes dengan dimensi 100x100 *pixel*. Data *training* akan mengambil sebanyak 300 *sample* sehingga total data keseluruhan untuk data *training* sebanyak 1500 *sample* untuk mata kiri dan 1500 *sample* untuk mata kanan. Data *testing* mengambil dari data *training* sebanyak 53 *sample* sehingga total data keseluruhan untuk data *testing* sebanyak 265 *sample* untuk mata kiri dan 265 *sample* untuk mata kanan. Data gambar tersebut akan di *preprocessing* dengan klasifikasi normal yang akan di *training* menggunakan *tensor flow*. Proses ini biasa disebut proses pelatihan *network* yang pada dasarnya merupakan proses penyesuaian bobot-bobot untuk masing – masing simpul antara lapisan input, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Penyesuaian bobot dilakukan secara terus-menerus sampai dicapai error yang paling minimum. Satu gambar mewakili dimensi 100x100 *pixel* sehingga jika dibuat matriks 1 baris menjadi 10.000. Karena gambar merupakan gambar berwarna yang mewakili RGB, maka 10.000 mewakili dimensi untuk matrik R, 10.000 mewakili untuk matrik G dan 10.000 mewakili untuk matrik B sehingga untuk satu gambar mewakili matrik RGB menjadi 30.000. Arsitektur *neural network* yang digunakan dalam penelitian ini sebagai berikut.

No	Arsitektur Neural Network	Jumlah
1	Input Neuron	30.000
2	Hidden Layer	2
3	Output Neuron	2

Gambar 10. Arsitektur Neural Network.

Data input yang digunakan sebagai pola dalam pelatihan *Neural Network* sebanyak 30.000 buah yang terdiri atas beberapa ciri statistik dan bias. Data input tersebut yaitu:

X1 = Mata Kiri

X2 = Mata Kanan

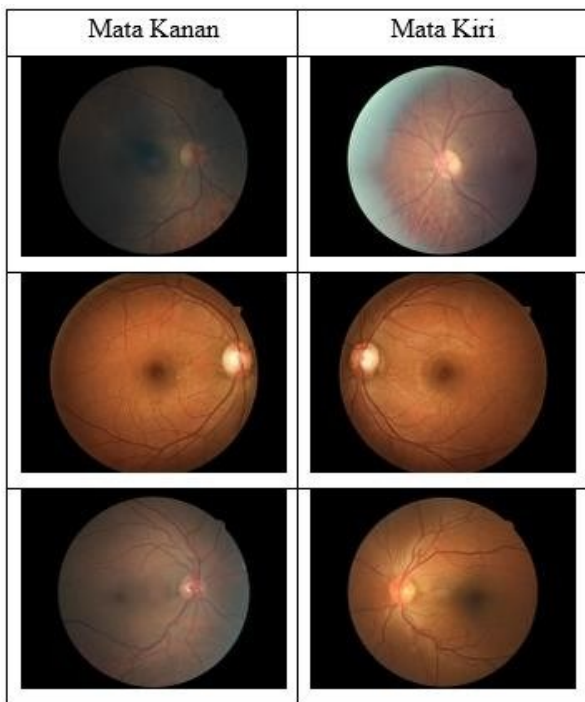
Jumlah *hidden layer* : 2 lapisan tersembunyi, dengan jumlah *neuron* dalam lapisan tersembunyi divariasikan antara 1 – 30.000 *neuron* pada setiap *hidden layer*.

### 3.4. Pengumpulan Data

Pengumpulan dataset untuk penelitian ini diambil langsung oleh peneliti dari situs pengkodean Kaggle (<https://www.kaggle.com>). Adapun data yang digunakan untuk data *training* sebanyak 3000 *sample* (mata kiri dan mata kanan) dan data *testing* sebanyak 530 *sample* (mata kiri dan mata kanan) dengan detail sebagai berikut :

#### 3.4.1. Mata Normal (*without* DR)

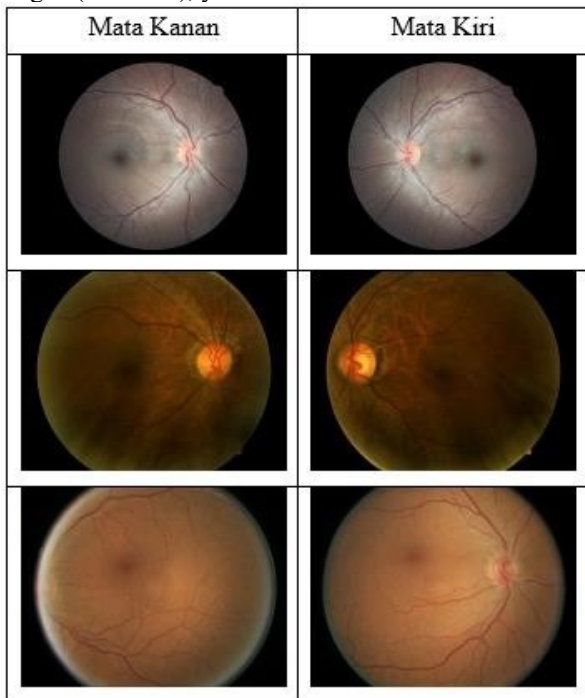
Berikut contoh gambar mata dengan kondisi normal (*without* DR), yaitu:



Gambar 11. Contoh Mata Normal (*without DR*)

3.4.2. Mata dengan DR Ringan (*Mild DR*)

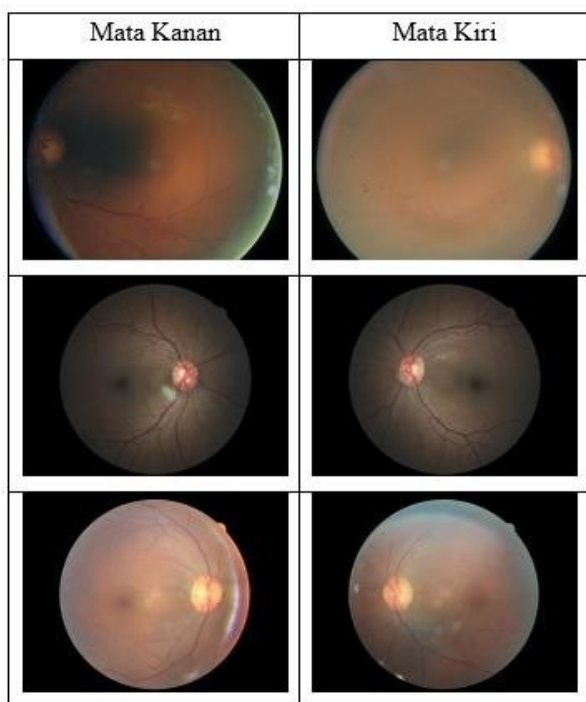
Berikut contoh gambar mata dengan DR ringan (*Mild DR*), yaitu:



Gambar 12. Contoh Mata dengan DR Ringan

3.4.3. Mata dengan DR Sedang (*Moderate DR*)

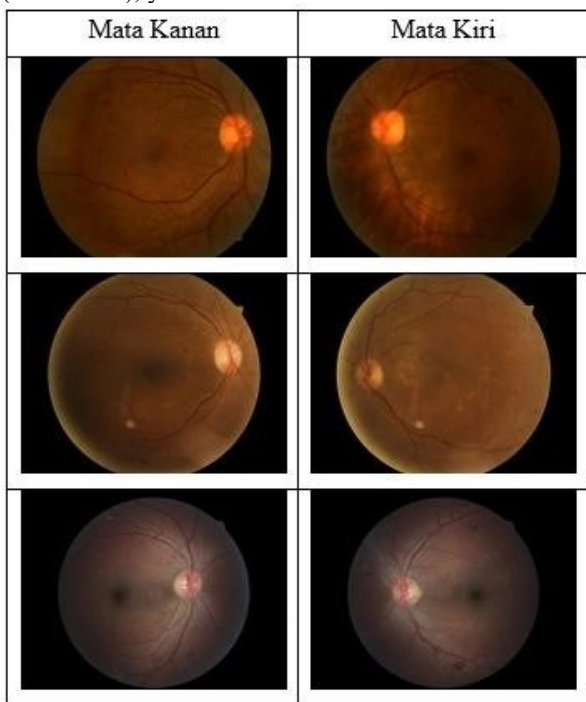
Berikut contoh gambar mata dengan DR sedang (*Moderate DR*), yaitu:



Gambar 13. Contoh Mata dengan DR sedang

3.4.4. Mata dengan DR Berat (*Severe DR*)

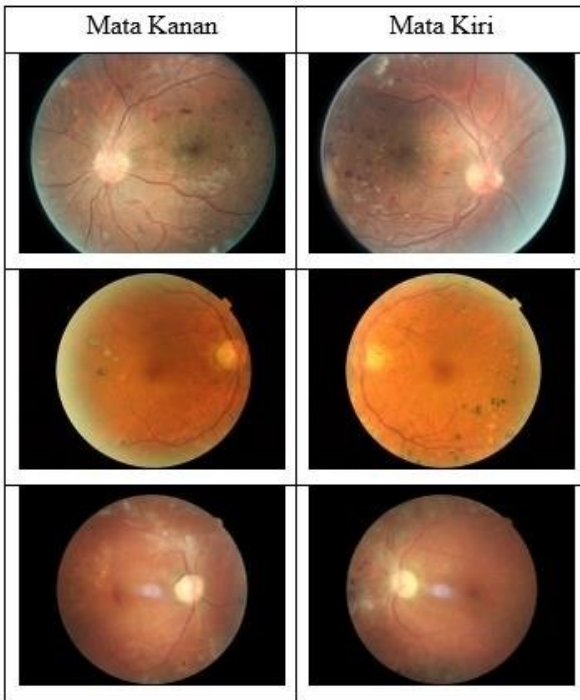
Berikut contoh gambar mata dengan DR berat (*Severe DR*), yaitu:



Gambar 14. Contoh Mata dengan DR berat

3.4.5. Mata dengan DR sangat berat

Berikut contoh gambar mata dengan DR sangat berat (*Proliferative DR*), yaitu:



Gambar 13. Contoh Mata dengan DR sangat berat

#### 4. Pembahasan

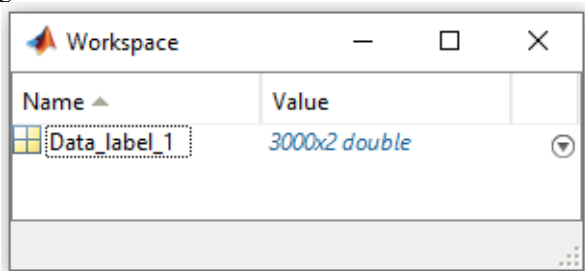
Matriks label yang dibuat berupa matriks yang akan digunakan untuk data *training* dan data *testing* dimana matriks yang dibuat merupakan matriks diagonal yang disesuaikan dengan klasifikasi dari retinopati diabetes. Untuk membuat matriks label dapat dibuat dengan menggunakan Matlab dengan membuat modul dengan *script* sebagai berikut.

```
function Y =
matrix_diagonal_save(Banyak_matrix_diagonal,banyak_jumlah_gambar,Save_nama_C
SV)
% Asep arwan sulaeman
YX= eye(Banyak_matrix_diagonal);
for i=1:banyak_jumlah_gambar
    if i == 1
        Y = {YX};
    else
        Y = [Y;YX];
    end
end
csvwrite(Save_nama_CSV,Y)
```

Kemudian untuk proses eksekusi modul tersebut, diberikan *script command* untuk matriks label data *training* sebagai berikut.

```
function Y = Creating_Label_Train_Testing()
Data_label_1 = matrix_diagonal_save(2,1500, 'Label_Train.csv');
end
```

Ketika dieksekusi maka akan membuat file *Label\_Train.csv* dengan matriks 3000x2 seperti gambar berikut.

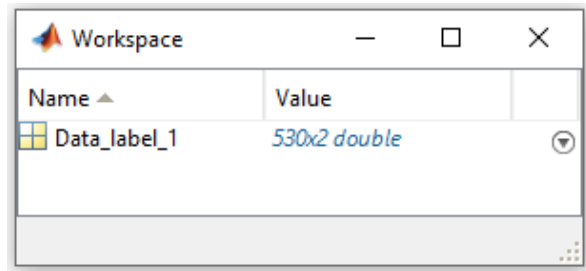


Gambar 14. Matriks label data *training*

Kemudian untuk proses eksekusi modul diatas, diberikan *script command* untuk matriks label data *testing* sebagai berikut.

```
function Y = Creating_Label_Train_Testing()
Data_label_1 = matrix_diagonal_save(2,265, 'Label_Testing.csv');
end
```

Ketika dieksekusi maka akan membuat file *Label\_Testing.csv* dengan matriks 530x2 seperti gambar berikut.



Gambar 15. Matriks label data *testing*

#### 4.1. Konversi Gambar RGB pada Matlab

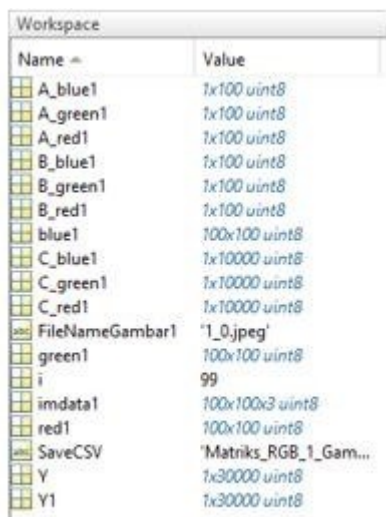
Gambar yang digunakan merupakan gambar berwarna yang terdiri dari elemen-elemen RGB sehingga dalam satu gambar ada 3 dimensi matriks. Dengan menggunakan Matlab, file gambar dapat dijadikan matriks RGB dengan menggunakan modul sebagai berikut.

```
function Y = matrix_1_gambar(FileNameGambar1,SaveCSV)
% Asep arwan sulaeman
imdatal = imread(FileNameGambar1);
red1 = imdatal(:,:,1); % Red channel
green1 = imdatal(:,:,2); % Green channel
blue1 = imdatal(:,:,3); % Blue channel
%disesuaikan dengan dimensi pikselnya, 100x100 = 99
for i=1:99
    if i == 1
        %1
        A_red1 = red1(i,:);
        B_red1 = red1((i+1),:);
        C_red1 = horzcat(A_red1,B_red1);
        A_green1 = green1(i,:);
        B_green1 = green1((i+1),:);
        C_green1 = horzcat(A_green1,B_green1);
        A_blue1 = blue1(i,:);
        B_blue1 = blue1((i+1),:);
        C_blue1 = horzcat(A_blue1,B_blue1);
    else
        %1
        B_red1 = red1((i+1),:);
        C_red1 = horzcat(C_red1,B_red1);
        B_green1 = green1((i+1),:);
        C_green1 = horzcat(C_green1,B_green1);
        B_blue1 = blue1((i+1),:);
        C_blue1 = horzcat(C_blue1,B_blue1);
    end
end
%1
Y1 = horzcat(C_red1,C_green1);
Y1 = horzcat(Y1,C_blue1);
Y = [Y1];
csvwrite(SaveCSV,Y)
```

Kemudian untuk proses eksekusi modul tersebut dengan menggunakan *command* sebagai berikut.

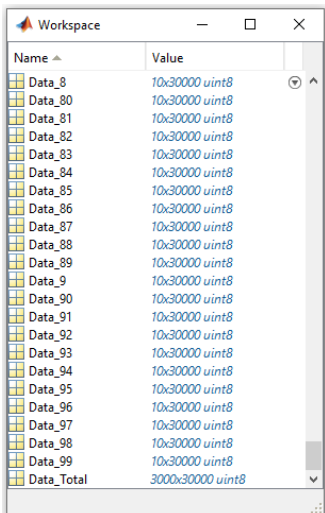
```
function Y = Creating_Matriks_RGB_1_Gambar()
Data_label_1 =
matrix_1_gambar('1_0.jpeg', 'Matriks_RGB_1_Gambar.csv');
end
```

Ketika dieksekusi maka akan membuat file *Matriks\_RGB\_1\_Gambar.csv* dengan keterangan sebagai berikut.



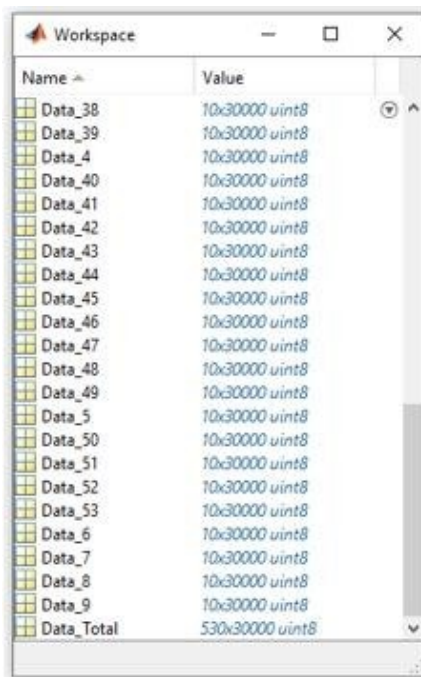
Gambar 16. Hasil matriks yang dihasilkan untuk satu file gambar

Diatas merupakan cara membuat matriks RGB dari satu gambar untuk membuat matriks satu baris sedangkan dalam penelitian ini menggunakan *sample* sebanyak 3000 *sample* untuk data *training* maka *script* untuk satu gambar di modifikasi modul menjadi 10 gambar dengan hasil keterangan matriks 3000x30000 seperti dibawah ini.



Gambar 17. Hasil matriks yang dihasilkan untuk data *training*

Pada data testing yang menggunakan *sample* sebanyak 530 *sample* menghasilkan matriks 530x30000 seperti gambar dibawah ini.

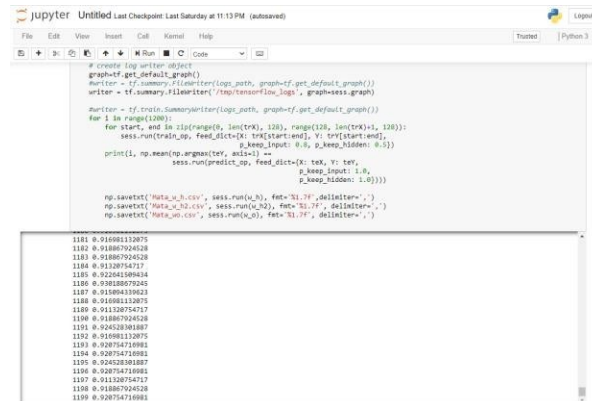


Gambar 18. Hasil matriks yang dihasilkan untuk data testing

Penelitian ini akan menggunakan *sample* sebanyak 3000 *sample* dimana gambar yang digunakan terdiri dari gambar mata normal (*without DR*), mata dengan DR ringan (*Mild DR*), mata dengan DR sedang (*Moderate DR*), mata dengan DR berat (*Severe DR*), dan mata dengan DR sangat berat (*Proliferative DR*) untuk mata kiri dan mata kanan. *Script* untuk menghasilkan data *training* dan data *testing* untuk *sample* sebanyak 3000 *sample* dan 530 *sample* untuk mata kiri dan mata kanan yang dibuat di Matlab merupakan *script* yang dibuat secara terprogram.

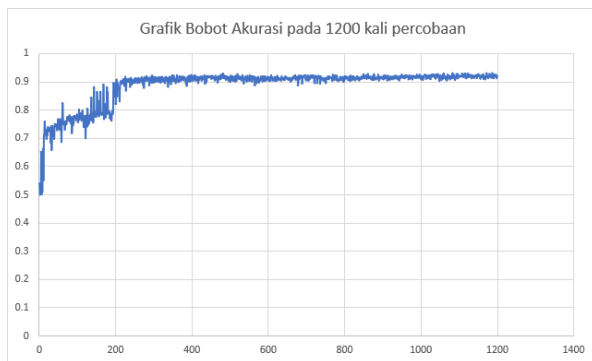
### 4.2. Hasil Pengujian

Berdasarkan hasil dari akurasi yang sudah dijalankan tersebut sudah mendapatkan akurasi yang cukup tinggi yaitu 92% dan untuk hasil tersebut bisa berarti pemodelannya sudah cukup bagus. Berikut merupakan gambar dari hasil percobaan yang dilakukan.



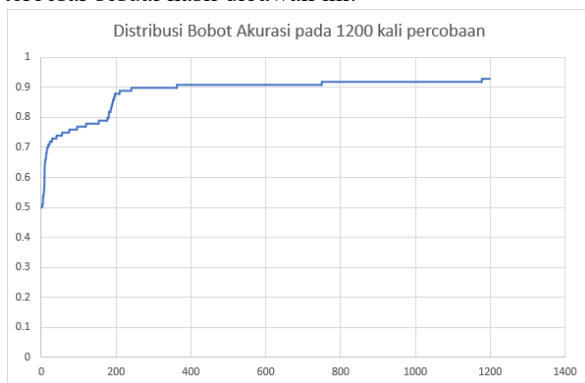
Gambar 19. Hasil percobaan pada Tensorflow

Jika digambarkan dalam bentuk grafik, maka dapat terbentuk data bobot sebagai berikut.



**Gambar 20. Bobot Akurasi**

Untuk memudahkan pengelompokan bobot dalam setiap percobaan maka akan diurutkan dari hasil bobot akurasi tersebut dari yang terkecil ke yang terbesar sesuai hasil dibawah ini.



**Gambar 21. Bobot Akurasi yang diurutkan**

Berdasarkan hasil dari percobaan yang sudah dijalankan tersebut didapatkan akurasi rata-rata sebesar 88.61% dengan standar deviasi sebesar 6.41%, adapun nilai tertinggi akurasi adalah 93.01% dan akurasi terendahnya 50.18%. Jika dilihat dari distribusi data bobot yang tersebar, masih cukup besar peluang untuk meningkatkan bobot akurasi tersebut menjadi lebih tinggi dengan menambahkan perulangan ataupun dengan menambahkan hidden layer namun dalam pecobaannya membutuhkan beberapa hari untuk menyelesaikan hasil percobaan tersebut.

## 5. Penutup

Dari hasil penelitian, didapatkan kesimpulan dan sara, yaitu:

1. Teknik dengan menggunakan metode *neural network* pada Tensorflow menunjukan lebih baik dalam metode pengenalan retinopati diabetes.
2. Pada tahap *preprocessing* sangat menentukan sekali dalam proses penghitungan dan peningkatan nilai akurasi pada algoritma yang digunakan.
3. Hasil bobot akurasi dari 1200 kali percobaan didapatkan rata-rata bobot akurasi sebesar 88.61% dengan standar deviasi sebesar

6.41%, dengan range akurasi tertinggi sebesar 93.01% dan akurasi terendah 50.18%.

4. Dataset yang diambil agar dapat diperbaiki dari dimensi pixel gambar yang lebih spesifik dan gradasi warna atau kecerahan yang lebih seragam dengan konsistensi posisi pengambilan gambar secara tetap guna memberikan kualitas dataset yang lebih baik.

## Daftar Pustaka

- [1] Kaggle Diabetic Retinopathy Detection Competition on <https://www.kaggle.com/c/diabetic-retinopathy-detection>. Access July 2, 2017.
- [2] Navoneel Chakrabarty. A Deep Learning Method for the detection of Diabetic Retinopathy. 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics, and Computer Engineering (UPCON), 2018.
- [3] K. R. Ananthapadmanaban, G. Parthiban. Prediction of Chances-Diabetic Retinopathy using Data Mining Classification Techniques. Indian Journal of Science and Technology, Vol7(10), 1498-1503, October 2014.
- [4] P. V. Rao, Gayathri. R, Sunitha. R. A Novel Approach for Design and Analysis of Diabetic Retinopathy Glaucoma Detection using Cup to Disk Ration and ANN. 2nd International Conference on Nanomaterials and Technologies (CNT 2014), Procedia Materials Science 10(2015) 446-454, 2015.
- [5] Cut Fiarni, Evasaria. M. Sipayung, Siti Maemunah. Analysis and Prediction of Diabetes Complication Disease using Data Mining Algorithm. The Fifth Information System International Conference, Procedia Computer Science 161 (2019) 449-457, 2019.
- [6] Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, " Teori Pengolahan Citra Digital ", Andi Yogyakarta dan UDINUS Semarang, 2009.
- [7] Purnomo, MH., Muntasa, A., 2010, Konsep Pengolahan Citra Digital dan Ekstraksi Fitur. Yogyakarta : Graha Ilmu.
- [8] Shukla, A., Tiwari, R., & Kala, R. Real Life Application of Soft Computing. CRC Press, 2010.