



Perbandingan Kinerja CNN, RNN, dan FNN dalam Klasifikasi Citra Retina untuk Deteksi Penyakit Mata

Hanif Rizaqi^{1*}, Imam Tahyudin²

¹*Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto,
Jl. Letjend Pol. Soemarto No.127, Banyumas 53127, Indonesia*

²*Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto,
Jl. Letjend Pol. Soemarto No.127, Banyumas 53127, Indonesia*

*Email Penulis Koresponden: hanifrizaqii@gmail.com

Abstrak:

Penyakit mata menjadi salah satu penyebab utama gangguan penglihatan yang dapat mengarah pada kebutaan permanen jika tidak ditangani dengan baik. Beberapa penyakit mata yang sering terjadi antara lain katarak, glaukoma, dan retinopati diabetik. Penelitian ini bertujuan untuk membandingkan performa tiga model deep learning yaitu Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), dan Feedforward Neural Network (FNN) dalam klasifikasi citra retina. Dataset yang digunakan terdiri dari 4.217 citra fundus mata yang mencakup empat kelas: normal, katarak, glaukoma, dan diabetic retinopathy. Proses preprocessing mencakup pembagian data, resizing, dan normalisasi. Setiap model dilatih menggunakan teknik early stopping untuk mencegah overfitting. Evaluasi dilakukan menggunakan confusion matrix, akurasi, dan nilai loss. Hasil menunjukkan bahwa model CNN memiliki performa terbaik dengan akurasi sebesar 85% dan nilai loss 0,3841, diikuti oleh RNN 83% dan FNN 78%. Implementasi model dilakukan dalam aplikasi berbasis Flask untuk memberikan prediksi terhadap citra retina yang diunggah pengguna. Penelitian ini menunjukkan bahwa CNN unggul dalam mengekstraksi fitur spasial pada citra fundus, dan menjadi pendekatan yang paling efektif dalam klasifikasi penyakit mata berdasarkan data yang digunakan.

This is an open access article under the [CC BY-NC](#) license



Kata Kunci:

*Penyakit Mata;
Convolutional Neural Network;
Recurrent Neural Network;
Feedforward Neural Network.*

Riwayat Artikel:

Diserahkan 10 Desember 2024

Direvisi 17 Juni 2025

Diterima 17 Juli 2025

DOI:

10.22441/incomtech.v15i2.31261

1. PENDAHULUAN

Penyakit mata, terutama yang disebabkan oleh kelainan pada retina dan saraf optik, menjadi salah satu penyebab utama gangguan penglihatan yang dapat mengarah pada kebutaan permanen. Menurut data dari Badan Kesehatan Dunia (WHO), lebih dari 2,2 miliar orang di seluruh dunia mengalami masalah

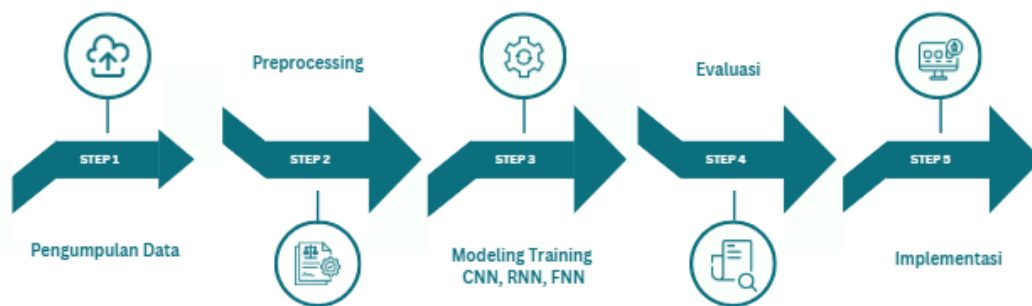
penglihatan, dengan sekitar 1 miliar kasus di antaranya dapat dicegah dan sementara itu, 1 miliar lainnya masih belum dapat diobati[1]. Beberapa kondisi utama yang menyebabkan gangguan penglihatan dan kebutaan mencakup katarak (94 juta orang), kelainan refraksi (88,4 juta orang), glaukoma (7,7 juta orang), retinopati diabetik (3,9 juta orang), dan presbiopia (826 juta orang). Di Indonesia, angka penderita kebutaan akibat penyakit mata masih cukup tinggi, Menurut data Kementerian Kesehatan RI, katarak menjadi penyebab dari 50% kebutaan di Indonesia. Dimana Indonesia menempati posisi kedua di dunia dalam jumlah penderita katarak setelah negara Etiopia[2]. Untuk itu, penerapan teknologi medis yang lebih efisien dalam diagnosa penyakit mata menjadi urgensi dalam mengurangi angka kebutaan dan meningkatkan tingkat keberhasilan.

Model deep learning telah banyak digunakan dalam berbagai bidang, termasuk dalam klasifikasi penyakit mata, berkat kemampuannya untuk menangani data dalam jumlah besar dan kompleks, seperti citra medis. Beberapa model yang dapat digunakan dalam klasifikasi citra seperti Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), dan Feedforward Neural Network (FNN). Convolutional Neural Network (CNN) merupakan model deep learning yang paling banyak digunakan dalam klasifikasi citra medis, model ini memiliki kemampuan untuk mengekstraksi fitur penting dari citra tanpa memerlukan pra-pemrosesan fitur secara manual[3]. Recurrent Neural Network (RNN), meskipun lebih sering diterapkan pada data sekuensial, metode ini juga memiliki potensi dalam pemrosesan citra medis, Salah satu varian dari RNN, yaitu Long Short-Term Memory (LSTM), dapat menangkap dependensi jangka panjang yang ada dalam data citra[4]. Feedforward Neural Network (FNN), meskipun lebih sederhana dibandingkan dengan CNN dan RNN, juga dapat memberikan kontribusi penting dalam klasifikasi citra, terutama jika jumlah fitur tidak terlalu banyak. Dimana metode ini bekerja dengan cara menghubungkan setiap neuron di lapisan input ke neuron di lapisan tersembunyi, dan akhirnya ke lapisan output tanpa adanya umpan balik[5].

Beberapa penelitian yang pernah dilakukan terkait penyakit mata oleh para peneliti diantaranya adalah seperti penelitian yang dilakukan oleh William dan Chairisni Lubis dalam Klasifikasi Penyakit Mata menggunakan CNN menghasilkan akurasi sebesar 91%[6], Kemudian Penelitian Fani Nurona Cahya dkk menggunakan CNN menghasilkan akurasi 98.37%[2]. Penelitian Omar Bernabé dkk meneliti Klasifikasi Penyakit Mata pada Gambar Fundus dengan algoritma CNN menghasilkan Akurasi 99.89%[7]. Penelitian yang dilakukan oleh Masum Shah Junayed dkk dalam mendeteksi Katarak menggunakan algoritma CNN menghasilkan 99.13%[8]. Kemudian penelitian yang dilakukan oleh Suvajit Dutta dkk yang bertujuan untuk mengidentifikasi dan mengklasifikasikan gambar retinopati diabetik menggunakan berbagai model, seperti Feed Forward Neural Network (FNN), Deep Neural Network (DNN), Fuzzy C-Means Clustering (FCM), dan Convolutional Neural Network (CNN), metode CNN lebih baik dalam hasil pelatihan mampu menghasilkan akurasi 72,5%[9]. Sebagian besar penelitian masih terfokus pada penggunaan CNN untuk deteksi penyakit mata, sementara potensi kombinasi antara CNN, Recurrent Neural Networks (RNN), dan FNN untuk klasifikasi citra penyakit mata belum dieksplorasi secara maksimal. Pada penelitian ini, bertujuan untuk mengembangkan dan membandingkan model CNN, RNN, dan FNN untuk klasifikasi citra penyakit mata.

2. METODE

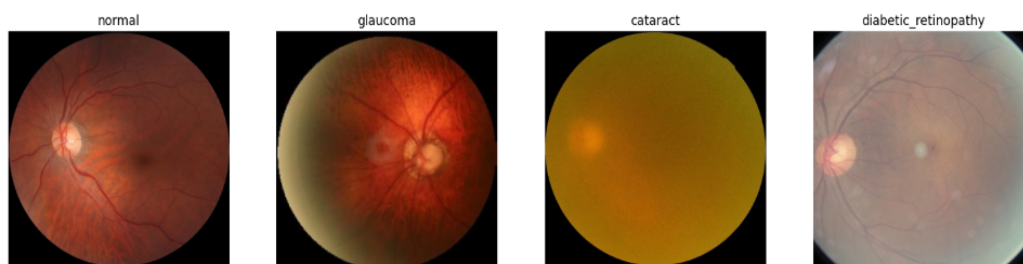
Pada penelitian ini menggunakan 3 metode Deep Learning yaitu CNN, RNN, dan FNN untuk melakukan pengujian. Metode penelitian kali ini dimulai dari Pengumpulan dataset, kemudian Preprocessing, Modeling training, Evaluasi, Serta tahap terakhir Implementasi seperti digambarkan pada gambar 1 alur penelitian dibawah.



Gambar 1. Alur Penelitian

2. 1. Pengumpulan Data

Datasheet yang digunakan pada penelitian ini adalah datasheet berupa citra image/gambar terkait penyakit mata. Data ini diambil dari website kaggle: <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>. dengan jumlah total 4217 data gambar dengan format seperti jpg dan png. Data ini terdiri dari 4 kelas yaitu katarak, glaucoma, diabetic_retinopathy dan normal. Contoh image yang digunakan dapat dilihat pada gambar 2.



Gambar 2. Contoh datasheet citra setiap kelas

2. 2. Preprocessing

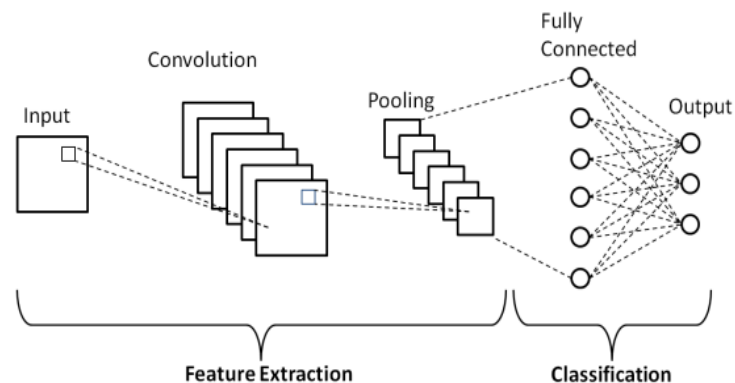
Salah satu langkah penting dalam pengolahan data adalah preprocessing. Terkadang, data dapat mengandung berbagai masalah yang dapat memengaruhi hasil pemrosesan. Untuk mengatasi hal ini, langkah preprocessing dilakukan guna memperbaiki dan mempersiapkan data agar dapat diproses dengan lebih optimal[10]. Sebelum memasuki tahap klasifikasi, analisis data awal dilakukan pada tahapan ini. Proses preprocessing mencakup pembagian dataset menjadi data pelatihan, validasi, dan pengujian, serta dilakukan augmentasi dan normalisasi data. Tujuan dari preprocessing ini adalah untuk mempersiapkan data agar diharapkan dapat digunakan secara optimal oleh model CNN, RNN, dan FNN dalam klasifikasi penyakit mata.

2. 3. Modeling Training

Tahapan modeling pada penelitian ini datasheet citra penyakit mata dilakukan training dengan menggunakan model CNN, RNN, dan FNN. Model-model tersebut dilatih dengan data training serta menggunakan *callback* seperti EarlyStopping yang kemudian digunakan metrik seperti loss dan akurasi. Model dilatih guna mengenali pola dan fitur penting dari citra untuk mengklasifikasikan penyakit mata ke dalam kategori yang telah ditentukan, seperti normal, katarak, glaukoma, dan retinopati diabetik. Hasil model training disimpan dalam format .h5 untuk digunakan lebih lanjut dalam tahap implementasi pengujian.

a. Convolutional Neural Network (CNN)

Jaringan Saraf Convolutional (CNN) merupakan jenis jaringan saraf tiruan yang dirancang khusus untuk menangani data yang berbentuk grid, seperti citra. CNN sangat efisien dalam mengekstraksi fitur spasial dari gambar, memungkinkan model ini untuk secara otomatis mengenali pola, objek, atau tekstur dalam citra. Struktur model CNN terdiri dari beberapa lapisan, antara lain convolutional layer, activation function, pooling layer, flat layer, dan fully connected layer. Setiap lapisan ini memiliki peran tertentu untuk memproses dan mengolah informasi visual secara bertahap[11]. CNN bekerja dengan mengaplikasikan operasi konvolusi pada citra input untuk menghasilkan fitur-fitur yang relevan, kemudian meneruskan hasilnya melalui lapisan-lapisan jaringan untuk klasifikasi atau deteksi. CNN memiliki kemampuan untuk mengekstrak fitur secara otomatis, mengurangi kebutuhan akan teknik ekstraksi fitur manual, serta meningkatkan akurasi dalam tugas-tugas klasifikasi citra[12].

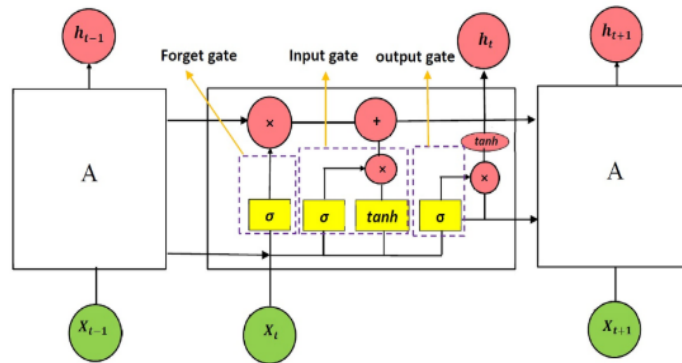


Gambar 3. Arsitektur Convolutional Neural Network [13]

b. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) merupakan jenis algoritma yang pertama kali diperkenalkan pada tahun 1980-an, RNN mengalami pengembangan, salah satunya melalui Long Short-Term Memory (LSTM). Cara kerja RNN dengan mengolah input secara sekuensial, dimana hasil keputusan melibatkan loop yang menghubungkan dipengaruhi oleh informasi sebelumnya, hal ini terjadi karena RNN memiliki memori internal yang memungkinkan untuk mengingat informasi lampau [14]. RNN dapat digabungkan dengan Convolutional Neural Networks (CNN) untuk memperluas cakupan analisis piksel dalam citra. Pendekatan pelatihan yang digunakan untuk RNN, khususnya pada komponen LSTM.

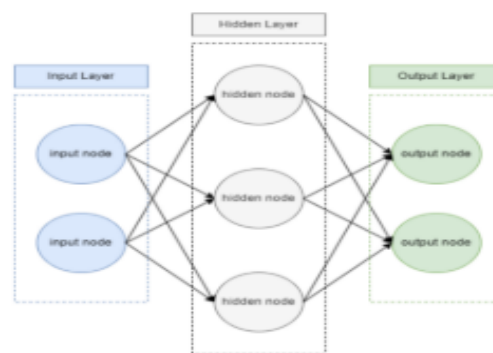
Arsitektur LSTM terdiri dari menggunakan tiga gerbang (forget, input, dan output) untuk mengelola aliran informasi secara dinamis. Gerbang forget adalah fungsi yang memutuskan informasi mana yang harus dipertahankan dan mana yang harus dilupakan. Gerbang input memutuskan informasi baru apa yang akan disimpan di setiap sel. Gerbang output memilih informasi yang relevan dari status sel saat ini dan menampilkannya sebagai output [15].



Gambar 4. Arsitektur RNN “Long Short Term Memory (LSTM)” [15]

c. Feedforward Neural Network (FNN)

Feedforward Neural Network (FNN) adalah jenis jaringan syaraf tiruan yang paling sederhana dan paling dasar, di mana informasi mengalir satu arah dari input ke output tanpa adanya koneksi balik (*no feedback loops*) [16]. FNN memproses data secara maju melalui sejumlah lapisan (layers) yaitu lapisan input, satu atau lebih lapisan tersembunyi (hidden layers), dan lapisan output. Model ini beroperasi dengan menerima vektor dimulai dari lapisan input, kemudian melalui lapisan tersembunyi (hidden layers) yang berinteraksi satu sama lain, hingga menghasilkan keluaran pada lapisan output [17]. Proses ini terstruktur dalam beberapa lapisan, di mana input diproses secara berurutan dari satu lapisan ke lapisan berikutnya.



Gambar 5. Arsitektur Feedforward Neural Network [17]

2. 4. Evaluasi

Pada tahap evaluasi model, setelah proses pelatihan selesai, model diuji menggunakan data uji untuk mengevaluasi performanya. Confusion matrix

digunakan untuk mengevaluasi kinerja model klasifikasi dengan memperlihatkan empat komponen:

- True Positive (TP): Data dengan label positif yang berhasil diprediksi sebagai positif oleh model.
- True Negative (TN): Data dengan label negatif yang tepat diprediksi sebagai negatif.
- False Positive (FP): Data dengan label negatif yang keliru diprediksi sebagai positif.
- False Negative (FN): Data dengan label positif yang salah diklasifikasikan sebagai negatif.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 6. Confusion Matrix

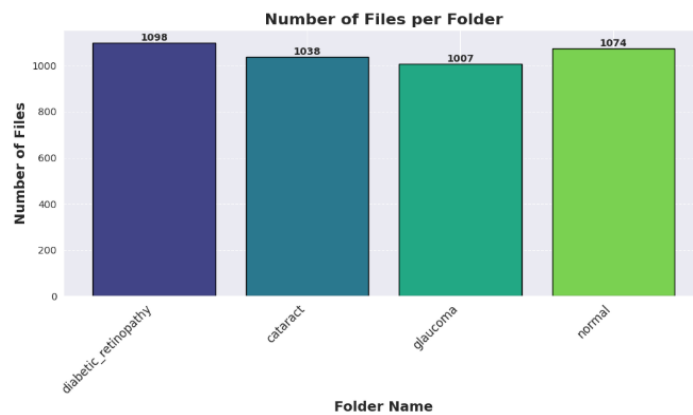
2. 5. Implementasi

Pada tahapan implementasi, model yang telah dilatih dan disimpan diintegrasikan ke dalam aplikasi berbasis Flask untuk melakukan prediksi terhadap gambar yang diunggah oleh pengguna. Flask, sebagai framework web ringan berbasis Python, memungkinkan pembuatan aplikasi web yang sederhana namun fungsional. Dalam aplikasi ini, pengguna dapat mengunggah gambar, dan sistem akan memproses gambar tersebut untuk memberikan prediksi berdasarkan klasifikasi yang telah dipelajari sebelumnya. Flask memfasilitasi komunikasi antara antarmuka pengguna dan model yang telah dilatih, sehingga hasil prediksi dapat ditampilkan secara real-time di aplikasi web.

3. HASIL DAN PEMBAHASAN

3. 1. Dataset

Dataset yang digunakan dalam penelitian ini berisikan 4217 gambar retina, yang yang terbagi menjadi 4 jenis penyakit mata dengan jumlah data normal 1074, data glaucoma 1007, karatak 1038, dan diabetic retinopathy 1098 data seperti pada gambar 7.



Gambar 7. Jumlah data tiap kelas

3. 2. Preprocessing

Pada tahap preprocessing data, langkah pertama yang dilakukan adalah membagi dataset. Dataset yang berjumlah 4.217 data dibagi menjadi tiga bagian dengan proporsi 70% untuk training set, 15% untuk validation set, dan 15% untuk test set. Training set digunakan untuk melatih model, validation set untuk memantau kinerja model selama proses pelatihan, dan test set digunakan untuk menilai kinerja model pada data yang tidak digunakan selama pelatihan. Pembagian dataset menghasilkan 2.094 data untuk training, 633 data untuk validasi, dan 635 data untuk testing, seperti yang terlihat pada gambar 8.

```
# Function to count images in a directory
def count_images(directory):
    total_images = 0
    for root, _, files in os.walk(directory):
        for file in files:
            if file.lower().endswith(('.png', '.jpg', '.jpeg')):
                total_images += 1
    return total_images

# Count images in each directory
train_count = count_images(train_path)
val_count = count_images(valid_path)
test_count = count_images(directory=test_path)

# Print the counts
print(f"Jumlah data pada train: {train_count}")
print(f"Jumlah data pada validation: {val_count}")
print(f"Jumlah data pada test: {test_count}")

Jumlah data pada train: 2949
Jumlah data pada validation: 633
Jumlah data pada test: 635
```

Gambar 8. Pembagian Datasheet

Kemudian selanjutnya dilakukan penyesuaian gambar (resizing) dan normalisasi untuk meningkatkan kualitas dan variasi data sebelum digunakan dalam pelatihan model. Gambar diubah ukurannya menjadi 224x224 piksel agar konsisten memastikan semua gambar memiliki ukuran yang sama selama pelatihan dengan menggunakan ukuran batch sebesar 32 serta membagi nilai piksel dari rentang 0-255 menjadi [0-1] seperti yang dapat dilihat pada gambar 9 dan hasilnya seperti gambar 10.

```

# Define image size and batch size
IMG_SIZE = (224, 224)
BATCH_SIZE = 32

# Create data generators for image normalization (without augmentation)
train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)

# Train generator (without augmentation)
train_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

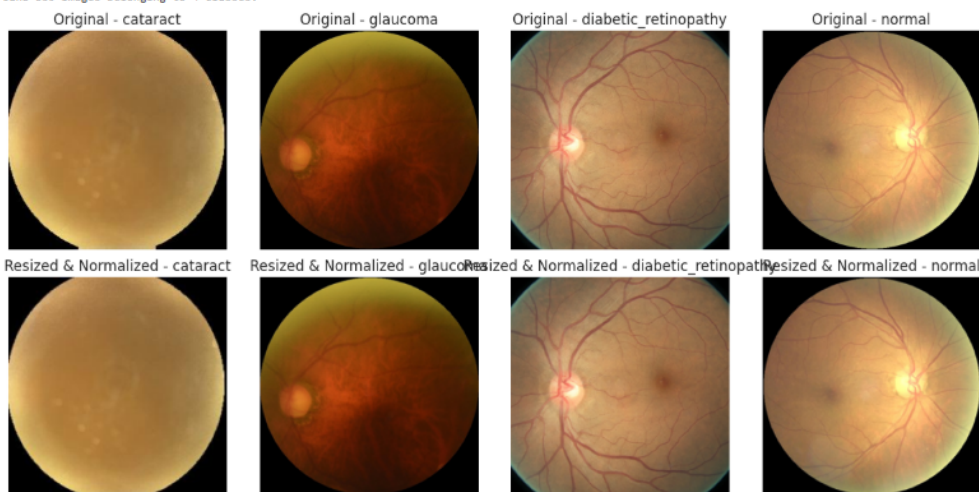
# Validation generator (without augmentation)
val_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)
val_generator = val_datagen.flow_from_directory(
    valid_path,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical'
)

# Test generator (without augmentation)
test_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
    test_path,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False # Important for evaluating test data
)

```

Gambar 9. Resize dan Normalisasi

Found 2949 Images belonging to 4 classes.
 Found 633 Images belonging to 4 classes.
 Found 635 Images belonging to 4 classes.



Gambar 10. Hasil Resize dan Normalisasi

3.3. Model Training

Pelatihan model merupakan tahapan penting dalam pengembangan sistem klasifikasi citra penyakit mata berbasis deep learning. Pada tahap ini, ketiga arsitektur yang digunakan, yaitu CNN, RNN, dan FNN, dilatih untuk mengenali pola dan fitur visual dari citra retina. Proses pelatihan dilakukan menggunakan data training yang telah melalui tahap preprocessing, dan dievaluasi dengan data validasi guna memantau kinerja model selama pelatihan. Pada Gambar 11 adalah tahapan pelatihan model menggunakan arsitektur model CNN dengan total parameter 22,278,084. Selama proses pelatihan, diterapkan teknik Early Stopping untuk memantau kinerja model pada data validasi dan secara otomatis menghentikan

pelatihan jika tidak ada peningkatan kinerja yang signifikan setelah beberapa epoch berturut-turut.

Model: "sequential_5"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_7 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_8 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_8 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_2 (Flatten)	(None, 86528)	0
dense_7 (Dense)	(None, 256)	22,151,424
dense_8 (Dense)	(None, 128)	32,896
dense_9 (Dense)	(None, 4)	516

Total params: 22,278,084 (84.98 MB)
 Trainable params: 22,278,084 (84.98 MB)
 Non-trainable params: 0 (0.00 B)

Gambar 11. Model Training CNN

Gambar 12 dibawah adalah tahapan pelatihan model menggunakan arsitektur model RNN dengan total parameter 22,302,788. Arsitektur model ini menggabungkan CNN dan RNN, khususnya menggunakan lapisan LSTM. Selama proses pelatihan, teknik Early Stopping digunakan untuk memantau kinerja model pada data validasi dan secara otomatis menghentikan pelatihan jika tidak ada peningkatan performa setelah beberapa epoch berturut-turut.

Model: "sequential_7"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_13 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_13 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_14 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_14 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_4 (Flatten)	(None, 86528)	0
dense_12 (Dense)	(None, 256)	22,151,424
reshape_3 (Reshape)	(None, 8, 32)	0
lstm_6 (LSTM)	(None, 8, 64)	24,832
lstm_7 (LSTM)	(None, 64)	33,024
dense_13 (Dense)	(None, 4)	260

Total params: 22,302,788 (85.08 MB)
 Trainable params: 22,302,788 (85.08 MB)
 Non-trainable params: 0 (0.00 B)

Gambar 12. Model Training RNN

Kemudian pada Gambar 13 adalah tahapan pelatihan model menggunakan arsitektur model FNN dengan total parameter 38,576,836. Dalam proses pelatihan ini, diterapkan teknik Early Stopping untuk memonitor kinerja model pada data validasi dan secara otomatis menghentikan pelatihan jika tidak ada perbaikan yang signifikan setelah beberapa epoch berturut-turut.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 150528)	0
dense_4 (Dense)	(None, 256)	38,535,424
dense_5 (Dense)	(None, 128)	32,896
dense_6 (Dense)	(None, 64)	8,256
dense_7 (Dense)	(None, 4)	260

Total params: 38,576,836 (147.16 MB)
 Trainable params: 38,576,836 (147.16 MB)
 Non-trainable params: 0 (0.00 B)

Gambar 13. Model Training FNN

3. 4. Evaluasi

Hasil evaluasi model CNN, RNN, dan FNN dalam mengklasifikasikan empat jenis penyakit mata, yaitu Katarak, Diabetic Retinopathy, Glaukoma, dan Normal, dilakukan menggunakan Confusion Matrix. Matriks ini mengukur empat komponen utama yang mencakup True Positive (TP), False Positive (FP), False Negative (FN), dan True Negative (TN). Penjelasan lebih detail mengenai hasil evaluasi dapat dilihat pada Tabel 1.

Tabel 1. Evaluasi Confusion Matrix

Kelas	Model	TP	FP	FN	TN
Katarak	CNN	114	42	14	423
	RNN	120	36	28	406
	FNN	117	39	82	324
Diabetic Retinopathy	CNN	165	0	3	372
	RNN	165	0	3	361
	FNN	163	2	29	278
Glaucoma	CNN	117	35	36	420
	RNN	98	54	22	428
	FNN	37	115	12	404
Normal	CNN	141	21	45	396
	RNN	143	19	56	383
	FNN	124	38	71	317

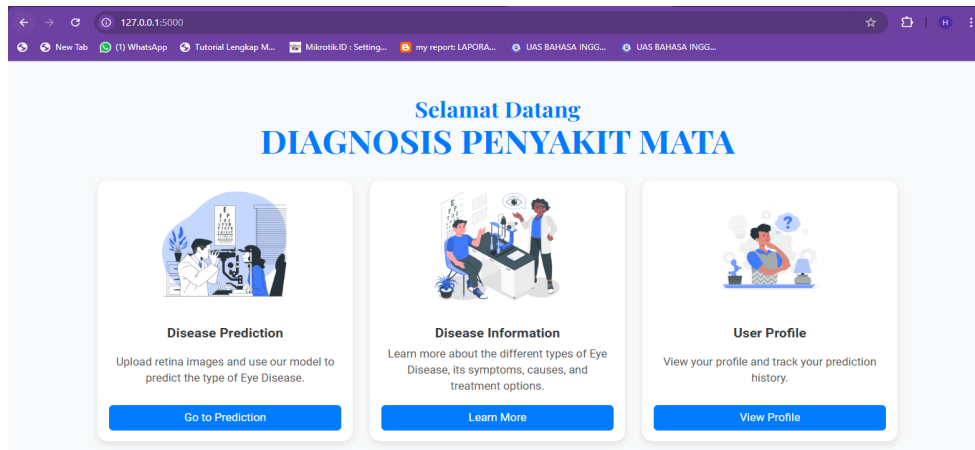
Tabel 2 dibawah menunjukkan Hasil Evaluasi setiap model, Model CNN Menghasilkan Test Akurasi sebesar 85% dengan Test Lost sebesar 0,3841 mengungguli jauh lebih baik dibandingkan model RNN dan FNN. Dengan Test Akurasi Sebesar 83% dan Test Loss sebesar 0,4287 untuk RNN. Kemudian FNN dengan Test akurasi sebesar 69% dan Test Loss sebesar 0,6845

Tabel 2. Evaluasi Perbandingan Model

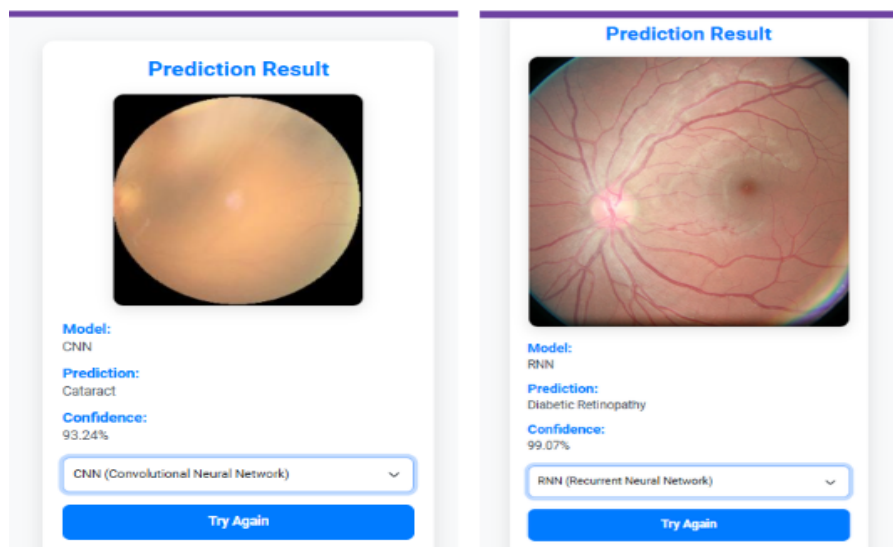
Nama Model	Test Akurasi	Test Lost
CNN	0.8457	0,3841
RNN	0,8283	0,4287
FNN	0,7843	0,5578

3. 5. Implementasi

Implementasi akhir dari program ini dilakukan dalam bentuk aplikasi berbasis Flask. Model CNN, RNN, dan FNN yang telah dilatih menggunakan dataset citra penyakit mata disimpan dalam format file .h5 untuk kemudian diintegrasikan ke dalam aplikasi Flask. Hasil pengujian terhadap citra uji ditampilkan melalui beberapa contoh visualisasi yang diperlihatkan pada Gambar 14, Gambar 15, dan Gambar 16.



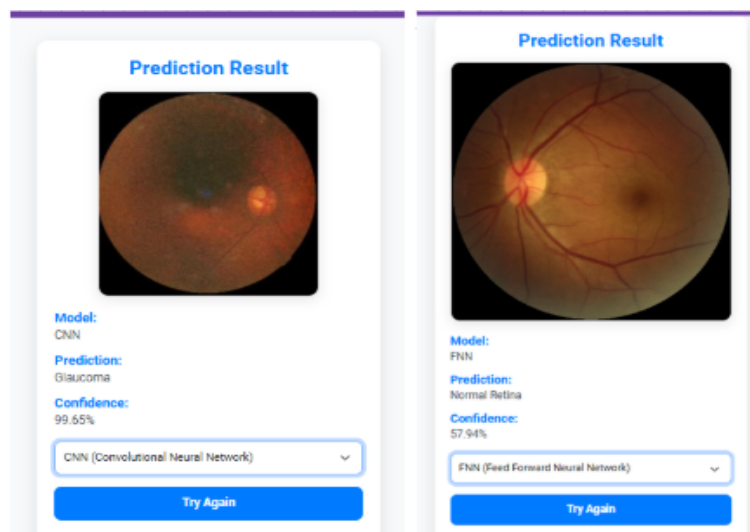
Gambar 14. Tampilan Aplikasi Flask



(a)

(b)

Gambar 15. (a) Hasil Predict Katarak (b) Hasil Predict Diabetic Retinopathy



(a)

(b)

Gambar 16. (a) Hasil Predict Gloucoma (b) Hasil Predict Normal

4. KESIMPULAN

Berdasarkan hasil penelitian tentang Analisis Performa Model Deep Learning dalam Deteksi Penyakit Mata: Studi CNN, RNN, dan FNN menunjukkan bahwa ketiga model ini mampu melakukan klasifikasi. Namun dalam penelitian ini, hasil evaluasi Model CNN memberikan hasil terbaik untuk sebagian besar kategori penyakit, disusul oleh RNN, sedangkan FNN memiliki performa terendah. Dengan hasil test akurasi sebesar 85% dengan test loss sebesar 0,3841 untuk model CNN, Kemudian RNN 83% dengan test loss 0,4287, dan model FNN hanya mencapai akurasi 78% dengan test loss 0,5578. Hasil ini menegaskan keunggulan metode CNN dalam menganalisis citra penyakit mata dibandingkan model lainnya. Performa yang belum terlalu optimal kemungkinan disebabkan oleh arsitektur model yang tidak cukup kompleks. Kedepannya untuk meningkatkan akurasi, disarankan menggunakan transfer learning dengan model pretrained seperti VGG16, ResNet, atau EfficientNet, yang memiliki arsitektur kompleks dan telah dilatih pada dataset besar. Pendekatan ini diharapkan mampu memperbaiki kinerja sistem klasifikasi citra penyakit mata agar akurasi prediksi model dapat jauh meningkat.

REFERENSI

- [1] M. S. Qulub and S. Agustin, "Indentifikasi Penyakit Mata Dengan Klasifikasi Citra Foto Fundus Menggunakan Convolutional Neural Network (Cnn)," 2024.
- [2] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiani, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *SISTEMASI*, vol. 10, no. 3, pp. 618–626, 2021.
- [3] V. H. Phung and E. J. Rhee, "A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences (Switzerland)*, vol. 9, no. 21, Nov. 2019, doi: 10.3390/app9214500.
- [4] S. Dutta, B. C. S. Manideep, S. M. Basha, R. D. Caytiles, and N. C. S. N. Iyengar, "Classification of diabetic retinopathy images by using deep learning models," *International Journal of Grid and Distributed Computing*, vol. 11, no. 1, pp. 89–106, 2018, doi: 10.14257/ijgdc.2018.11.1.09.

- [5] H. Z. Alemu, W. Wu, and J. Zhao, "Feedforward neural networks with a hidden layer regularization method," *Symmetry (Basel)*, vol. 10, no. 10, 2018, doi: 10.3390/sym10100525.
- [6] William and C. Lubis, "Klasifikasi Penyakit Mata Menggunakan CNN," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 10, no. 1, pp. 1–4, Mar. 2022.
- [7] O. Bernabe, E. Acevedo, A. Acevedo, R. Carreno, and S. Gomez, "Classification of Eye Diseases in Fundus Images," *IEEE Access*, vol. 9, pp. 101267–101276, 2021, doi: 10.1109/ACCESS.2021.3094649.
- [8] M. S. Junayed, M. B. Islam, A. Sadeghzadeh, and S. Rahman, "CataractNet: An automated cataract detection system using deep learning for fundus images," *IEEE Access*, vol. 9, pp. 128799–128808, 2021, doi: 10.1109/ACCESS.2021.3112938.
- [9] S. Dutta, B. C. S. Manideep, S. M. Basha, R. D. Caytiles, and N. C. S. N. Iyengar, "Classification of diabetic retinopathy images by using deep learning models," *International Journal of Grid and Distributed Computing*, vol. 11, no. 1, pp. 89–106, 2018, doi: 10.14257/ijgdc.2018.11.1.09.
- [10] Y. J. Nurrisi and Alamsyah, "Optimasi Deep Convolutional Neural Network (Deep CNN) untuk Deteksi Aritmia Melalui Sinyal EKG Menggunakan Arsitektur Conv1D," *Indonesian Journal of Mathematics and Natural Sciences*, vol. 46, no. 1, pp. 10–20, Apr. 2023.
- [11] M. Zahir and R. A. Saputra, "DEteksi Penyakit Retinopati Diabetes Menggunakan Citra Mata Dengan Implementasi Deep Learning CNN," *Jurnal Teknoinfo*, vol. 18, no. 1, pp. 121–132, Jan. 2024.
- [12] K. Azmi, S. Defit, and Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *Jurnal Unitek*, vol. 16, no. 1, 2023.
- [13] V. H. Phung and E. J. Rhee, "A deep learning approach for classification of cloud image patches on small datasets," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 3, pp. 173–178, 2018, doi: 10.6109/jicce.2018.16.3.173.
- [14] D. Esterlina Br Jabat, L. Yanti Sipayung, and K. Raih Syahputra Dakhi, "Penerapan Algoritma Recurrent Neural Networks (RNN) Untuk Klasifikasi Ulos Batak Toba," *SNISTIK : Seminar Nasional Inovasi Sains Teknologi Informasi Komputer*, vol. 1, no. 2, 2024.
- [15] S. Gheisari *et al.*, "A combined convolutional and recurrent neural network for enhanced glaucoma detection," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-81554-4.
- [16] D. Aprillia, T. Rohana, T. Al Mudzakir, and D. Wahiddin, "Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 4, 2024, doi: 10.30865/klik.v4i4.1711.
- [17] F. Maulana and E. B. Setiawan, "Performance of Deep Feed-Forward Neural Network Algorithm Based on Content-Based Filtering Approach," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 8, no. 2, pp. 278–294, Aug. 2024, doi: 10.29407/intensif.v8i2.22904.