

# DETEKSI PENYAKIT DIABETES, KATARAK DAN GLAUKOMA PADA CITRA FUNDUS RETINA MATA MANUSIA MENGGUNAKAN CNN DENGAN ARSITEKTUR ALEXNET

Cicik Rafka Mulyasari<sup>1</sup>, Asep Id Hadiana<sup>2</sup>, Agus Komarudin<sup>3</sup>

<sup>1</sup> Universitas Jenderal Achmad Yani

<sup>2</sup> Universitas Jenderal Achmad Yani

<sup>3</sup> Universitas Jenderal Achmad Yani

Email : cicikrafkams18@if.unjani.ac.id

## ABSTRAK

*Penglihatan merupakan salah satu indera yang sangat penting bagi manusia. Jika terjadi kelainan atau penyakit pada mata, dapat berdampak serius, seperti diabetes yang dapat menyerang bola mata, katarak, dan glaukoma. Karena itu, menjaga kesehatan mata harus menjadi prioritas sehari-hari. Sayangnya, fasilitas medis terbatas di pusat kesehatan dan rumah sakit, serta kurangnya jumlah dokter mata yang mampu melakukan pemeriksaan dan operasi mata, menjadikan pengobatan penyakit mata menjadi sulit. Namun, seiring dengan pesatnya perkembangan teknologi, bidang kedokteran juga mengadopsi teknologi untuk meningkatkan pelayanan kepada masyarakat secara luas. Keterbatasan waktu yang dialami oleh dokter-dokter tersebut telah mendorong penggunaan sistem pakar dalam membantu diagnosis berbagai penyakit. Untuk mengatasi masalah dalam diagnosis penyakit diabetes, katarak, dan glaukoma, telah dikembangkan beberapa solusi. Salah satunya adalah dengan menggunakan metode pengolahan citra bola mata dan bahasa pemrograman untuk menciptakan perangkat lunak yang dapat melakukan diagnosis dan klasifikasi yang akurat terhadap keempat jenis penyakit tersebut. Dalam hal ini, metode yang digunakan adalah convolutional neural network (CNN). Sistem dirancang menggunakan metodologi CNN dengan arsitektur Alexnet berdasarkan gambar bola mata. Masukkan kumpulan data gambar yang diwarisi dari Kaggle yang disebut "Klasifikasi Penyakit Mata" dan gunakan total 4.217 gambar, 75% di antaranya adalah data latih dan 25% adalah gambar data uji. Pada penelitian ini, hasil terbaik untuk setiap skenario pengujian adalah presisi 87%, presisi 88%, recall 88%, skor f1 88% dan loss 0,48.*

**Kata kunci:** Convolutional Neural Network (CNN), Alexnet, Diabetes, Katarak, Glaukoma

## ABSTRACT

*Vision is one of the crucial senses for humans. Any abnormalities or diseases in the eyes can have serious implications, such as diabetes affecting the eyes, cataracts, and glaucoma. Therefore, maintaining eye health should be a daily priority. Unfortunately, medical facilities in health centers and hospitals are limited, and the insufficient number of eye doctors capable of conducting eye examinations and surgeries makes treating eye diseases difficult. However, with the rapid advancement of technology, the field of medicine has also embraced technology to enhance services for the public extensively. The time constraints faced by these doctors have led to the use of expert systems to assist in diagnosing various diseases. To address issues in diagnosing diabetes, cataracts, and glaucoma, several solutions have been developed. One of them involves using image processing of the eyeball and programming languages to create software capable of accurately diagnosing and classifying these four types of diseases. In this case, the method used is the convolutional neural network (CNN). The system is designed using the CNN methodology with an AlexNet architecture based on eyeball images. A dataset inherited from Kaggle, called "Eye Disease Classification," is used, comprising a total of 4,217 images, with 75% used as training data and 25% as test data. In this research, the best results for each testing scenario were precision at 87%, precision at 88%, recall at 88%, an F1 score of 88%, and a loss of 0.48.*

**Keywords:** Convolutional Neural Network (CNN), AlexNet, Diabetes, Cataracts, Glaucoma.

## 1. PENDAHULUAN

Mata adalah salah satu dari lima indera manusia yang sangat penting. Informasi yang diterima manusia sebagian besar berasal dari penglihatan (Kumasela, 2013). Gangguan mata seperti katarak, retinopati diabetik, dan glaukoma disebabkan oleh faktor-faktor seperti gizi yang tidak mencukupi, kurangnya olahraga, dan penggunaan berlebihan perangkat yang memancarkan cahaya ultraviolet. Intervensi dini sangat penting untuk mencegah risiko kebutaan (Indraswari, 2022). Menurut survei Vision Atlas 2020, ada 1,1 miliar orang di seluruh dunia yang mengalami gangguan penglihatan. Mendeteksi dan mengatasi penyakit mata secara dini dapat mencegah masalah ini (Burton, 2021).

Gambar fundus, juga dikenal sebagai foto fundus, sering digunakan untuk mendeteksi berbagai gangguan mata seperti retinopati diabetik, glaukoma, katarak, dan miopia. Gambar ini memberikan representasi dua dimensi dari struktur retina tiga dimensi, diambil dengan mikroskop perbesaran rendah yang khusus. Gambar fundus efektif dan terjangkau untuk mendeteksi komplikasi (Gour, 2022). Namun, metode tradisional melibatkan analisis manual oleh dokter berpengalaman, yang dapat menghasilkan variasi hasil (kesalahan antar-pengamat). Untuk mengatasi keterbatasan ini, Sistem Diagnosis Berbantu Komputer (CAD) telah dikembangkan untuk membantu dokter menganalisis gangguan mata secara lebih konsisten dan akurat.

Metode pembelajaran dalam yang dalam, terutama Jaringan Saraf Konvolusi (CNN), telah maju pesat, mengurangi ketergantungan pada segmentasi dan ekstraksi fitur manual. CNN secara otomatis mempelajari hubungan antara fitur-fitur dalam gambar untuk klasifikasi instan, menghilangkan kebutuhan ekstraksi fitur manual (Gulshan, 2016). Teknik deep learning menggunakan CNN telah menunjukkan akurasi tinggi dalam klasifikasi gambar dengan dataset pelatihan besar. Sebagai contoh, algoritma deep learning telah mendeteksi retinopati diabetik dengan akurasi tinggi (Gulshan, 2016). Studi lain mendeteksi glaukoma dengan AUC 96% menggunakan berbagai arsitektur CNN (Diaz-Pinto, 2019). Metode perhatian berbasis CNN mencapai akurasi 95,3% dalam mendeteksi glaukoma (Li, 2019). Teknik seperti RCNN telah berhasil mendeteksi dan mengklasifikasikan glaukoma secara andal (Bajwa, 2019). Arsitektur lain seperti InceptionResNetV2 dan YOLO v3 juga menunjukkan hasil yang baik (Joshi, 2020).

Meskipun berbagai metode deep learning telah menunjukkan hasil yang baik dalam mendeteksi gangguan mata, tantangan tetap ada dalam hal peningkatan akurasi dan efisiensi deteksi untuk aplikasi klinis sehari-hari. Sistem yang ada masih menghadapi kendala dalam hal akurasi deteksi yang konsisten, efisiensi waktu, dan penerapan dalam skala besar.

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi penyakit mata berbasis CNN yang efektif dan efisien menggunakan gambar fundus. Dengan menggunakan gambar fundus dari Kaggle dan mengadopsi arsitektur AlexNet, penelitian ini bertujuan untuk meningkatkan akurasi dan efisiensi deteksi penyakit mata seperti retinopati diabetik, katarak, dan glaucoma, menyediakan solusi berbasis deep learning yang dapat diandalkan untuk membantu dokter dalam diagnosis dini penyakit mata, mengeksplorasi potensi penggunaan CNN, khususnya AlexNet, dalam aplikasi medis untuk pengenalan dan klasifikasi penyakit mata.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam bidang deteksi dini penyakit mata, meningkatkan kualitas diagnosis, dan mengurangi risiko kebutaan melalui intervensi dini yang lebih akurat.

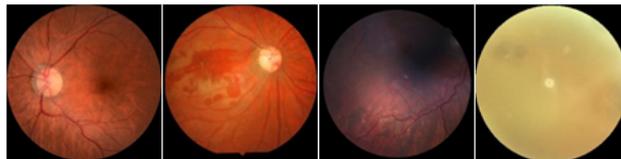
## 2. METODE & TEORI PENDUKUNG

### 2.1. Pengumpulan Data

Data penelitian merupakan data publik yang diambil dari Kaggle, Kaggle adalah platform populer bagi praktisi *data science* dan *machine learning* di Indonesia. Kaggle memungkinkan praktisi *data science* untuk mendapatkan data berkualitas guna penelitian mereka, termasuk data untuk pendekatan *Deep Learning* seperti *Convolutional Neural Network*, Putra (2016).

Kumpulan data citra fundus adalah data dari Kaggle yang terdiri dari 4.217 citra fundus. Kumpulan data memiliki empat kelas yaitu Normal, Katarak, Glaukoma, dan Diabetes, dimana pada kelas Normal memiliki 1074 total citra, kelas Katarak 1038 total citra, kelas Glaukoma 1007 total citra, dan pada kelas diabetes retina 1098 total citra, sehingga total data dataset yang digunakan pada penelitian ini berjumlah 4.217 citra fundus. Dataset pada penelitian ini dibagi menjadi empat kelas, sehingga penelitian ini difokuskan pada data yang tidak seimbang pada dataset (Tidak ada isu terkait dengan dataset yang tidak seimbang) yang berpengaruh terhadap performa akurasi algoritma pembelajaran mendalam, Indraswari (2019), Indraswari (2022).

Pada penelitian ini, total dataset dibagi menjadi 75% data pelatihan dan 25% untuk data pengujian. Untuk sample dataset yang akan digunakan dari masing-masing kelas dapat dilihat pada Gambar 1.



**Gambar 1. Sample Dataset (A)Normal, (B)Glaukoma, (C)Diabetes Retina, (D)Katarak**

### 2.2. Praproses Data

Praproses adalah tahap awal sebelum citra dimasukkan ke dalam sistem. Tujuannya adalah meningkatkan kualitas citra dan menjadikannya seragam agar dapat meningkatkan performa sistem. Tanpa pre-processing, model akan mengalami kesulitan dalam mengolah data dan performa sistem tidak akan optimal. Salah satu langkah dalam pre-processing adalah melakukan resizing citra.

Resizing citra bertujuan untuk menyesuaikan ukuran gambar agar seragam dalam satuan piksel. Hal ini penting karena ukuran citra yang tidak seragam dapat menghambat proses pemrosesan di dalam sistem. Lain dari pada itu, ukuran gambar yang memiliki ukuran besar dapat memperlambat waktu pengolahan, sementara ukuran gambar yang lebih kecil dapat mengurangi kualitas gambar. Oleh karena itu, dalam pre-processing, resizing gambar dilakukan pada semua rancangan pengujian untuk memastikan ukuran gambar seragam dan sesuai dengan kebutuhan sistem.

### 2.3. Parameter Pengujian

Pada penelitian kali ini penulis mempunyai lima parameter yang digunakan untuk menguji model sistem yaitu:

- a. Pengaruh Resize citra pada saat Pre-processing
- b. Pengaruh penggunaan Adam, Nadam dan RMSprop
- c. Pengaruh Nilai Learning Rate pada setiap layer terhadap akurasi, dimana nilai Learning Rate yang digunakan yaitu: 0.01, 0.001, 0.0001
- d. Pengaruh Nilai Epoch terhadap Akurasi, dimana nilai Epoch yang digunakan yaitu: 5, 10, 25, 35
- e. Pengaruh nilai Batch Size terhadap Akurasi, dimana nilai Batch Size yang digunakan yaitu: 8, 16, 32

Pengujian kelima parameter tersebut diambil dari penelitian sebelumnya, sistem performansi parameter tersebut akan dilakukan kedalam model alexnet pada penelitian ini.

#### **2.4. Citra Digital**

Citra adalah fungsi dua dimensi yang menggambarkan intensitas cahaya dalam sebuah gambar. Variabel  $x$  dan  $y$  merepresentasikan koordinat dalam citra, sedangkan  $f$  merepresentasikan tingkat keabuan dari setiap titik  $(x, y)$ , Kusumanto (2011).

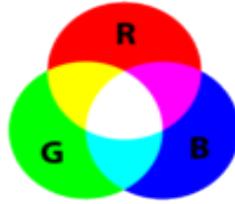
Citra digital adalah citra berkelanjutan yang diubah menjadi bentuk diskrit, baik dalam koordinat citra maupun tingkat cahaya. Proses ini melibatkan pemindaian berurutan dari gambar asli dengan jarak tetap. Citra digital dapat direpresentasikan sebagai matriks dengan ukuran  $M \times N$ , di mana  $M$  mengindikasikan tinggi dan  $N$  mengindikasikan lebar citra, Kusumanto (2011). Intensitas cahaya dalam citra digital diwakili oleh nilai-nilai piksel yang terdapat dalam matriks tersebut.

#### **2.5. Citra *Greyscale***

*Greyscale* atau skala keabuan adalah representasi data matriks di mana nilai-nilainya mewakili tingkat keabuan dalam citra. Ketika citra digital mempunyai skala keabuan 8bit, rentang nilai intensitasnya adalah 0 hingga 255. Sedangkan jika citra memiliki skala keabuan 16-bit, rentang nilai intensitasnya adalah 0 hingga 65535. Pada skala keabuan ini, nilai minimal mewakili warna hitam, sedangkan nilai maksimal mewakili warna putih, Suyanto (2023).

#### **2.6. Citra RGB**

Rancangan warna RGB dilandasi pada cara retina mata pada manusia memahami cahaya dan mengubahnya menjadi gerakan pada otak. Model ini menggunakan kombinasi dari tiga unsur warna utama, yaitu Merah (*Red / R*), Hijau (*Green / G*), dan Biru (*Blue / B*), untuk membentuk berbagai macam warna. Setiap komponen warna dasar memiliki rentang nilai tertentu. Pada layar komputer, nilai minimum adalah 0 dan nilai maksimum adalah 255. Skala 256 ini digunakan karena representasi angka biner 8 bit yang digunakan dalam komputer. Dengan menggunakan metode ini, kita dapat memiliki total 16.777.216 jenis warna yang dapat dihasilkan. Warna-warna ini dapat dianggap sebagai vektor dalam ruang tiga dimensi, dengan koordinat yang direpresentasikan oleh tiga angka, yaitu komponen  $x$ ,  $y$ , dan  $z$ . Komponen R (Merah), G (Hijau), dan B (Biru) ini dapat dituliskan sebagai vektor  $r = (x, y, z)$ . Sebagai contoh, sebuah warna dapat dituliskan sebagai warna = RGB (70, 70, 255). Warna putih direpresentasikan sebagai RGB (255, 255, 255), sementara warna hitam direpresentasikan sebagai RGB (0, 0, 0). Visualisasi representasi warna RGB dapat dilihat pada Gambar 2.



**Gambar 2. Representasi warna RGB**

## **2.7. Deep Learning**

*Deep Learning* adalah salah satu cabang dari *Machine Learning* yang menggunakan algoritma *neural network* yang kompleks dengan banyak lapisan (layer) untuk mempelajari pola-pola yang tersembunyi dalam data, sehingga dapat melakukan tugas-tugas seperti klasifikasi, regresi, atau pengenalan pola secara lebih akurat dan efisien, Suyanto (2023).

## **2.8. Convolutional Neural Network (CNN)**

CNN adalah salah satu jenis jaringan saraf yang sering digunakan untuk memproses data gambar. CNN memiliki kemampuan untuk mendeteksi dan mengenali objek dalam gambar. Inspirasi utama dalam pengembangan CNN berasal dari cara kerja otak manusia dalam melakukan persepsi visual, terutama dalam mendeteksi dan mengenali objek, Bejiga (2017).

Arsitektur CNN sangat sederhana yaitu terdiri atas satu lapis masukan (*input layer*), suatu lapisan keluar (*output layer*), dan sejumlah lapisan tersembunyi (*hidden layer*), Mo (2019). Lapisan tersembunyi biasanya mencakup kedalam tiga layer utama yaitu *convolutional layers*, *pooling layers* dan *fully connected layers*, Putra (2016).

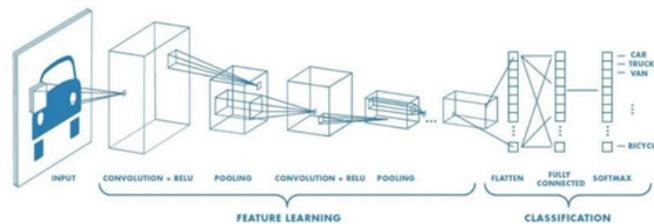
*Convolutional Layer* adalah salah satu lapisan yang terdapat dalam Bagian Ekstraksi Fitur pada *Convolutional Neural Network* (CNN). Pada lapisan ini, terdapat *neuron-neuron* yang berdekatan membentuk filter dua dimensi dengan ukuran panjang dan lebar (piksel). Misalnya, jika citra masukan memiliki ukuran 5x5x3, berarti memiliki panjang 5 piksel, lebar 5 piksel, dan kedalaman 3 piksel sesuai dengan saluran citra tersebut. Kemudian, dilakukan operasi dot product antara masukan dan nilai filter untuk menghasilkan peta fitur keluaran. Stride berfungsi untuk menentukan pergeseran filter dengan menentukan jumlah piksel yang digeser secara horizontal dan vertikal. Semakin kecil nilai stride, maka hasil yang diperoleh akan semakin detail, tetapi waktu komputasi yang dibutuhkan juga akan lebih lama.

Lapisan Pooling terdiri dari filter dan stride dengan ukuran tertentu. Tujuan penggunaan lapisan ini adalah untuk mengurangi dimensi peta fitur (*downsampling*), sehingga prosesnya menjadi efisien dengan mengurangi jumlah parameter yang perlu diperbarui, dan juga untuk membantu mengatasi masalah *overfitting*. Lapisan Pooling juga digunakan untuk mengurangi volume data dengan menggunakan metode *Max Pooling* atau *Average Pooling*.

Pada *Max Pooling*, nilai maksimum diambil dari area filter saat dilakukan pergeseran, sedangkan pada *Average Pooling*, nilai rata-rata diambil dari area filter tersebut. Dengan demikian, informasi yang relevan tetap dipertahankan dalam representasi fitur, namun dengan mengurangi jumlah data yang perlu diproses. Hal ini membantu mengurangi kompleksitas perhitungan dan penggunaan memori, serta memberikan invariansi terhadap translasi dalam variasi pergeseran spasial pada data gambar.

*Fully connected layer* adalah lapisan yang semua neuronnya saling terhubung penuh satu sama lain, karena itu lapisan ini disebut dengan *fully connected layer*. Penggunaan *fully connected layer* hanya dapat digunakan pada akhir jaringan

Dalam perbandingan dengan pendahulunya, CNN memiliki keuntungan utama karena dapat mengotomatisasi deteksi fitur penting melalui pembelajaran, interpretasi, dan ekstraksi data dalam bentuk gambar. Gambar 3, Mo (2019). menunjukkan arsitektur CNN yang digunakan dalam hal ini.

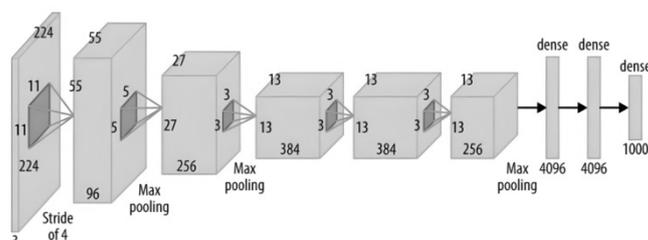


**Gambar 3. CNN**

Pada awa-awal kemunculannya, arsitektur CNN didesain dengan hanya berisi sedikit lapisan (kurang dari sepuluh) [21]. Perkembangan CNN sangat pesat hanya dalam beberapa tahun. Para ahli mendesain CNN dengan ratusan bahkan ribuan lapisan. Beberapa arsitektur CNN yang populer yaitu: LeNet-5, AlexNet, ZFNet, GoogleNet, CapsNet, dan ResNet.

## 2.9. AlexNet

Arsitektur AlexNet terdiri dari 5 lapis *convolutional*, 3 lapis *pooling*, 2 lapis *dropout*, dan 3 lapis *fully connected*, Nugroho (2019). Konvolusi adalah istilah matematika yang berarti berulang kali menerapkan suatu fungsi ke output fungsi lain untuk membuat peta fitur. Fungsi konvolusi digunakan untuk mengekstrak ciri gambar. *Pooling* atau subsampling mengurangi ukuran matriks. Yang biasa digunakan itu adalah max pooling dan average pooling, Bejiga (2017). *Dropout* yaitu salah satu teknik regularisasi yang digunakan dalam jaringan saraf tiruan (*neural network*) untuk mencegah *overfitting* pada data *training* dengan probabilitas 50% untuk didropout (diabaikan) pada setiap iterasi *training*. Dengan demikian, *dropout* pada AlexNet dapat membantu mengurangi ketergantungan antar neuron dan memperkuat generalisasi model *Fully Connected layer* ini bertugas untuk menghubungkan *output* dari *layer* konvolusi dan pooling sebelumnya sehingga dapat digunakan untuk melakukan klasifikasi pada data input, Mo (2019). Arsitektur Alexnet ditunjukkan pada Gambar 4, Sinthanayothin (2022), dan rincian lapisan Alexnet ditunjukkan pada Tabel 1.



**Gambar 4. Arsitektur Alexnet**

**Tabel 1. Detail Dari Arsitektur Alexnet**

Layer	Feature Map	Size	Karnel Size	Stride	Activation
Input	1	$32 \times 32 \times 3$	-	-	-
Convolution 1	96	$16 \times 16 \times 96$	$11 \times 11$	4	relu
Max Pooling 1	96	$8 \times 8 \times 96$	$3 \times 3$	2	relu
Convolution 2	256	$8 \times 8 \times 256$	$5 \times 5$	1	relu
Max Pooling 2	256	$4 \times 4 \times 256$	$3 \times 3$	2	relu
Convolution 3	384	$4 \times 4 \times 384$	$3 \times 3$	1	relu
Convolution 4	384	$4 \times 4 \times 384$	$3 \times 3$	1	relu
Convolution 5	256	$4 \times 4 \times 256$	$3 \times 3$	1	relu
MaxPooling 3	256	$2 \times 2 \times 256$	$3 \times 3$	2	relu
Flatten	-	1024	-	-	relu
Fully connected 1	-	4096	-	-	relu
Fully connected 2	-	4096	-	-	relu
Fully connected 3	-	1000	-	-	Softmax

## 2.10. Confusion Matrix

Pengukuran kinerja sistem yang telah dibuat merupakan langkah kritis untuk mencapai hasil yang optimal. Salah satu metode umum yang digunakan dalam mengukur kinerja sistem, terutama dalam *deep learning*, adalah *confusion matrix*.

*Confusion matrix* biasanya memberikan pengetahuan tentang perbandingan hasil klasifikasi dengan nilai yang sebenarnya dalam bentuk tabel matriks. Tabel matriks ini memaparkan performansi sistem klasifikasi pada sekumpulan data uji yang diketahui nilai sebenarnya. Ditunjukkan pada Tabel 2.2 *Confusion Matrix Multiclass*, ada empat kombinasi nilai prediksi dan aktual yang berbeda yang tercermin dalam matriks kebingungan, Putra (2022).

Pada kasus klasifikasi dengan lebih dari dua kelas, digunakan confusion matrix multikelas. Oleh karena itu, jika ada N kelas yang digunakan untuk klasifikasi, maka confusion matrix akan memiliki dimensi  $N \times N$ , seperti yang ditunjukkan dalam contoh Tabel 2 berikut ini.

**Tabel 2. Confusion matrix multiclass**

		<i>Detect Class</i>			
		A	B	C	D
<i>True Class</i>	A	$X_{AA}$	$X_{AB}$	$X_{AC}$	$X_{AD}$
	B	$X_{BA}$	$X_{BB}$	$X_{BC}$	$X_{BD}$
	C	$X_{CA}$	$X_{CB}$	$X_{CC}$	$X_{CD}$
	D	$X_{DA}$	$X_{DB}$	$X_{DC}$	$X_{DD}$

Dalam Tabel 2, kita dapat menentukan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Sebagai contoh, untuk menghitung nilai TP, TN, FP, FN untuk Kelas A, kita dapat melakukan perhitungan sebagai berikut.

$$TP = X_{AA}$$

$$TN = (X_{BB} + X_{BC} + X_{BD} + X_{CB} + X_{CC} + X_{CD} + X_{DB} + X_{DC} + X_{DD})$$

$$FP = (X_{BA} + X_{CA} + X_{DA})$$

$$FN = (X_{AB} + X_{AC} + X_{AD})$$

Metode yang sama berlaku untuk kelas lainnya, sehingga dapat memperoleh semua nilai untuk setiap kelas. Tujuannya adalah untuk mengevaluasi kinerja dari sistem yang telah dirancang. Untuk melakukan evaluasi ini, digunakan perhitungan metrik kinerja berdasarkan matriks kebingungan multikelas, yang dapat dijelaskan sebagai berikut.

- Akurasi

Akurasi mengukur seberapa akurat sistem klasifikasi yang telah dirancang dalam mengklasifikasikan data. Akurasi dapat dihitung sebagai rasio antara jumlah prediksi yang benar (*true positive* dan *true negative*) dengan keseluruhan data yang diamati. Persamaannya dapat dilihat pada persamaan 1, Putra (2022).

$$\text{Akurasi} = \frac{TP(A) + TP(B) + TP(C) + TP(D)}{TP + TN + FP + FN} \quad (1)$$

- Presisi

Presisi (*precision*) mengukur tingkat keakuratan sistem dalam memprediksi data positif. Presisi dapat dihitung sebagai rasio antara prediksi benar positif (*true positive*) dan jumlah keseluruhan prediksi positif (*true positive* dan *false positive*). Persamaannya dapat dilihat pada persamaan 2, Putra (2022).

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2)$$

- Recall

Recall mengukur kemampuan sistem dalam menemukan kembali informasi yang benar. Recall dapat dihitung sebagai rasio antara prediksi benar positif (*true positive*) dan jumlah keseluruhan kasus positif (*true positive* dan *false negative*). Persamaannya dapat dilihat pada persamaan 3, Putra (2022).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

- F1-Score

F1-score digunakan untuk mencapai keseimbangan antara presisi (*precision*) dan recall. F1-score membantu dalam menghitung presisi dan recall secara bersamaan, dan memberikan metrik yang menggabungkan kedua nilai tersebut. Persamaan yang digunakan untuk menghitung F1-score dapat dilihat pada persamaan 4.

$$\text{F1-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (4)$$

- Loss

Untuk mengetahui seberapa besar kesalahan identifikasi pada sistem maka digunakan Loss. Penelitian ini menggunakan Categorical Cross Entropy sebagai Loss Function yang biasa digunakan pada model klasifikasi dan digunakan untuk klasifikasi multiclass, Putra (2022).

$$\text{CCE} = \sum_i^c t_i \log (f(s)_i) \quad (5)$$

Dimana CCE adalah Categorical Cross Entropy,  $t_i$  adalah True Label, dan  $f(s)$ , adalah Softmax.

Macro Average (macro avg): Macro average menghitung rata-rata metrik evaluasi untuk setiap kelas secara terpisah, dan kemudian mengambil rata-rata dari metrik tersebut. Ini memberikan bobot yang sama untuk setiap kelas, tanpa memperhatikan seberapa banyak sampel yang termasuk dalam setiap kelas. Dalam hal ini, setiap kelas dianggap sama pentingnya, dan kesalahan pada kelas minoritas memiliki dampak yang sama dengan kesalahan pada kelas mayoritas.

Weighted Average (weighted avg): Weighted average juga menghitung rata-rata metrik evaluasi untuk setiap kelas, tetapi bobotnya berdasarkan jumlah sampel dalam setiap kelas. Metrik evaluasi dihitung terlebih dahulu untuk setiap kelas, dan kemudian rata-rata diambil dengan memperhitungkan jumlah sampel dalam setiap kelas. Dalam hal ini, setiap kelas memiliki bobot yang berbeda tergantung pada proporsi sampelnya. Metode ini memberikan penekanan lebih pada kinerja kelas mayoritas karena bobotnya lebih tinggi, Putra (2022).

### 3. HASIL DAN DISKUSI

Penelitian dilakukan dengan melakukan pengujian terhadap beberapa skenario sehingga menghasilkan beberapa hasil pengujian berupa :

#### a. Hasil Pengujian *Resize* Citra

Dalam skenario pengujian ini, kami melakukan penyesuaian ukuran citra dataset agar memiliki ukuran yang sama untuk keperluan klasifikasi. Pada tahap ini, untuk merubah gambar dataset menjadi seragam, dilakukan proses *resize*. Parameter sistem yang digunakan dalam pengujian ini telah ditentukan sebagai berikut: menggunakan optimizer Adam, learning rate 0,0001, epoch 35, dan batch size 32.

Pada skenario pertama, kami melakukan perubahan ukuran gambar menjadi 32x32 piksel, 64x64 piksel, dan 128x128 piksel. Hasil yang didapatkan dari pengujian pertama ini dapat dilihat dalam Tabel 3 berikut ini:

**Tabel 3. Perbandingan hasil pengujian *resize***

Parameter	Hasil Pengujian <i>Resize</i>		
	32×32 px	64×64 px	128×128 px
Akurasi	0,82	0,85	0,87
Presisi	0,82	0,84	0,87
<i>Recall</i>	0,82	0,84	0,87
<i>F1-Score</i>	0,82	0,84	0,87
<i>Loss</i>	0,82	0,62	0,61

Dari Tabel 3, dapat diketahui bahwa skenario pengujian *resize* dengan menggunakan ukuran 128×128 piksel menghasilkan akurasi terbaik sebesar 87%, serta presisi, recall, dan F1-score yang sama, yaitu 87%. Nilai loss yang diperoleh adalah 0,61. Hal ini disebabkan oleh fakta bahwa penggunaan ukuran *resize* 32×32 piksel dan 64×64

piksel menyebabkan hilangnya beberapa informasi penting dalam dataset yang digunakan, baik karena ukuran yang terlalu kecil maupun terlalu besar. Oleh karena itu, nilai akurasi yang diperoleh dari kedua ukuran tersebut kurang optimal. Dengan demikian, dapat disimpulkan bahwa skenario pengujian resize dengan ukuran 128×128 piksel lebih cocok digunakan dalam penelitian ini. Tabel 4 menampilkan informasi tentang waktu komputasi yang dibutuhkan dalam skenario pengujian resize.

**Tabel 4. Waktu komputasi pada pengujian resize**

<i>Resize (px)</i>	Waktu (s)
32×32	55
64×64	83
128×128	144

Berdasarkan Tabel 4, diketahui bahwa pemakaian resize citra sebesar 128×128 piksel memproses waktu yang paling lama, yaitu 144 detik. Hal ini disebabkan oleh fakta bahwa semakin besar ukuran resize, semakin banyak piksel yang harus diproses, yang pada gilirannya memerlukan waktu yang lebih lama. Sebaliknya, semakin kecil ukuran resize, maka waktu yang dibutuhkan akan semakin sedikit. Oleh karena itu, ukuran resize citra menjadi faktor penting dalam mengoptimalkan waktu eksekusi sistem, di mana ukuran yang lebih kecil dapat membantu mengurangi waktu yang diperlukan untuk proses pengolahan citra.

b. Hasil Pengujian Nilai *Learning Rate*

Pengujian *Learning Rate* adalah salah satu skenario lain dalam parameter pengujian yang dipakai untuk menilai kinerja sistem. Dalam skenario pengujian *Learning Rate* ini, parameter pengujian terbaik yang diperoleh dari skenario sebelumnya, yaitu resize citra sebesar 128×128 piksel dan *optimizer* Adam, serta nilai epoch 35 dan *batch size* 32, akan digunakan. Pengujian dilakukan untuk mencari nilai *Learning Rate* terbaik dengan menguji nilai 0.01, 0.001, dan 0.0001.

Berikut adalah hasil dari skenario pengujian *Learning Rate* yang ditampilkan dalam Tabel 5.

**Tabel 5. Hasil pengujian learning rate**

Parameter	Hasil Pengujian		
	0,01	0,001	0,0001
Akurasi	0,25	0,84	0,87
Presisi	0,062	0,83	0,87
<i>Recall</i>	0,25	0,83	0,87
<i>F1-Score</i>	0,1025	0,83	0,87
<i>Loss</i>	1,39	0,44	0,61

Berdasarkan Tabel 5, dapat disimpulkan bahwa pengujian dengan learning rate 0,0001 menghasilkan performa terbaik. Pada pengujian tersebut, diperoleh akurasi sebesar 87%, loss sebesar 0,61, serta nilai presisi, recall, dan F1-score yang sama, yaitu 87%. Hal ini menunjukkan bahwa dengan menggunakan *learning rate* yang lebih kecil, sistem dapat bekerja secara optimal dan menghasilkan akurasi yang tinggi. Sebaliknya, jika learning rate memiliki nilai yang lebih besar, sistem dapat mengalami kinerja yang buruk dan mempengaruhi nilai akurasi yang dicapai. Oleh karena itu, penting untuk memilih learning rate yang sesuai agar dapat mencapai hasil yang optimal dalam proses pembelajaran.

#### c. Hasil Pengujian Nilai Epoch

Epoch merupakan jumlah iterasi atau perulangan yang dilakukan saat proses pembelajaran untuk mengolah data. Penggunaan nilai epoch yang tepat dapat mempengaruhi tingkat akurasi model yang dihasilkan. Dalam skenario pengujian ini, akan dilakukan pengujian menggunakan beberapa nilai epoch, yaitu 5, 10, 25, dan 35. Parameter-parameter terbaik yang telah ditemukan sebelumnya, yaitu resize citra sebesar 128x128 piksel, learning rate 0,0001, optimizer Adam, dan batch size 32, akan digunakan dalam pengujian ini. Berikut ini adalah Tabel 6 yang menunjukkan hasil dari skenario pengujian epoch:

**Tabel 6. Hasil pengujian epoch**

Parameter	Hasil Pengujian <i>Epoch</i>			
	5	10	25	35
Akurasi	0,77	0,85	0,86	0,87
Presisi	0,8	0,85	0,86	0,87
<i>Recall</i>	0,77	0,85	0,86	0,87
<i>F1-Score</i>	0,76	0,84	0,86	0,87
<i>Loss</i>	0,54	0,41	0,47	0,61

Berdasarkan data yang terdapat pada Tabel 6, terlihat bahwa dalam pengujian untuk nilai epoch, ditemukan hasil terbaik pada nilai 35. Pada nilai ini, tingkat akurasi, presisi, recall, dan F1-score semua mencapai 87%, sementara nilai loss mencapai 0,61. Dengan demikian, dapat disimpulkan bahwa dalam model AlexNet dan dataset yang digunakan dalam penelitian ini, nilai epoch yang cocok untuk pengujian adalah 35. Meskipun nilai loss sedikit lebih tinggi, mungkin ada kemungkinan untuk memanipulasi parameter pengujian lainnya guna meningkatkan kinerja model.

#### d. Hasil Pengujian Nilai Batch Size

Pengujian keempat dilakukan untuk menentukan nilai batch size yang optimal. Berdasarkan hasil pengujian sebelumnya, telah ditemukan nilai resize terbaik yaitu 128x128 piksel, learning rate 0,0001, dan epoch 35. Oleh karena itu, pada pengujian keempat ini, kita akan menggunakan nilai-nilai terbaik dari pengujian sebelumnya. Selain itu, optimizer yang digunakan adalah Adam.

Dalam pengujian ini, parameter yang akan divariasikan adalah batch size dengan nilai 8, 16, dan 32. Tabel 7 berikut menampilkan hasil dari pengujian batch size tersebut.

**Tabel 7. Hasil pengujian nilai *batch size***

Parameter	Hasil Pengujian <i>Batch Size</i>		
	8	16	32
Akurasi	0,87	0,84	0,87
Presisi	0,87	0,85	0,87
<i>Recall</i>	0,86	0,84	0,87
<i>F1-Score</i>	0,87	0,84	0,87
<i>Loss</i>	0,76	0,6	0,61

Dari data yang terdapat pada Tabel 7, dapat disimpulkan bahwa hasil pengujian menunjukkan bahwa batch size 32 memberikan hasil terbaik. Pada batch size tersebut, diperoleh tingkat akurasi, presisi, recall, dan F1-score sebesar 87% dengan nilai loss sebesar 0,61. Oleh karena itu, dalam pengujian batch size dengan menggunakan dataset dan model yang sama, dapat disimpulkan bahwa penggunaan batch size 32 memberikan hasil yang lebih optimal dibandingkan dengan nilai batch size lainnya.

e. Hasil Pengujian Optimizer

Dalam pengujian optimizer, akan dilakukan variasi parameter menggunakan metode Adam, Nadam, dan RMSprop. Pengujian ini akan menggunakan nilai parameter terbaik yang telah ditemukan pada skenario pengujian sebelumnya, yaitu resize citra 128x128 piksel, learning rate 0,0001, epoch 35, dan batch size 32. Penentuan parameter ini dilakukan karena pengujian ini merupakan pengujian terakhir dan tidak ada nilai parameter tambahan yang ditentukan.

Hasil pengujian optimizer dapat ditemukan pada Tabel 8, yang memberikan informasi mengenai performa masing-masing optimizer berdasarkan parameter-parameter yang telah ditentukan sebelumnya.

**Tabel 8. Hasil pengujian optimizer**

Parameter	Hasil <i>Optimizer</i>		
	Adam	Nadam	RMSprop
Akurasi	0,87	0,87	0,87
Presisi	0,87	0,88	0,87
<i>Recall</i>	0,87	0,88	0,87
<i>F1-Score</i>	0,87	0,88	0,87
<i>Loss</i>	0,61	0,48	0,62

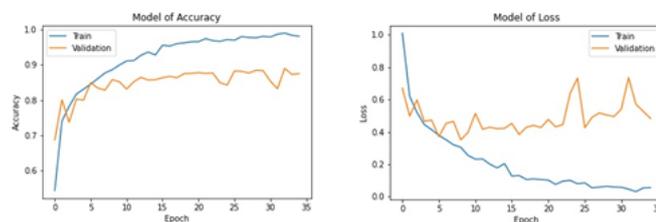
Hasil pengujian optimizer yang tercantum pada Tabel 8 menunjukkan bahwa penggunaan optimizer terbaik pada model dan dataset penelitian ini adalah menggunakan optimizer Nadam. Penggunaan optimizer Nadam menghasilkan akurasi

sebesar 87% serta presisi, recall, dan F1-score sebesar 88%. Selain itu, terdapat loss sebesar 0,48 yang tercapai.

f. Hasil Pengujian Terbaik

Setelah melakukan lima skenario pengujian parameter, telah dilakukan variasi pada resizing citra, learning rate, epoch, batch size, dan optimizer. Setelah pengujian selesai, diperoleh beberapa parameter terbaik dari setiap skenario pengujian. Berikut adalah nilai terbaik yang ditemukan dari masing-masing skenario: resizing citra dengan ukuran 128x128 piksel, learning rate sebesar 0,0001, epoch sebanyak 35 kali, batch size sebesar 32, dan optimizer yang optimal adalah Nadam.

Untuk melihat visualisasi dari model akurasi dan loss dari skenario terbaik, Anda dapat melihatnya pada Gambar 5 berikut.



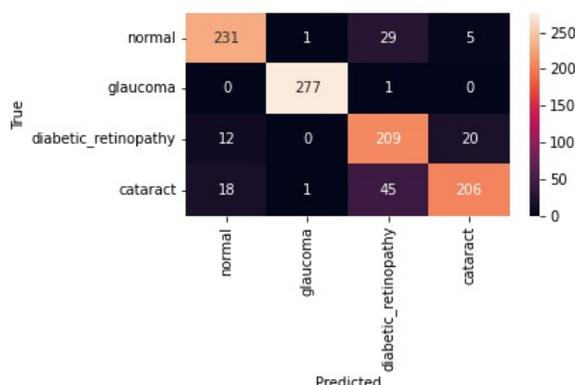
**Gambar 5. Grafik model akurasi dan loss terbaik**

**Tabel 9. Hasil performansi dari parameter pengujian terbaik**

Parameter		Performansi	
<i>Resize</i>	128 × 128 piksel	Akurasi	0,87
<i>Learning Rate</i>	0,0001	Presisi	0,88
<i>Epoch</i>	35	Recall	0,88
<i>Batch Size</i>	32	F1-Score	0,88
<i>Optimizer</i>	Nadam	Loss	0,48

Untuk mengevaluasi hasil pelatihan, kita dapat menggunakan Confusion Matrix yang memberikan informasi tentang jumlah data yang berhasil dideteksi dengan benar atau salah oleh sistem. Gambar 6 berikut menunjukkan Confusion Matrix dari hasil kinerja terbaik:

## Deteksi Penyakit Diabetes, Katarak Dan Glaukoma Pada Citra Fundus Retina Mata Manusia Menggunakan Cnn Dengan Arsitektur Alexnet



**Gambar 6. Hasil confusion matrix dari skenario terbaik**

Dari hasil Confusion Matrix yang terlihat pada Gambar 6, dapat dilihat bahwa dari total 1055 citra yang digunakan sebagai data uji, sebanyak 923 citra berhasil terdeteksi dengan benar, sementara 165 citra terdeteksi secara salah.

### 4. KESIMPULAN

Berdasarkan hasil penelitian dapat ditarik beberapa kesimpulan yaitu : CNN AlexNet mampu mengklasifikasikan dengan akurasi 87%, serta memiliki nilai daya ingat, skor F1, dan akurasi yang serupa, yaitu 88%, dan loss 0,48. Pengubahan parameter seperti ukuran gambar 128×128 piksel, kecepatan pembelajaran 0,0001, epoch 35, ukuran batch 32, dan pengoptimal Nadam memberikan skor terbaik. Setiap parameter uji memiliki pengaruh signifikan terhadap kinerja sistem, nilai terlalu besar/kecil dapat mengakibatkan tingkat kesalahan yang berbeda. Penelitian ini merekomendasikan penggunaan AlexNet dengan parameter di atas.

### DAFTAR RUJUKAN

- G. P. Kumasela, J. S. M. Saerang, dan L. Rares, "HUBUNGAN WAKTU PENGGUNAAN LAPTOP DENGAN KELUHAN PENGLIHATAN PADA MAHASISWA FAKULTAS KEDOKTERAN UNIVERSITAS SAM RATULANGI," *Jurnal e-Biomedik*, vol. 1, no. 1, Mar 2013, doi: 10.35790/ebm.1.1.2013.4361.
- Rarasmaya Indraswari, Wiwiet Herulambang, dan Rika Rokhana, "Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN)," *Techno.COM*, vol. 21, no. 2, hlm. 378–389, Mei 2022.
- M. J. Burton dkk., "The Lancet Global Health Commission on Global Eye Health: vision beyond 2020," *Lancet Glob Health*, vol. 9, no. 4, hlm. e489–e551, Apr 2021, doi: 10.1016/S2214-109X(20)30488-5.
- N. Gour, M. Tanveer, dan P. Khanna, "Challenges for ocular disease identification in the era of artificial intelligence," *Neural Comput Appl*, Jan 2022, doi: 10.1007/s00521-021-06770-5.
- V. Gulshan dkk., "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," *JAMA*, vol. 316, no. 22, hlm. 2402, Des 2016, doi: 10.1001/jama.2016.17216.
- A. Diaz-Pinto, S. Morales, V. Naranjo, T. Köhler, J. M. Mossi, dan A. Navea, "CNNs for automatic glaucoma assessment using fundus images: an extensive validation," *Biomed Eng Online*, vol. 18, no. 1, hlm. 29, Des 2019, doi: 10.1186/s12938-019-0649-y.
- L. Li, M. Xu, X. Wang, L. Jiang, dan H. Liu, "Attention Based Glaucoma Detection: A Large-Scale Database and CNN Model," dalam 2019 IEEE/CVF Conference on Computer Vision

- and Pattern Recognition (CVPR), IEEE, Jun 2019, hlm. 10563–10572. doi: 10.1109/CVPR.2019.01082.
- M. N. Bajwa dkk., "Correction to: Two-stage framework for optic disc localization and glaucoma classification in retinal fundus images using deep learning," *BMC Med Inform Decis Mak*, vol. 19, no. 1, hlm. 153, Des 2019, doi: 10.1186/s12911-019-0876-y.
- R. C. Joshi, M. K. Dutta, P. Sikora, dan M. Kiac, "Efficient Convolutional Neural Network Based Optic Disc Analysis Using Digital Fundus Images," dalam *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, Jul 2020, hlm. 533–536. doi: 10.1109/TSP49548.2020.9163560.
- A. Krizhevsky, I. Sutskever, dan G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, hlm. 84–90, Mei 2017, doi: 10.1145/3065386.
- R. Indraswari, T. Kurita, A. Z. Arifin, N. Suciati, dan E. R. Astuti, "Multi-projection deep learning network for segmentation of 3D medical images," *Pattern Recognit Lett*, vol. 125, hlm. 791–797, Jul 2019, doi: 10.1016/j.patrec.2019.08.003.
- R. Indraswari, R. Rokhana, dan W. Herulambang, "Melanoma image classification based on MobileNetV2 network," *Procedia Comput Sci*, vol. 197, hlm. 198–207, 2022, doi: 10.1016/j.procs.2021.12.132.
- RD. Kusumanto dan Alan Novi Tomponu, "PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB," *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, 2011.
- Suyanto, Kurniawan Nur Ramadhani, dan Satria Mandala, *Deep learning : Modernisasi Machine Learning Untuk Big Data*, 978-623-7131-21-2 ed. Penerbit Informatika Bandung, 2019.
- H. A. Nugroho dkk., "Performance of Convolutional Neural Network in Detecting Plasmodium Parasites," dalam *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, IEEE, Okt 2019, hlm. 331–336. doi: 10.1109/ICSEngT.2019.8906303.
- C. Sinthanayothin dkk., "Automated detection of diabetic retinopathy on digital fundus images," *Diabetic Medicine*, vol. 19, no. 2, hlm. 105–112, Mar 2002, doi: 10.1046/j.1464-5491.2002.00613.x.
- M. Bejiga, A. Zeggada, A. Nouffidj, dan F. Melgani, "A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery," *Remote Sens (Basel)*, vol. 9, no. 2, hlm. 100, Jan 2017, doi: 10.3390/rs9020100.
- W. Mo, X. Luo, Y. Zhong, dan W. Jiang, "Image recognition using convolutional neural network combined with ensemble learning algorithm," *J Phys Conf Ser*, vol. 1237, no. 2, hlm. 022026, Jun 2019, doi: 10.1088/1742-6596/1237/2/022026.
- W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, Mar 2016, doi: 10.12962/j23373539.v5i1.15696.
- T. Iskandar Zulkarnain Maulana Putra, A. Farhan Bukhori, dan Ilmu Pengetahuan Alam, dan U. Gadjah Mada, "Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia (Classification Model Based on Multiclass Classification with a Combination of Indobert Embedding and Long Short-Term Memory for Indonesian-language Tweets)," *Jurnal Ilmu Siber dan Teknologi Digital (JISTED)*, vol. 1, no. 1, hlm. 1–28, 2022, doi: 10.35912/jisted.v1i1.1509.