

KLASIFIKASI PENYAKIT MATA MANUSIA PADA CITRA *FUNDUS* RETINA MENGUNAKAN *SIMPLE CNN ARCHITECTURE*

Vivin Kusuma Wijayanti

Program Studi S-1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231

E-mail : vivin.20064@mhs.unesa.ac.id

Rr. Hapsari Peni Agustin Tjahyaningtjas

Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231

E-mail : hapsaripeni@unesa.ac.id

Abstrak

Berdasarkan data World Health Organization (WHO), terdapat sekitar 18 juta orang di dunia yang mengalami kebutaan, sehingga deteksi dini gangguan mata menjadi langkah penting untuk membantu penanganan yang lebih cepat dan tepat. Namun, proses diagnosis penyakit mata umumnya masih dilakukan melalui rangkaian pemeriksaan manual, yang berisiko memunculkan *human error* dan membutuhkan waktu relatif lebih lama. Penelitian ini mengusulkan pendekatan Simple Convolutional Neural Network (Simple CNN) dengan dropout variatif pada setiap lapisan (0,2–0,5) untuk menekan *overfitting* dan meningkatkan kinerja klasifikasi citra fundus retina. Model dilatih menggunakan hyperparameter Adam optimizer, ReLU activation function, batch size 32, dan epoch 100, lalu dibandingkan dengan model CNN lain yang diuji pada dataset fundus retina dari Kaggle berisi 4.000 citra (.jpg) dengan empat kelas (normal, katarak, glaukoma, dan retinopati diabetik). Hasil evaluasi menunjukkan bahwa arsitektur Simple CNN dengan dropout variatif memberikan performa terbaik dibandingkan model pembanding, dengan akurasi 87,96%, precision 88,03%, recall 87,96%, dan F1-score 87,99%. Temuan ini mengindikasikan bahwa rancangan Simple CNN yang sederhana namun teratur melalui strategi dropout bertingkat berpotensi menjadi pendekatan yang efektif untuk membantu proses klasifikasi penyakit mata berbasis citra fundus retina.

Kata Kunci: klasifikasi, penyakit mata, *simple CNN*, akurasi.

Abstract

Based on data from the World Health Organization (WHO), approximately 18 million people worldwide experience blindness, making early detection of eye disorders essential to enable faster and more appropriate treatment. However, eye disease diagnosis is still commonly conducted through a series of manual examinations, which can be time-consuming and prone to human error. This study proposes a Simple Convolutional Neural Network (Simple CNN) approach with layer-wise varying dropout (0.2–0.5) to reduce overfitting and improve the classification performance of retinal fundus images. The model was trained using the following hyperparameters: the Adam optimizer, ReLU activation function, batch size of 32, and 100 epochs, and was compared with other CNN models. Experiments were conducted on a Kaggle retinal fundus dataset comprising 4,000 .jpg images across four classes (normal, cataract, glaucoma, and diabetic retinopathy). The evaluation results show that the proposed Simple CNN with varying dropout achieved the best performance among the compared models, attaining an accuracy of 87.96%, precision of 88.03%, recall of 87.96%, and F1-score of 87.99%. These findings indicate that a simple yet well-regularized Simple CNN architecture employing staged dropout can serve as an effective approach to support eye disease classification using retinal fundus images.

Keywords: classification; eye disease; Simple CNN; accuracy.

PENDAHULUAN

Gangguan penglihatan masih menjadi persoalan kesehatan yang berdampak langsung pada kualitas hidup, produktivitas, dan beban layanan kesehatan. Di antara berbagai penyebabnya, katarak menempati posisi penting karena kontribusinya yang besar terhadap kebutaan;

laporan Infodatin Kementerian Kesehatan RI merangkum bahwa katarak merupakan penyebab utama kebutaan, termasuk di Indonesia (Kementerian Kesehatan RI, 2014). Katarak sendiri ditandai oleh kekeruhan pada lensa yang dapat berkembang perlahan hingga menurunkan ketajaman penglihatan (Asmara et al., 2023). Selain katarak, glaukoma dan retinopati

diabetik juga dikenal sebagai gangguan serius yang dapat berujung pada kerusakan penglihatan permanen apabila terlambat terdeteksi (Papeo et al., 2023; Oktavia et al., 2022). Karena itu, upaya deteksi dini dan skrining berbasis bukti menjadi krusial, terutama untuk kelompok berisiko.

Pemeriksaan klinis berbasis citra, terutama citra fundus retina, menjadi salah satu pendekatan yang banyak dipakai untuk membantu identifikasi berbagai kondisi mata. Citra fundus merekam struktur penting seperti papil optik, pembuluh darah, dan area makula sehingga perubahan patologis dapat diamati melalui pola visual tertentu. Contoh citra fundus retina mata normal ditunjukkan pada gambar berikut.

Seiring meningkatnya ketersediaan data citra dan komputasi, pendekatan kecerdasan artifisial semakin relevan untuk mendukung analisis medis berbasis citra (Agatha et al., 2023). Pada ranah ini, Convolutional Neural Network (CNN) menonjol karena kemampuannya mengekstraksi fitur visual secara otomatis, mulai dari pola sederhana hingga representasi yang lebih kompleks, tanpa bergantung pada perancangan fitur manual. Secara konseptual, alur umum CNN dapat dipahami sebagai rangkaian *feature learning* (konvolusi dan pooling) yang diakhiri tahap klasifikasi (flatten–fully connected–softmax), seperti ilustrasi berikut.

Meski demikian, temuan penelitian terdahulu menunjukkan bahwa capaian klasifikasi penyakit mata pada citra fundus retina masih bervariasi dan pada sejumlah studi masih berada pada rentang yang relatif moderat, sehingga ruang perbaikan masih terbuka (Cahya et al., 2021; Juniati & Suwanda, 2022; Mutawalli et al., 2023). Pada pendekatan berbasis fitur dan kluster, misalnya, Juniati dan Suwanda (2022) memanfaatkan dimensi fraktal *box counting* dan *fuzzy K-means* pada dataset Kaggle, namun akurasi yang dilaporkan masih sekitar 46%. Hutabri et al. (2019) menerapkan PCA dan K-NN untuk deteksi katarak pada dataset Kaggle dengan akurasi sekitar 50%. Firdaus et al. (2022) mengembangkan sistem klasifikasi katarak berbasis CNN (web-based) menggunakan dataset Kaggle dengan akurasi sekitar 52,20%.

Pada sisi deep learning, beberapa studi memperlihatkan peningkatan performa namun tetap beragam. Handono et al. (2020) menunjukkan potensi CNN dalam mengenali pola patologis pada citra retina untuk deteksi retinopati diabetik. Marcella et al. (2020) menggunakan CNN arsitektur VGG-19 (pra-latih ImageNet) dan memperoleh akurasi sekitar 65,29%, mengindikasikan bahwa arsitektur yang lebih dalam dapat meningkatkan representasi fitur meski kebutuhan komputasinya ikut bertambah. Mutawalli et al. (2023) membandingkan CNN dan ResNet untuk klasifikasi tingkat keganasan diabetes pada citra retinopati (dataset Kaggle) dengan akurasi sekitar 68,52%. Indaswari et al. (2022) juga melaporkan bahwa CNN dapat digunakan

untuk mendeteksi penyakit mata pada citra fundus dengan akurasi sekitar 72%.

Jika ditarik benang merah, variasi performa tersebut mengisyaratkan bahwa persoalannya bukan semata “apakah CNN digunakan atau tidak”, melainkan bagaimana arsitektur dirancang dan distabilkan agar mampu mempelajari pola citra secara efektif tanpa jatuh pada overfitting (Handono et al., 2020; Marcella et al., 2022; Mutawalli et al., 2023; Indaswari et al., 2022). Dalam naskah ini, overfitting dipahami sebagai situasi ketika model terlalu “hafal” data latih sehingga kinerjanya menurun pada data baru; karena itu diperlukan strategi regularisasi yang membuat model lebih general (Srivastava et al., 2014). Dengan kata lain, kontribusi riset yang lebih “mengikat” dapat diarahkan pada rancangan model yang tetap ringan namun memiliki mekanisme regularisasi yang tepat sehingga peningkatan performa tidak semata lahir dari kompleksitas arsitektur, melainkan dari stabilitas pembelajaran.

Berangkat dari celah tersebut, penelitian ini mengambil posisi yang lebih praktis: bukan mengejar arsitektur yang semakin rumit, tetapi menyusun Simple CNN yang tetap ringan (parameter lebih sedikit) dan dioptimalkan melalui dropout variatif pada beberapa layer (0,2 hingga 0,5) untuk menekan overfitting dan mendorong peningkatan akurasi. Pendekatan ini diuji pada dataset open data Kaggle berjumlah 4.000 citra fundus retina dalam empat kelas (normal, cataract, glaucoma, dan diabetic retinopathy). Dengan rancangan tersebut, penelitian ini juga menekankan perbandingan performa terhadap model CNN lain yang diuji, agar kontribusi peningkatan kinerja dapat terlihat secara lebih objektif.

Secara lebih terarah, fokus penelitian ini dirumuskan ke dalam dua pertanyaan utama: (1) bagaimana membangun model arsitektur klasifikasi penyakit mata menggunakan Simple CNN, dan (2) bagaimana kinerja model tersebut pada data uji. Selaras dengan itu, tujuan penelitiannya adalah menjelaskan proses pembangunan model dan mengukur kinerja model klasifikasi penyakit mata berbasis Simple CNN.

METODE

Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Telekomunikasi, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya. Waktu penelitian dilaksanakan selama 6 bulan yaitu pada semester genap 2023/2024.

Instrumen Penelitian

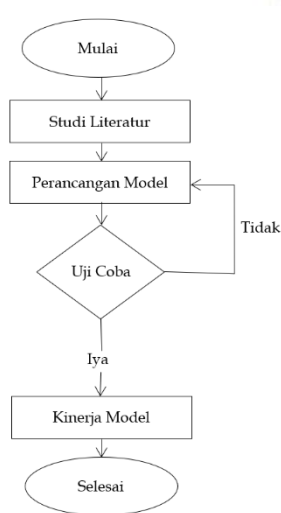
Instrumen penelitian adalah sebuah alat bantu yang digunakan oleh peneliti untuk mengumpulkan data penelitian yang relevan dan dilakukan secara sistematis. Dalam penelitian ini, instrumen penelitian yang

digunakan adalah laptop dan *platform Google Collab* yang digunakan untuk menganalisis data citra. Data kuantitatif berupa citra yang diperoleh dari *open data Kaggle*.

Dataset

Pengambilan dataset ini diperoleh dari *open data Kaggle* sebagai berikut (<https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification-tahun-2022>). Dengan sumber data yang diperoleh dari *IDRiD*, *HRF*, *Oculur Recognition*, dan *retinal* dataset. Dataset *Eye Disease Classification* berisi kumpulan jenis mata normal, katarak, glaukoma dan retinopati diabetik. Data yang diperoleh berjumlah 4.000 citra yang terdiri atas 1.000 citra mata normal, 1.000 citra mata katarak, 1.000 citra mata glaukoma, dan 1.000 citra mata retinopati diabetik.

Rancangan Penelitian



Gambar 1 Diagram Alir Alur Penelitian

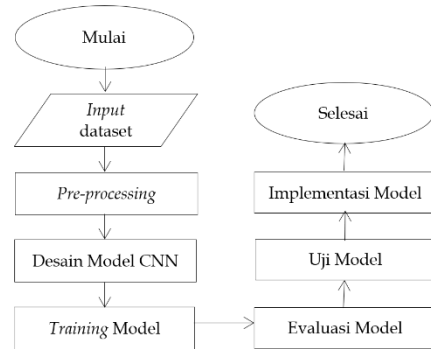
Prosedur Penelitian

Studi Literatur

Studi literatur dilaksanakan dengan mempelajari beberapa buku dan jurnal terkait tentang klasifikasi jenis penyakit mata pada citra fundus retina menggunakan *Simple CNN architecture* serta bahan referensi lain yang terkait dengan penelitian ini.

Perancangan Desain Model

Dalam merancang desain model arsitektur CNN, platform yang digunakan adalah *Google Collab*. Proses merancang desain arsitektur CNN dimulai dari pengolahan citra hingga klasifikasi.



Gambar 2 Diagram Alir Metode CNN

HASIL DAN PEMBAHASAN

Hasil

Hasil pengujian dari implementasi metode *Simple Convolutional Neural Network* dengan beberapa skenario dalam mengklasifikasikan jenis penyakit mata pada citra fundus retina.

Tahapan Pembuatan Model

Path Citra Dataset

```

Input : Image dataset
Output : Set and Print classes of dataset

1. Initialize library python(os, matplotlib, numpy, pandas, tensorflow).
2. Initialize batch normalization
3. Mounting dataset('/content/gdrive')
4. Directory dataset(dataset_mata2_dir = 'content/gdrive/.../').
5. Set classes and print all of the dataset(print('Jumlah data X:', len(os.listdir(X)))).
    
```

Gambar 3 Pseudo-code dari Tahapan Persiapan Dataset.

Langkah pertama dalam mempersiapkan dataset adalah mendeklarasikan modul *python* dan *import library* yang dibutuhkan. Langkah-langkah pemrograman yang dilakukan dinyatakan dalam *pseudo-code* berikut.

Dalam penelitian ini, digunakan dataset yang berisi 4000 citra fundus retina yang terdiri 1000 citra mata normal, 1000 citra mata katarak, 1000 citra mata glaukoma, dan 1000 citra mata retinopati diabetik. Setelah dataset siap, langkah selanjutnya adalah melakukan *pre-processing*.

Pre-Processing

Pada tahap ini, dataset dimuat ke dalam program dalam bentuk *array*. Representasi dalam bentuk *array* ini memudahkan untuk *pre-processing* citra. Tahapan dalam

melakukan *pre-processing* citra dapat dilihat pada Gambar 4.

```
Input : Image dataset
Output : Resize image
1. Initialize          resize
   image(width and height).
2. See name of class from
   directory(dataset_mata2='/co
   ntent/gdrive.../').
3. Print the name of class.
```

Gambar 4 *Pseudo-code* dari Tahapan *Pre-processing*

Berdasarkan Gambar 4, ditampilkan *pseudo-code* yang menjelaskan tahapan dari *pre-processing*. Pada tahap ini, dilakukan perubahan ukuran (*resize*) gambar menjadi 128x128 piksel. Langkah ini bertujuan untuk mempercepat proses pengolahan data. Selain itu, langkah ini juga menampilkan nama-nama kelas yang terdapat dalam direktori dataset. Tahap berikutnya adalah melakukan pembagian dataset (*splitting data*) yang terdapat pada Gambar 5.

Split Dataset Training dan Testing

Setelah seluruh dataset digabungkan menjadi satu, langkah berikutnya adalah membagi dataset menjadi data *training* dan data *testing*. Data *training* digunakan untuk melatih model agar dapat mengenali pola sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model yang belum pernah dilakukan sebelumnya. Proses pembagian dataset ini dapat dilihat pada *pseudo-code* berikut.

```
Input : Image dataset 128x128
        pixel
Output : Splitting data train and
        test
1. Image data generator(training
   and testing).
2. Split Data 0.2
3. Training generator and Testing
   generator.
```

Gambar 5 *Pseudo-code* dari Proses *Splitting Data*

Gambar 5 menampilkan *pseudo-code* dari proses *splitting data*. Pada tahap ini, gambar-gambar diubah ukurannya menjadi 128x128 piksel (*resize image*). Setelah itu, dataset dibagi menjadi data *training* sebesar 80% dan data *testing* sebesar 20%. Pembagian data ini bertujuan untuk memastikan model dapat dilatih dengan data yang cukup serta dievaluasi pada data yang belum pernah diujikan. Pada bagian ini, *class mode* yang digunakan adalah *categorical* karena dataset terdiri dari empat kelas (*normal, cataract, glaucoma, diabetic retinopathy*). Dengan menggunakan mode ini, label data akan direpresentasikan sesuai dengan masing-masing kelas. Tahap selanjutnya adalah melakukan pembuatan model arsitektur *simple CNN* yang terdapat pada Gambar 6.

Pembuatan Model *Simple CNN*

```
Input : Image, Label,
Output : Trained Model
1. Input image.
2. Convolution Layer 1 (model.add).
3. Convolution Layer 2 (model.add).
4. Convolution Layer 3 (model.add).
5. Fully Connected Layer (Flatten Layer).
6. Output Layer.
```

Gambar 6 *Pseudo-code* dari Model *Simple CNN*

Berdasarkan Gambar 6 menampilkan *pseudo-code* dari pembuatan model arsitektur *simple CNN* yang terdapat penambahan pada *layer*. Terdapat penambahan pada convolutional layer sebanyak tiga layer dengan hyperparameter pada bagian convolutional layer seperti ReLU dan terdapat penambahan dropout untuk mengurangi terjadinya overfitting. Overfitting adalah keadaan saat suatu model mempelajari data training terlalu baik dan spesifik sehingga membuatnya berkinerja buruk pada data baru. Dropout bekerja dengan acak menonaktifkan sejumlah unit dalam jaringan selama proses training sehingga menghasilkan model yang lebih umum dan dapat diterapkan dengan baik.

Terdapat perbedaan antara arsitektur yang diajukan dengan arsitektur CNN dasar. CNN dasar memiliki kekurangan yaitu memiliki arsitektur sederhana yang tidak dioptimalkan dengan baik sehingga menyebabkan training yang lebih lambat dan CNN dasar memiliki sedikit lapisan konvolusi, sehingga kemampuannya untuk menangkap pola kompleks dalam data menjadi terbatas. Sedangkan, *simple CNN* yang diajukan memiliki keunggulan yaitu arsitektur yang dirancang secara sederhana dengan parameter yang lebih sedikit, dapat melakukan training data dalam jumlah banyak, mengurangi overfitting dengan penggunaan dropout agar dapat menghasilkan nilai akurasi yang lebih tinggi.

Training dan Testing Data

Pada tahap ini, dilakukan proses *training data* yang telah dibagi menggunakan *validation split*. Proses *training data* dapat dilihat pada Gambar 4.6 sebagai berikut.

```
Input : Model Simple CNN
Output : Save Model
1. Initialize optimizer.
2. Defined the epoch number.
3. Training data.
4. Testing data.
5. Save          model
   (model_path="/content/gdrive/.../Mode
   l/best_model-e100.h5"
   Model.save(model_path).
```

Gambar 7 *Pseudo-code* dari Model *Simple CNN*

Berdasarkan Gambar 7 menampilkan *pseudo-code* dari proses *training data* dengan model yang telah

dibangun sebelumnya. Pada tahap ini, melakukan inisialisasi *optimizer* menggunakan ADAM. Setelah itu, ditentukan jumlah *epoch* yang akan digunakan, yaitu 50, 70, dan 100.

Prediksi

Setelah model selesai *training*, selanjutnya model yang telah dilakukan *training* sebelumnya digunakan untuk menguji dengan data *testing*. Tahap selanjutnya adalah proses prediksi.

```
Input : Testing and training model
Output : Print prediction result
1. Model
   evaluate(results=model.evaluate)
   .
2. Print test loss and test accuracy.
3. Model
   predict(y_pred=(model.predict(test_gen))).
4. Print prediction result.
```

Gambar 8 Pseudo-code dari Prediksi

Pada Gambar 8 ditampilkan *pseudo-code* dari proses prediksi untuk mengevaluasi akurasi model tersebut. Fungsi *testing* dibuat menggunakan variabel *y_pred*, yang digunakan untuk memprediksi label pada data uji dan mengubah nilai 0 dan 1 untuk merepresentasikan *true* dan *false*. Selanjutnya, untuk melakukan proses klasifikasi dapat dilakukan dengan menggunggah gambar berformat *.jpg*. Sistem klasifikasi akan berjalan berdasarkan gambar yang telah diunggah tersebut.

Confusion Matrix

Fungsi *confusion matrix* dirancang untuk mengevaluasi performa model yang digunakan, seperti yang ditampilkan pada Gambar 9.

```
1. Confusion matrix.
2. Initialize cmap.
3. Classification
   Report(accuracy, f1 score,
   recall, precision).
```

Pada Gambar 10 ditampilkan *pseudo-code* dari fungsi *confusion matrix* untuk mengevaluasi model dengan deklarasi *cmap*, kelas target, dan tampilan warna *font*. Setelah menghitung nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)* dari *confusion matrix*. Langkah selanjutnya adalah menghasilkan *classification report*, yaitu nilai *Accuracy*, *F1 Score*, *Precision*, dan *Recall*.

Uji Coba Model

Dalam penelitian ini, dilakukan pengujian terhadap model arsitektur pada *deep learning* dengan mengevaluasi penggunaan yang telah diterapkan pada

beberapa penelitian sebelumnya. Pengujian ini menggunakan ADAM *optimizer* dan ReLU *activation function*, dengan jumlah *epoch* yang bervariasi, yaitu 50, 70 dan 100. Tujuan dari uji coba ini adalah untuk mengetahui kinerja model arsitektur seperti *accuracy*, *f1 score*, *precision*, dan *recall*.

Hasil Performa

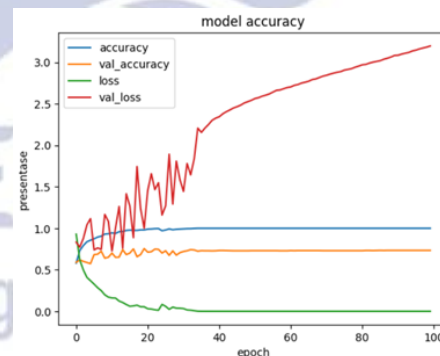
Penelitian ini bertujuan untuk mengklasifikasikan jenis penyakit mata dengan menggunakan tiga metode pengujian yang telah dijelaskan sebelumnya. Berikut adalah penjelasan mengenai pengujian yang dilakukan pada tiga model yang digunakan yaitu model CNN dasar, model CNN-1, dan model CNN-2 (*Simple CNN*) yang diusulkan dalam penelitian ini.

Model CNN Dasar

Skenario pengujian pertama menggunakan arsitektur CNN dasar dengan *hyperparameter* menggunakan *optimizer* ADAM dan *activation function* ReLU serta menggunakan jumlah *epoch* yang berbeda yaitu 50, 70, dan 100. Tabel 4.1 menampilkan hasil performa dari model arsitektur CNN dasar.

Tabel 1 Hasil Performa CNN Dasar

Hyper parameter	Epoch	CNN			
		Performa (%)			
		Precision	Recall	F1-Score	Akurasi
Opt.ADAM	50	64,56	63,46	66,08	63,46
Act. Funct	70	73,84	68,64	71,08	68,64
ReLU	100	84,11	73,37	78,37	73,37



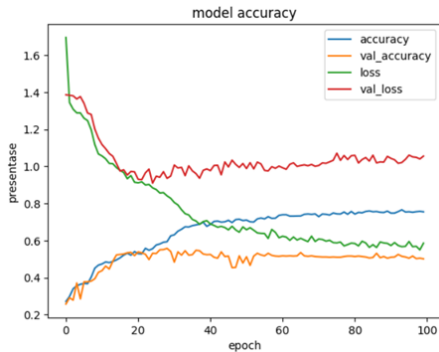
Gambar 10 Grafik Akurasi dan Loss Arsitektur CNN dasar Epoch 100

Model CNN-1

Skenario pengujian kedua menggunakan modifikasi arsitektur CNN dengan *hyperparameter* menggunakan *optimizer* ADAM, *activation function* ReLU dan menggunakan nilai dropout yang sama yaitu sebesar 0.8 serta menggunakan jumlah *epoch* yang berbeda yaitu 50, 70, dan 100. Tabel 2 menampilkan hasil performa dari model arsitektur CNN-1.

Tabel 2 Hasil Performa CNN-1

CNN					
Hyper parameter	Epoch	Performa (%)			
		Precision	Recall	F1-Score	Akurasi
Opt.ADAM Act. Funct ReLU	100	72,44	49,79	59,01	50,12



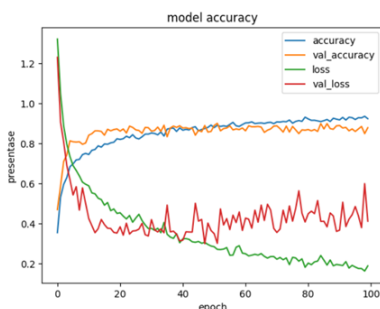
Gambar 11 Grafik Akurasi dan Loss Arsitektur CNN-1 Epoch 100

Model CNN-2 (Simple CNN)

Skenario pengujian untuk model *simple CNN* yang diusulkan menggunakan arsitektur CNN dengan *hyperparameter* menggunakan *optimizer* ADAM, *activation function* ReLU, nilai *dropout* yang berbeda tiap layer dimulai dari 0.2-0.5 serta menggunakan jumlah epoch yang berbeda yaitu 50, 70, dan 100. Tabel 3 menampilkan hasil performa dari model arsitektur CNN-2 (*Simple CNN*).

Tabel 3 Hasil Performa CNN-2 (Simple CNN)

CNN					
Hyper parameter	Epoch	Performa (%)			
		Precision	Recall	F1-Score	Akurasi
Opt.ADAM	50	66,58	65,44	65,46	65,44
Act. Funct	70	73,68	72,04	72,01	72,04
ReLU	100	88,03	87,96	87,99	87,96



Gambar 12 Grafik Akurasi dan Loss Arsitektur Simple CNN Epoch 100

Pembahasan

Dalam penelitian ini melakukan klasifikasi jenis penyakit mata menggunakan citra fundus retina dengan

menggunakan tiga pengujian yang telah dijelaskan sebelumnya. Pada CNN-1 hanya dilakukan penelitian menggunakan nilai epoch sebesar 100 saja. Hal ini dikarenakan penulis menggunakan nilai epoch terbaik selama penelitian untuk pembandingan dengan CNN-2 yang diusulkan. Berdasarkan hasil pengujian yang telah dilakukan perbandingan model arsitektur CNN dasar, CNN-1, dan CNN-2 (*Simple CNN*), hasil performa model terbaik selama penelitian menggunakan model arsitektur CNN-2 (*Simple CNN*) dengan *hyperparameter* ADAM *optimizer*, ReLU *activation function*, jumlah Epoch yang digunakan sebesar 100 serta menggunakan nilai *dropout* yang bervariasi di tiap layer nya mulai dari 0.2-0.5. Dalam penelitian ini, penggunaan *dropout* yang bervariasi untuk lapisan yang berbeda. Pada lapisan awal digunakan nilai *dropout* yang lebih rendah (0.2), gunanya untuk mengekstraksi fitur dasar dari dataset. Kemudian untuk lapisan tengah atau akhir digunakan nilai *dropout* yang lebih tinggi (0.3, 0.4, 0.5) untuk menangkap fitur yang lebih kompleks dan digunakan untuk mengurangi *overfitting* pada fitur.

Alasan penggunaan *maxpooling* setelah *Conv2D* adalah *Conv2D* bertugas untuk mengekstraksi fitur dari data input, setelah *Conv2D*, output biasanya berupa fitur peta yang detail dan berukuran besar. *Maxpooling* digunakan untuk menyederhanakan representasi fitur tersebut dan memastikan fitur utama tetap terjaga sambil menurunkan dimensi. Selain itu, *maxpooling* juga dapat mempertahankan fitur utama sambil menghilangkan informasi yang tidak dibutuhkan serta dapat memastikan model fokus pada gambar meskipun objek bergeser sedikit tetapi fitur tetap terdeteksi. *Maxpooling* juga dapat membantu menghilangkan *noise* yang tidak signifikan dalam fitur.

Pada arsitektur model CNN-2 terdapat penambahan *dropout* dan *dense* di layer akhir. Hal ini dikarenakan *dense layer* di akhir digunakan untuk model dapat belajar pola lebih kompleks dari fitur yang dihasilkan oleh CNN dengan lebih banyak neuron dan parameter serta mengubah fitur CNN menjadi prediksi akhir. Untuk menghindari model terlalu mengafal data training atau *overfitting* maka digunakan *dropout layer*. *Dropout* memaksa model untuk *generalize* karena tidak semua neuron selalu aktif. Karena mematikan sebagian neuron, perhitungan jadi lebih efisien dan dapat mempercepat training model. Penambahan keduanya pada lapisan akhir membuat model lebih akurat dan stabil saat diuji dengan data baru.

Simple CNN merupakan arsitektur CNN yang terdapat penambahan parameter *dropout* yang memiliki nilai yang bervariasi di tiap layernya. Model yang diusulkan menghasilkan nilai akurasi sebesar 87.96%, *f1-score* sebesar 87.99%, *precision* sebesar 88.03%, dan *recall* sebesar 87.96%. Sisa hasil confusion matrix merupakan data yang salah diklasifikasikan. Hal ini bisa disebabkan oleh ketidakseimbangan kelas, noise dalam kelas, atau model yang kurang optimal. Sisa hasil confusion matrix dapat digunakan untuk berbagai tujuan dalam evaluasi model dan perbaikan model. Beberapa

kemungkinan kemana data ini bisa diarahkan yaitu dapat digunakan untuk analisis error, menganalisis pola kesalahan untuk memahami mengapa model salah. Digunakan untuk training ulang, agar model lebih akurat. Digunakan dalam teknik data augmentasi, untuk meningkatkan keberagaman data. Dan disimpan sebagai data tes tambahan. Data yang salah diklasifikasikan dapat digunakan untuk dataset tes tambahan pada iterasi berikutnya.

Dibandingkan dengan beberapa model arsitektur lainnya dalam penelitian yang telah penulis lakukan yang menghasilkan nilai akurasi di bawah model yang diusulkan. Hal ini dikarenakan metode yang diusulkan menggunakan metode klasifikasi *deep learning* yang dapat mengatasi dataset dalam jumlah yang lebih banyak.

Sejumlah penelitian terdahulu menunjukkan bahwa performa klasifikasi penyakit mata berbasis citra fundus retina masih sangat beragam, bergantung pada pendekatan dan rancangan model yang digunakan. Pada kelompok metode non-deep learning, Suwanda dkk. (2022) menggunakan dimensi fraktal (box counting) yang dipadukan dengan Fuzzy K-Means pada dataset Kaggle, namun akurasi masih berada di kisaran 46%. Hasil yang relatif mirip juga terlihat pada Hutabri dkk. (2019) yang menerapkan PCA dan K-NN untuk deteksi katarak pada dataset Kaggle, dengan akurasi sekitar 50%. Masih pada rentang yang hampir sama, Firdaus dkk. (2022) mengembangkan klasifikasi katarak berbasis CNN (web-based) pada dataset Kaggle, tetapi akurasi dilaporkan sekitar 52,20%.

Ketika pendekatan beralih ke deep learning, capaian akurasi cenderung meningkat, meskipun belum konsisten tinggi. Handono dkk. (2020) melaporkan implementasi CNN untuk deteksi retinopati diabetik pada dataset Kaggle dengan akurasi sekitar 58%. Marcella dkk. (2020) kemudian menggunakan arsitektur yang lebih dalam, yaitu VGG-19, dengan rujukan dataset ImageNet, dan memperoleh akurasi sekitar 65,29%. Pada studi yang membandingkan arsitektur, Mutawalli dkk. (2023) mengkomparasikan CNN dengan ResNet untuk klasifikasi tingkat keganasan diabetes berbasis citra retinopathy pada dataset Kaggle, dengan akurasi sekitar 68,52%. Sementara itu, Indaswari dkk. (2022) melaporkan deteksi penyakit mata pada citra fundus menggunakan CNN dengan acuan dataset ImageNet, dan akurasi berada pada kisaran 72%.

Berangkat dari peta hasil tersebut, metode yang diusulkan dalam naskah ini mengambil posisi yang lebih “ringan namun ditata”: menggunakan Simple CNN tetapi diperkuat dengan dropout variatif antar-layer (0,2–0,5) untuk menekan overfitting, dan diuji pada dataset Kaggle berisi 4.000 citra dalam empat kelas. Hasil evaluasi menunjukkan performa terbaik pada skenario epoch 100, dengan akurasi 87,96%, disertai precision 88,03%, recall 87,96%, dan F1-score 87,99%.

Confusion Matrix

	cataract	diabetic_retinopathy	glaucoma	normal
Actual				
cataract	116	0	4	80
diabetic_retinopathy	1	199	0	0
glaucoma	7	2	92	99
normal	7	0	13	180
	cataract	diabetic_retinopathy	glaucoma	normal
Predicted				

Gambar 13 Confusion Matrix dari Model Arsitektur CNN Dasar

Confusion Matrix

	cataract	diabetic_retinopathy	glaucoma	normal
Actual				
cataract	9	8	0	183
diabetic_retinopathy	0	198	0	2
glaucoma	0	9	2	189
normal	0	7	1	192
	cataract	diabetic_retinopathy	glaucoma	normal
Predicted				

Gambar 14 Confusion Matrix dari Model Arsitektur CNN-1

Confusion Matrix

	cataract	diabetic_retinopathy	glaucoma	normal
Actual				
cataract	133	0	17	10
diabetic_retinopathy	0	160	0	0
glaucoma	29	0	127	4
normal	5	0	12	143
	cataract	diabetic_retinopathy	glaucoma	normal
Predicted				

Gambar 15 Confusion Matrix dari Model Arsitektur CNN-2 (*simple CNN*)

Pada evaluasi performa menampilkan confusion matrix. Dari tiga pengujian yang dilakukan, hasil performa terbaik ditunjukkan pada model Arsitektur Simple CNN. Ilustrasi confusion matrix tiap pengujian terbaik terdapat pada Gambar 13, 14, dan 15.

Berdasarkan persamaan, model arsitektur yang diuji dapat diketahui nilai precision, recall, f1-score dan akurasi. Hasil perbandingan model terbaik tiap pengujian tersebut dapat dilihat pada Tabel 4.

Tabel 4 Perbandingan Hasil Performa Tiap Model

Arsitektur	Epoch	Accuracy	Precision	Recall	F1-Score
CNN Dasar	50	65,44	64,56	63,46	66,08
	70	72,04	73,84	68,64	71,08
	100	87,96	84,11	73,37	78,37
CNN-1	100	100	72,44	49,79	59,01
CNN-2 (Simple CNN)	50	65,44	66,58	65,44	65,46
	70	72,04	73,68	72,04	72,01
	100	87,96	88,03	87,96	87,99

Berdasarkan Tabel 4, performa terbaik dari model yang diujikan adalah *simple CNN* dengan akurasi sebesar 87.96%. Hal ini menunjukkan model tersebut dapat mengklasifikasikan jenis penyakit mata dengan sangat baik. Model ini juga memiliki nilai precision sebesar 88.03%. Sebaliknya, performa terendah dari model-model lainnya yaitu CNN-1 dengan konfigurasi *hyperparameter* ADAM optimizer, ReLU activation function dan jumlah epoch sebanyak 100 serta nilai dropout sebesar 0.8, yang hanya menghasilkan akurasi sebesar 59.01. Sedangkan, *simple CNN* berhasil mencapai akurasi lebih tinggi dibandingkan model CNN lainnya, yaitu sebesar 87.96%.

PENUTUP

Simpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa model Simple CNN yang dirancang dengan Adam optimizer dan ReLU activation function, serta menerapkan dropout variatif pada setiap lapisan (0,2; 0,3; 0,4; 0,5), menghasilkan performa terbaik pada skenario training 100 epoch. Pada konfigurasi ini, model mencapai akurasi tertinggi sebesar 87,96%, yang menunjukkan bahwa strategi regularisasi melalui dropout bertingkat mampu mendukung pembelajaran fitur yang lebih stabil dan meningkatkan kemampuan generalisasi model pada data uji.

Selain itu, evaluasi kinerja model pada konfigurasi terbaik tersebut menunjukkan hasil yang konsisten di berbagai metrik, yaitu accuracy 87,96%, precision 88,03%, recall 87,96%, dan F1-score 87,99%. Temuan ini mengindikasikan bahwa model tidak hanya unggul dari sisi akurasi, tetapi juga mampu menjaga keseimbangan antara ketepatan prediksi (precision) dan kemampuan menangkap kasus secara menyeluruh (recall), yang tercermin pada nilai F1-score yang tinggi. Secara umum, hasil eksperimen juga memperlihatkan bahwa peningkatan jumlah epoch selama proses training cenderung berkontribusi pada peningkatan performa, karena model memiliki kesempatan lebih besar untuk mempelajari pola-pola penting pada citra fundus retina. Dengan kata lain, semakin banyak epoch yang

digunakan, hingga batas tertentu, semakin matang proses pembelajaran, sehingga keluaran model menjadi lebih akurat dan stabil.

Saran

Berdasarkan hasil dan simpulan yang diperoleh, penelitian ini masih membuka ruang pengembangan pada studi lanjutan. Pertama, penelitian berikutnya dapat mempertimbangkan penggunaan arsitektur CNN lain yang memiliki kompleksitas dan jumlah parameter lebih besar dibandingkan model yang diusulkan, sehingga kemampuan representasi fitur dapat ditingkatkan terutama untuk membedakan kelas-kelas yang memiliki kemiripan visual. Kedua, peningkatan kualitas pembelajaran juga dapat dilakukan dengan memperkaya data training melalui teknik *data augmentation*, sehingga variasi citra yang dihadapi model menjadi lebih beragam. Strategi ini berpotensi memperkuat generalisasi model, mengurangi risiko *overfitting*, dan pada akhirnya meningkatkan stabilitas performa ketika model diterapkan pada data uji maupun data baru di luar dataset training.

DAFTAR PUSTAKA

- Adiningsih, M., Yulianti, D., & Wibowo, A. (2023). Analisis penerapan forecasting dalam menentukan kebutuhan storage pada pusat data Kementerian Keuangan. *JRST (Jurnal Riset Sains dan Teknologi)*, 7(2), 229. <https://doi.org/10.30595/jrst.v7i2.17579>
- Agatha, C., et al. (2023). Penerapan artificial intelligence dalam computer aided instruction (CAI). *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 1(1), 739–746. <https://doi.org/10.37396/jsc.v1i1.6>
- Aldoseri, A., Al-Khalifa, K. N., & Hamouda, A. M. (2023). Re-thinking data strategy and integration for artificial intelligence: Concepts, opportunities, and challenges. *Applied Sciences*, 13(12). <https://doi.org/10.3390/app13127082>
- Ananda, E. P. (2017). The relationship between knowledge, sickness period, and intraocular pressure to the quality of life of glaucoma patient. *Jurnal Berkala Epidemiologi*, 4(2), 288. <https://doi.org/10.20473/jbe.v4i2.2016.288-300>
- Artini, W., & Dame. (2011). Glaucoma caused blindness with its characteristic in Cipto Mangunkusumo Hospital. *Jurnal Oftalmologi Indonesia*, 7(5), 189–193.
- Asmara, D., Amri, M. F., Pramudito, N. B., Syahir, R., & Fithri, N. K. (2023). Gambaran kejadian katarak pada pekerja dengan paparan radiasi UV di lingkungan kerja. *Jurnal Kesehatan Tambusai*, 4(2), 2442–2451. <https://doi.org/10.31004/jkt.v4i2.15125>

- Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan akurasi, recall, dan presisi klasifikasi pada algoritma C4.5, random forest, SVM, dan naive Bayes. *Jurnal Media Informatika Budidarma*, 5(2), 640. <https://doi.org/10.30865/mib.v5i2.2937>
- Cahya, F. N., Hardi, N., Riana, D., & Hadiyanti, S. (2021). Klasifikasi penyakit mata menggunakan convolutional neural network (CNN). *Sistemasi*, 10(3), 618. <https://doi.org/10.32520/stmsi.v10i3.1248>
- Cholissodin, I., & Soebroto, A. A. (2021). *AI, machine learning & deep learning (teori & implementasi)*. (Data penerbit tidak dicantumkan.)
- Diantika, A. S., & Firmanto, Y. (2019). Implementasi machine learning pada aplikasi penjualan produk digital (studi pada GrabKios). *Jurnal Ilmiah Mahasiswa FEB*, 53(9), 1689–1699.
- Febrina, C., et al. (2023). Menjaga kesehatan mata di era digitalisasi. *Empowerment Society Journal*, 4(1), 29–36.
- Firdaus, D. H., Imran, B., Bakti, L. D., & Suryadi, E. (2022). Klasifikasi penyakit katarak pada mata menggunakan metode convolutional neural network (CNN) berbasis web. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 1(3), 18–26.
- Gour, N., Tanveer, M., & Khanna, P. (2023). Challenges for ocular disease identification in the era of artificial intelligence. *Neural Computing and Applications*, 35(31), 22887–22909. <https://doi.org/10.1007/s00521-021-06770-5>
- Hakim, D. (2019). *Optical music recognition pada citra notasi musik menggunakan convolutional neural network*. (Data institusi/penerbit tidak dicantumkan.)
- Handono, S. F., Anggraeny, F. T., & Rahmat, B. (2020). Implementasi convolutional neural network (CNN) untuk deteksi retinopati diabetik. *Jurnal Informatika dan Sistem Informasi*, 1(1), 669–678.
- Hidayat, W., Ardiansyah, M., & Setyanto, A. (2021). Pengaruh algoritma ADASYN dan SMOTE terhadap performa support vector machine pada ketidakseimbangan dataset Airbnb. *Edumatic: Jurnal Pendidikan Informatika*, 5(1), 11–20. <https://doi.org/10.29408/edumatic.v5i1.3125>
- Hutabri, R. W., Magdalena, R., & Fu'adah, Y. N. (2018). Perancangan sistem deteksi katarak menggunakan metode principal component analysis (PCA) dan K-nearest neighbor (K-NN): Analisis performansi sistem deteksi katarak. Dalam *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri 2018* (hlm. 321–327). <https://ejournal.itn.ac.id/index.php/seniati/article/view/1390>
- Indraswari, R., Herulambang, W., & Rokhana, R. (2022). Deteksi penyakit mata pada citra fundus menggunakan convolutional neural network (CNN). *Techno.Com*, 21(2), 378–389. <https://doi.org/10.33633/tc.v21i2.6162>
- Jansen, D., Handhayani, T., & Hendryli, J. (2023). Penerapan metode long short-term memory dalam memprediksi data meteorologi di Kalimantan Timur. *Simtek: Jurnal Sistem Informasi dan Teknik Komputer*, 8(2), 348–352. <https://doi.org/10.51876/simtek.v8i2.202>
- Juniati, D., & Suwanda, A. E. (2022). Klasifikasi penyakit mata berdasarkan citra fundus retina menggunakan dimensi fraktal box counting dan fuzzy K-means. *Proximal: Jurnal Penelitian Matematika dan Pendidikan Matematika*, 5(1), 10–18. <https://doi.org/10.30605/proximal.v5i1.1623>
- Kaur, S., Larsen, H., & Nattis, A. (2019). Primary care approach to eye conditions. *Osteopathic Family Physician*, 11(2), 28–34.
- Kementerian Kesehatan Republik Indonesia. (2014). *Infodatin: Situasi gangguan penglihatan dan kebutaan (Pusat Data dan Informasi Kementerian Kesehatan RI)*. Kementerian Kesehatan RI.
- Kholifatullah, B. A. H., & Prihanto, A. (2023). Penerapan metode long short term memory untuk klasifikasi pada hate speech. *Jurnal Informatics and Computer Science*. <https://doi.org/10.26740/jinacs.v4n03.p292-297>
- Marcella, D., Yohannes, Y., & Devella, S. (2022). Klasifikasi penyakit mata menggunakan convolutional neural network dengan arsitektur VGG-19. *Jurnal Algoritme*, 3(1), 60–70. [c](https://doi.org/10.33604/jati.v7i2.6927)
- Muhamad Afnan, Z., Chrisnanto, Y. H., & Pudjiantoro, T. H. (2023). Optimasi parameter artificial neural network menggunakan algoritma genetika pada peramalan token metaverse Sandbox. *Jurnal Locus Penelitian dan Pengabdian*, 2(10), 1020–1026. <https://doi.org/10.58344/locus.v2i10.1733>
- Muhlashin, M. N. I., & Stefanie, A. (2023). Klasifikasi penyakit mata berdasarkan citra fundus menggunakan YOLO V8. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(2), 1363–1368. <https://doi.org/10.36040/jati.v7i2.6927>
- Mustamin, N. F., Sari, Y., & Khatimi, H. (2021). Klasifikasi kualitas kayu kelapa menggunakan arsitektur CNN. *KLIK: Kumpulan Jurnal Ilmu Komputer*, 8(1), 49. <https://doi.org/10.20527/klik.v8i1.370>
- Mutawalli, L., Zaen, M. T. A., & Yuliadi, Y. (2023). Komparasi CNN dengan ResNet untuk klasifikasi paling akurat tingkat keganasan diabetes berdasarkan citra retinopathy. *Journal of*

- Computer System Informatics*, 4(3), 522–529.
<https://doi.org/10.47065/josyc.v4i3.3248>
- Oktavia, F., Andreswari, D., & Susilo, B. (2022). Segmentasi citra Diaretdb1 pada area hemorrhages diabetic retinopathy menggunakan metode region growing. *Rekursif: Jurnal Informatika*, 10(1), 1–11.
<https://doi.org/10.33369/rekursif.v10i1.20669>
- Pane, B. S. (2015). Peranan olahraga dalam meningkatkan kesehatan. *Jurnal Pengabdian Kepada Masyarakat*, 21(79), 1–4.
<https://jurnal.unimed.ac.id/2012/index.php/jpkm/article/view/4646>
- Papeo, D. R. P., et al. (2023). Pola terapi pengobatan glaukoma di RSUD Hasri Ainun Habibie Kabupaten Gorontalo. *Journal of Pharmaceutical Science*, 6(4), 1508–1514.
<https://doi.org/10.36490/journal-jps.com.v6i4.191>
- Pertiwi, A. A., & Utomo, A. N. (2023). Implementation of support vector machine (SVM) method on fundus image of retina of the eye for the detection of glaucoma. *Jurnal Rekayasa Informatika*, 12(1), 19–27.
- Rahardja, U. (2022). Masalah etis dalam penerapan sistem kecerdasan buatan. *Technomedia Journal*, 7(2), 181–188.
<https://doi.org/10.33050/tmj.v7i2.1895>
- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtjas, H. P. A., Yustanti, W., & Prihanto, A. (2021). Analisa learning rate dan batch size pada klasifikasi Covid menggunakan deep learning dengan optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48.
<https://doi.org/10.26740/jieet.v5n2.p44-48>
- Sa'idah, S., Suparta, I., & Suhartono, E. (2022). Modification of convolutional neural network GoogLeNet architecture with dull razor filtering for classifying skin cancer. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 11(2), 148–153.
- Salam, A., Yanto, F., Agustian, S., & Ramadhani, S. (2023). Perbandingan klasifikasi citra CT-scan kanker paru-paru menggunakan contrast stretching pada CNN dengan. *[Nama jurnal tidak dicantumkan]*, 4(3), 1341–1351.
<https://doi.org/10.30865/klik.v4i3.1448>
- Sigani, N., Masse, B. A., & Nurdin, N. (2022). Sistem pakar untuk mendiagnosa penyakit mata manusia menggunakan metode fuzzy logic. *Jurnal Elektronika Sistem Informasi dan ...*, 2(10), 1–14.
<http://www.jesik.web.id/index.php/jesik/article/view/94>
- Song, S., Chen, Y., Han, Y., Wang, F., & Su, Y. (2021). Ophthalmology consultation plan in the context of 2019-nCoV. *Frontiers in Medicine*, 8, 1–5.
<https://doi.org/10.3389/fmed.2021.618599>
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.
- Sumiati, Prasetyo, A., & Romawati. (2015). Sistem pakar diagnosa penyakit mata dengan metode forward chaining. Dalam *Seminar Nasional Riset Terapan 2015* (hlm. 15).
- Susim, T., & Darujati, C. (2021). Pengolahan citra untuk pengenalan wajah (face recognition) menggunakan OpenCV. *Jurnal Syntax Admiration*, 2(3), 534–545.
<https://doi.org/10.46799/jsa.v2i3.202>