

# Implementation of Parameter Tuning for Optimizing CNN Model Performance

Implementasi Parameter Tuning Untuk Optimalisasi Performance Model Convolutional Neural Network

Febrian G. Lalamentik, Oktavian A. Lantang, Feisy D. Kambey

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : [febrianlalamentik02@gmail.com](mailto:febrianlalamentik02@gmail.com), [oktavian\\_lantang@unsrat.ac.id](mailto:oktavian_lantang@unsrat.ac.id), [feisykambey@unsrat.ac.id](mailto:feisykambey@unsrat.ac.id)

Received: 25 July 2024; revised: 04 October 2024; accepted: 09 January 2025

**Abstract**—Parameter tuning is a crucial factor that significantly impacts the performance of Convolutional Neural Networks (CNNs) in producing accurate and reliable predictions. This study focuses on experiments comparing various settings to determine the best model by adjusting parameters. The research has implemented a structured framework with a flow diagram to guide each stage of the experiment. By tuning parameters such as Learning Rate, Epochs, Optimizer, Activation Functions, Regularization, and other variables, the aim is to find the parameter combination that maximizes model performance in making predictions. The results of the study indicate that models using optimal parameter settings can significantly enhance their performance.

**Key words**—Model; Research; Tuning; Performance.

**Abstrak**—Pengaturan penyetelan parameter merupakan faktor penting yang sangat memengaruhi performa model Convolutional Neural Network (CNN) untuk menghasilkan prediksi yang akurat dan dapat diandalkan. Penelitian ini berfokus pada eksperimen yang membandingkan setiap percobaan eksperimen yang dilakukan dengan variasi penyetelan parameter untuk menemukan model terbaik. Penelitian ini telah mengimplementasikan kerangka kerja dengan alur diagram yang terstruktur untuk memandu setiap tahap eksperimen. Dengan melakukan penyetelan pada parameter seperti Learning Rate, Epoch, Optimizer, Fungsi Aktivasi, Regularisasi, dan variabel lainnya, tujuannya adalah untuk menemukan kombinasi parameter yang dapat memaksimalkan performa model dalam melakukan prediksi. Hasil dari penelitian menunjukkan bahwa model yang menggunakan pengaturan parameter yang optimal mampu meningkatkan performa model lebih baik dan signifikan.

**Kata kunci**—Model; Penelitian; Penyetelan; Performa.

## I. PENDAHULUAN

Salah satu model yang signifikan dalam perkembangan *Deep Learning* adalah *Convolutional neural network* (CNN). CNN adalah jenis jaringan yang digunakan khususnya untuk analisis gambar. Gambar melewati serangkaian lapisan konvolusi, di mana setiap lapisan mencoba mengenali pola berdasarkan filter yang diterapkan pada gambar. Melalui penggunaan teknik konvolusi ini, CNN memiliki kapabilitas untuk mengenali fitur-fitur yang lebih abstrak dan kompleks, memungkinkan pemahaman yang lebih mendalam terhadap citra dan membuka peluang untuk berbagai aplikasi dalam bidang analisis gambar [1].

Menurut pendapat Anhar dan Putra dinyatakan bahwa kekurangan membuat model CNN disebabkan karena

terjadinya *overfitting* [2]. Kemudian penelitian dari Firmansyah menyatakan bahwa *overfitting* terjadi pada saat model dengan sangat kompleks, ini terjadi karena model memprediksi dan menghafal pola dari data sehingga tidak dengan baik mempelajari pola sebenarnya [3]. Hal ini menyebabkan model memiliki performa yang buruk untuk memprediksi pada data baru karena hanya cocok pada sampel yang digunakan pada saat pelatihan.

Dalam menghindari *overfitting* maka model harus disetel dengan baik sehingga dapat menghasilkan performa yang baik pada data baru yang akan digunakan pada tahap pengujian model. Oleh karena itu perlu dilakukannya Penyetelan Parameter atau Parameter Tuning perlu dilakukan untuk memahami bagaimana parameter tersebut mempengaruhi kinerja yang dihasilkan [4] dengan cara melakukan penyesuaian nilai-nilai parameter (hyperparameter) dalam model dan prosesnya berjalan secara iteratif. Melibatkan cobaan beberapa kombinasi hyperparameter akan memberikan solusi untuk mengurangi *overfitting* [5].

Permasalahan tersebut diangkat menjadi sebuah penelitian yang berfokus untuk melakukan penyetelan parameter yang efisien, supaya terhindar dari *overfitting* sehingga menghasilkan model CNN dengan performa yang optimal dilakukan dengan rentetan eksperimen untuk menggapai target penelitian. Riset ini menggunakan dataset besar yang fokus pada diagnosis gambar *Optical Coherence Tomography* (OCT) untuk klasifikasi penyakit pada retina dan macula (bagian dari retina) mata manusia dengan kategori tiga macam penyakit dan satu macam kondisi normal.

### A. Penelitian Terkait

Adapun penelitian terkait yang menjadi acuan dalam melakukan riset ini, dikarenakan penelitian-penelitian ini membahas mengenai *Convolutional neural network* dan *Parameter tuning*:

- 1) "Retinal Disease Classification from OCT Images Using Deep Learning Algorithms" [6]. Model ini melakukan klasifikasi terhadap dataset OCT Retina pada kaggle menggunakan 6 macam arsitektur. Data yang digunakan berjumlah 250 data pada setiap kategori untuk digunakan sebagai data pengujian, normalisasi data yang digunakan adalah FCN dengan arsitektur U-Net, model ini dibuat dalam bentuk binary classification. Akurasi yang didapat pada

- model berbeda pada tiap label yaitu antara 9.6% sampai 9.9%.
- 2) "Control *Batch size* and Learning rate to Generalize Well: Theoretical and Empirical Evidence" [7]. Pada fokus penggunaan *optimizer* SGD, diterapkan kontrol terhadap *Batch size* dan learning rate. Penelitian ini menggunakan 20 nilai *Batch size* dan learning rate yang berbeda. Hasilnya ditentukan untuk tidak boleh menggunakan *Batch size* yang terlalu besar guna dan dan learning rate yang terlalu kecil mengoptimalkan performa model dengan *optimizer* SGD.
  - 3) "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Model ResNet50" [8]. Model yang dibuat dengan arsitektur ResNet50 ini meneliti tentang perbandingan hasil yang berbeda jika dilakukan split data latih dan data validasi yang berbeda. Hasilnya perbandingan 90:10 memiliki hasil lebih baik dengan selisih 3,26% terhadap pembagian data 70:30.
  - 4) "Penerapan Model Arsitektur VGG16 Untuk Klasifikasi Jenis Sampah" [9]. Pada penelitian ini, dilakukan klasifikasi jenis sampah berdasarkan citra dengan menggunakan metode CNN dan arsitektur VGG-16, memanfaatkan dataset Garbage Classification dari situs Kaggle. Arsitektur VGG-16 yang digunakan dalam penelitian ini terdiri dari 16 lapisan, yang meliputi 13 lapisan konvolusi dan 3 lapisan fully-connected. Model dilatih menggunakan library TensorFlow dengan pembagian data sebesar 7:1.5:1.5 untuk data latih, data uji, dan data validasi. Arsitektur CNN VGG-16 sebagai metode transfer learning dimodifikasi dengan menambahkan beberapa lapisan, termasuk *pooling Layer* dengan GlobalAveragePooling2D, *dense Layer* dengan aktivasi ReLU, dan *dense Layer* dengan aktivasi softmax sebagai lapisan fully-connected. Hasil pelatihan menunjukkan nilai akurasi sebesar 82,89%.
  - 5) "Epoch Tuning Hyperparameter in Fire Image Classification at University Sjakhyakirti" [10]. Pengaruh parameter epoch dalam penelitian ini sangat penting untuk dipertimbangkan karena dalam rentang (30, 50, 75, 100, dan 120) model mendapat hasil yang cenderung tidak tentu untuk mencapai nilai yang optimal, sehingga hasil yang didapatkan ada pada epoch 100 yang memiliki nilai akurasi 0.835% hanya selisih 0.010% dari nilai akurasi epoch 30. Namun hasil utama yang dicapai diambil pada epoch 30 dengan mempertimbangkan durasi waktu pelatihan yang lebih singkat.
  - 6) "Data Balancing Methods on Radiographic Image Classification on Unbalance Dataset" [11]. Dalam catatan jurnal ini memuat metode untuk melakukan oversampling atau undersampling pada data yang tidak seimbang. Model ini menggunakan arsitektur MobileNetV2 dan ResNet50 untuk diuji, hasil akurasi terbaiknya adalah 95.58%.
  - 7) "Dropout vs. batch normalization: an empirical study of their impact to Deep Learning" [12]. Penelitian ini menyetel parameter dengan nilai variatif pada epoch dan learning rate. *Batch size* bernilai 128, *Optimizer* yang digunakan RMSProp dan SGD, Activation Function yang digunakan ReLU, dan diberdayakan dropout serta batch normalization untuk mencegah *overfitting*. Hasil akurasi terbaik yang didapat dalam penelitian ini adalah 0.84% sebagai hasil terbaik setelah dilakukannya perbandingan dengan hasil eksperimen model lain yang sudah dilakukan.
  - 8) "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional neural network (CNN)" [13]. Penelitian ini melakukan klasifikasi terhadap objek dalam citra. Objek-objek dalam citra ini dikelompokkan ke dalam tiga kategori dengan masing masing kategori memiliki 100 data gambar. Model ini menggunakan confusion matrix untuk mengevaluasi performa model dan hasil akurasi terbaik pada data baru adalah 88.89%.
  - 9) "Penerapan Tuning Hyperparameter Randomsearchcv pada Adaptive Boosting Untuk Prediksi Kelangsungan Hidup Pasien Gagal Jantung" [14]. Dalam penelitian ini, dilakukan penyetelan parameter-parameter seperti Learning rate diterapkan dengan mengatur cross-validation untuk memilih nilai learning rate yang baik antara 0.1, 0.01, dan 1.0. Model yang dihasilkan mendapatkan skor akurasi terbaik yaitu 0,897 dengan penetapan learning rate 0.01.

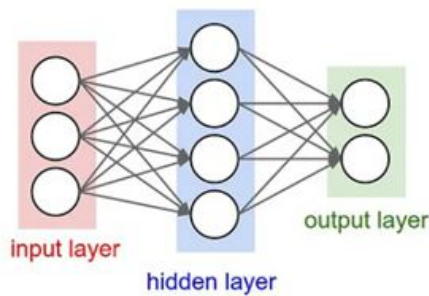
## B. Artificial Intelligence

Artificial Intelligence (AI) adalah kemampuan komprehensif mesin untuk "berpikir" atau meniru perilaku manusia, serta menganalisis data secara mandiri tanpa dipengaruhi oleh faktor manusia [15]. AI mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti membuat keputusan, menyelesaikan masalah, dan belajar dari pengalaman. Semua ini dilakukan dengan tingkat kemandirian yang tinggi tanpa banyak campur tangan manusia. Oleh karena itu, AI memungkinkan mesin untuk memproses informasi, mengenali pola, dan membuat prediksi dengan efisiensi dan akurasi yang semakin meningkat.

Machine Learning (ML), sebagai cabang dari AI, biasanya berfokus pada tugas-tugas pembelajaran yang diawasi atau tidak diawasi, di mana hasilnya didasarkan pada masukan dan diproses melalui algoritma pengenalan pola yang berulang. Pembelajaran mesin terus mengubah cara pengambilan keputusan dalam kehidupan sehari-hari. Dalam bidang layanan kesehatan, digitalisasi telah memungkinkan pengumpulan dan analisis berbagai data klinis, seperti catatan medis, grafik kesehatan, gambar diagnostik, dan hasil tes laboratorium, serta data nonklinis, seperti informasi sensus dan studi populasi atau komunitas [15].

## C. Deep Learning

*Deep Learning* berfokus pada pemecahan masalah dengan dataset yang besar, dimana teknik ini menggunakan jaringan saraf tiruan. DL merupakan bagian penting dari pendekatan Supervised Learning dalam memecahkan permasalahan yang kompleks [16].

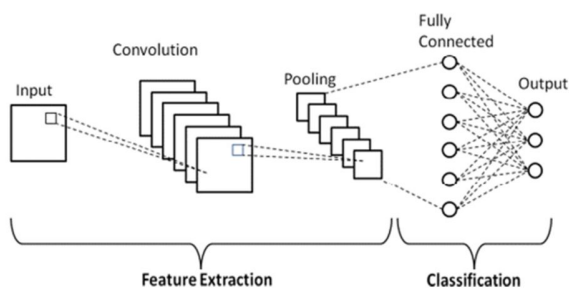


Gambar 1. Ilustrasi Lapisan Neuron dalam Deep Learning

#### D. Convolutional Neural Network

Jaringan saraf konvolusional (CNN) bekerja dengan menggunakan kernel atau filter konvolusi yang bergerak di sepanjang matriks gambar untuk mengekstraksi fitur-fitur tingkat tinggi.

Struktur CNN umumnya terdiri dari ekstraktor fitur yang mencakup beberapa lapisan konvolusi, lapisan pooling, lapisan fully connected, dan diakhiri dengan pengklasifikasi softmax. Lapisan konvolusi berfungsi mengekstraksi fitur penting dari data input seperti tepi, tekstur, dan pola menggunakan filter atau kernel yang menghasilkan peta fitur. Setelah itu, lapisan pooling digunakan untuk mengurangi dimensi data, baik melalui *max pooling* atau *average pooling*, guna menghemat waktu komputasi dan mengurangi kompleksitas model, serta membantu dalam mengurangi *overfitting*. Hasil ekstraksi fitur ini kemudian diteruskan ke lapisan fully connected yang menghubungkan setiap neuron ke semua neuron di lapisan berikutnya, memungkinkan integrasi informasi dari seluruh peta fitur. Akhirnya, lapisan softmax mengubah hasil tersebut menjadi probabilitas untuk setiap kelas yang mungkin, memungkinkan model untuk melakukan pengklasifikasian akhir dengan akurasi tinggi. Dengan kombinasi komponen-komponen ini, CNN mampu menggabungkan kemampuan ekstraksi fitur lokal dengan integrasi informasi global untuk membuat prediksi yang efisien dan akurat [17].



Gambar 2. Arsitektur CNN

#### E. Tensorflow

Tensorflow merupakan framework dan library Python yang dikembangkan oleh Google. Operator-operator TensorFlow

yang mendukung GPU (Graphics Processing Unit) memberikan keunggulan yang besar dibandingkan dengan penggunaan CPU dalam pelatihan model.

#### F. Parameter Tuning

*Parameter Tuning* adalah proses penyesuaian nilai-nilai parameter dalam model CNN untuk mencapai nilai akurasi yang optimal dengan kinerja yang lebih efisien. Dalam *parameter tuning* ada istilah *Hyperparameter* yang berarti nilai-nilai yang ditetapkan sebelum proses pembelajaran dimulai. Untuk meningkatkan hasil pengelompokan, penting untuk memilih parameter yang cocok dalam model yang diajukan. *Hyperparameter* digunakan untuk mengendalikan berbagai aspek pembelajaran mesin yang berpengaruh besar terhadap kinerja dan bentuk akhir model yang dibuat [14]. Sederhananya *hyperparameter* adalah nilai-nilai yang akan dipertimbangkan terlebih dahulu untuk menjadi input parameter pada proses tuning (penyetelan), sehingga parameternya dapat memberikan hasil yang optimal dalam pelatihan model CNN. *Parameter Tuning* akan meliputi:

##### 1) Regularization

Regularisasi adalah teknik untuk menghindari *overfitting* dengan menambahkan komponen tambahan ke dalam fungsi objektif yang harus dioptimalkan selama pelatihan [18]. Penggunaan *dropout* dan *batch normalization* adalah salah satu cara melakukan regularisasi.

##### 2) Activation Function

Fungsi aktivasi seperti ReLU, *Sigmoid*, atau *Tanh* dapat mempengaruhi cara model belajar. Pilihan fungsi aktivasi yang tepat dapat mempercepat konvergensi dan mengatasi masalah *vanishing gradient* [19]. Fungsi aktivasi dapat digunakan sebagai *hyperparameter* untuk mencegah *overfitting*. Fungsi aktivasi yang akan digunakan untuk masalah model CNN penelitian adalah berikut.

##### a) ReLU

ReLU adalah fungsi aktivasi yang efisien, mirip dengan sigmoid karena bersifat non-linear, namun lebih sering dipakai dalam banyak kasus karena mampu untuk menghindari *vanishing gradient*. ReLU hanya boleh digunakan pada lapisan tersembunyi yang berada di antara lapisan *input* dan lapisan *output*. ReLU akan meamtikan *input* yang bernilai negatif dan, sedangkan nilai positif akan dikembalikan sesuai dengan nilai *inputnya*.

##### b) Softmax

Softmax adalah fungsi aktivasi yang akan mengubah input menjadi probabilitas yang memiliki jumlah seluruh probabilitasnya adalah 1, ini digunakan untuk memprediksi masalah multi-kelas. Setiap vektor input akan diubah menjadi bentuk eksponensialnya, kemudian setiap nilai eksponensialnya dijumlahkan, selanjutnya setiap nilai eksponensialnya dibagi dengan jumlah seluruh eksponensial, hasil total dari semua vektor output akan bernilai 1.

### c) Smoothswish

Fungsi aktivasi 'smoothswish' menggabungkan elemen linier dan sigmoid untuk meningkatkan kinerja model jaringan saraf. Fungsi ini dihitung dengan mengalikan input 'x' dengan nilai dari fungsi sigmoid yang diterapkan pada (beta dikali x), dengan 'beta' mengatur tingkat kekeruhan sigmoid. Hasilnya adalah fungsi yang halus dan kontinu, yang mendukung gradien yang lebih stabil dan membantu mengatasi masalah gradien yang terlalu kecil selama pelatihan [20].

### 3) Optimization Algorithm

Algoritma optimisasi seperti Stochastic Gradient Descent (SGD), Adam, atau RMSprop memiliki parameter-parameter khusus seperti momentum atau learning rate yang dapat diatur untuk mengoptimalkan kinerja selama pelatihan [21]. Pemilihan Algoritma optimisasi sangat penting dan bergantung pada situasi.

#### a) Adam (Adaptive Moment Estimation)

Teknik optimisasi Adam digunakan karena meningkatkan efisiensi komputasi model dan untuk mengoptimalkannya dengan parameter serta data yang terbilang besar [17]. Adam mampu menyesuaikan laju pembelajaran didasarkan pada seberapa besar dan seberapa sering perubahan gradien terjadi. Sangat bagus untuk data dengan gradien yang tidak stabil.

#### b) SGD (Stochastic Gradient Descent)

Optimisasi SGD bersifat sederhana, fokus untuk gradien curam yang dipatok dari fungsi biaya sehingga ditemukan nilai minimumnya. SGD umumnya memberikan hasil yang baik secara praktis dalam berbagai aplikasi [18]. Optimisasi ini sangat sesuai dengan gradien yang condong konsisten.

#### c) RMSProp (Root Mean Square Propagation)

RMSprop mengurangi fluktuasi laju pembelajaran dan memfasilitasi konvergensi yang lebih cepat menuju solusi terbaik. Optimisasi ini cocok untuk data gradien bervariasi karena dapat menghitung rata-rata kuadrat gradien untuk setiap parameter, ini memungkinkan peningkatan laju pembelajar model menyesuaikan dengan perubahan gradien.

### 4) Learning rate

Learning rate adalah parameter yang mengontrol seberapa besar langkah yang diambil oleh algoritma optimisasi selama proses pelatihan. Learning rate yang optimal memastikan bahwa model konvergen dengan cepat ke minimum lokal yang baik tanpa melewatkannya atau terjebak di dalamnya [22].

### 5) Epoch

Epoch adalah jumlah kali seluruh dataset digunakan untuk melatih model. Jumlah epoch yang tepat penting untuk memungkinkan model mempelajari pola yang ada dalam data. Terlalu banyak epoch bisa menyebabkan *overfitting* [23].

### 6) Batch size

*Batch size* adalah jumlah sampel yang digunakan untuk menghitung gradien dalam satu iterasi pelatihan. *Batch size* yang lebih besar dapat mempercepat pelatihan karena update parameter dilakukan lebih jarang, sementara *Batch size* yang lebih kecil dapat membantu mencegah *overfitting* dan memungkinkan update yang lebih sering [22].

### 7) Callbacks

Komponen ini memuat fungsi yang dapat mencegah *overfitting* atau membantu peningkatan optimalitas kinerja model, dua fungsi diantaranya adalah Early stopping dan Model checkpoint.

### 8) Fine Tuning

Penggunaan model yang telah dilatih sebelumnya sebagai inisialisasi, kemudian melatih ulang model tersebut untuk tugas yang berbeda, dikenal sebagai *Fine-tuning*. Proses ini terbukti secara signifikan meningkatkan kemampuan model dalam beradaptasi dengan tugas baru. *Fine-tuning* memungkinkan model untuk memanfaatkan pengetahuan yang diperoleh dari pelatihan sebelumnya dan menyesuaikannya dengan data spesifik dari tugas baru. Dengan demikian, model tidak hanya lebih cepat beradaptasi tetapi juga cenderung mencapai kinerja yang lebih baik pada tugas baru dibandingkan jika dilatih dari awal [24].

## G. Arsitektur Model

Arsitektur model CNN membantu untuk memproses fitur dan pola pada gambar untuk diidentifikasi. Pada penelitian ini akan digunakan pre-trained model dan akan dilakukan perbandingan efektivitas arsitektur model tersebut pada *input* data penelitian [25].

### 1) Residual Network

Residual Network, atau ResNet adalah arsitektur jaringan saraf yang dirancang untuk mengatasi dua masalah utama yang muncul saat melatih jaringan saraf yang sangat dalam: masalah gradien yang menghilang dan masalah degradasi. Cara kerja ResNet menggunakan blok residual, di mana input awal ditambahkan ke output dari lapisan-lapisan konvolusional setelah melalui fungsi aktivasi, yang umumnya adalah ReLU (Rectified Linear Unit). Penambahan ini membantu menjaga gradien tetap ada, sehingga memungkinkan pelatihan jaringan yang sangat dalam tanpa kehilangan informasi. ResNet50, yang memiliki 50 lapisan, dan ResNet101, yang memiliki 101 lapisan, adalah contoh dari arsitektur ini.

### 2) VGG

Model VGG dikenal sebagai arsitektur Deep Learning yang sangat umum dalam teknik visi komputer. Model ini merupakan jaringan saraf konvolusi (CNN) yang dirancang untuk memproses dan mengklasifikasikan [26]. Nama VGG diambil dari nama grup riset yang membuatnya yaitu Visual Geometry Group. VGG16 memiliki 16 lapisan, sedangkan VGG19 memiliki 19 lapisan merupakan dua macam Layer arsitektur

VGG. Tujuan dari arsitektur ini adalah untuk memperbaiki akurasi klasifikasi gambar dengan menambah kedalaman jaringan, yang memungkinkan model untuk belajar representasi fitur yang lebih kompleks dan detail dari data visual. Model ini menggunakan lapisan konvolusi yang diikuti oleh lapisan pooling untuk mengurangi dimensi data, dan akhirnya diakhiri dengan beberapa lapisan fully connected untuk klasifikasi.

#### H. Metrik Evaluasi Performa

Dalam melakukan evaluasi dibutuhkan metrik metrik evaluasi bertujuan untuk mengukur performa model sebagai bagian dari tahapan untuk menggapai hasil yang optimal [10].

##### 1) Confusion matrix

Confusion matrix mengukur akurasi prediksi model dibandingkan dengan label yang actual untuk dapat mengambil poin evaluasi performa model. Alat ini menyajikan tabel yang memperlihatkan jumlah prediksi yang benar dan salah di setiap kelas, dengan memperhitungkan true positives (TP), false positives (FP), true negatives (TN), dan false negatives (FN). *Confusion matrix* memungkinkan analisis yang mendalam mengenai bagaimana model berperilaku di setiap label, serta mengidentifikasi area di mana model mungkin membuat kesalahan, sehingga membantu dalam perbaikan dan penyempurnaan model lebih lanjut.

##### 2) Accuracy

Akurasi menilai seberapa baik model dalam membuat prediksi yang benar di seluruh data yang digunakan. Metrik ini menghitung persentase total prediksi yang benar dibandingkan dengan keseluruhan jumlah data. Setiap data dianggap memiliki bobot yang sama, sehingga semua unit data memberikan kontribusi yang setara terhadap perhitungan akurasi. Dengan kata lain, akurasi memberikan gambaran umum tentang seberapa efektif model dalam memprediksi hasil yang benar di seluruh dataset tanpa memperhitungkan perbedaan antara kelas-kelas yang mungkin ada.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

##### 3) Presicion

Presisi mengungkapkan proporsi unit yang dikatakan model kita sebagai Positif dan benar-benar Positif. Dengan kata lain, presisi memberi tahu kita seberapa banyak kita bisa mempercayai model ketika ia memprediksi data sebagai Positif.

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

##### 4) Recall

Recall menilai seberapa baik model dalam mendeteksi semua unit Positif dalam dataset. Dengan kata lain, Recall menunjukkan kemampuan model untuk mengidentifikasi seluruh unit Positif yang ada.

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

##### 5) F1-Score

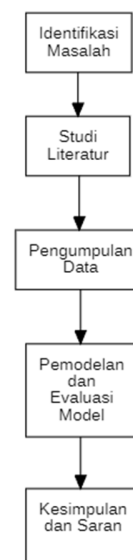
F1-Score adalah metrik yang menggabungkan Presisi dan Recall menjadi satu nilai, dengan nilai terbaik F1-Score mencapai 1 dan nilai terburuknya 0. F1-Score memberikan kontribusi yang sama terhadap presisi dan Recall, menggunakan rata-rata harmonik untuk memastikan keseimbangan yang baik antara kedua metrik ini. Ini berarti F1-Score sangat berguna ketika kita ingin mengevaluasi model dalam situasi di mana baik presisi maupun Recall penting dan harus dipertimbangkan secara bersamaan. F1-Score membantu menemukan kompromi yang optimal antara menangkap sebanyak mungkin unit Positif (Recall) dan memastikan bahwa prediksi Positif model benar-benar akurat (Presisi).

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

## II. METODE

### A. Eksperimen

Penelitian ini menggunakan kerangka pikir penelitian sebagai panduan logis alur penelitian. Dalam bagian ini dicantumkan tambahan flowchart untuk membuat model CNN untuk digunakan dalam riset.



Gambar 3. Kerangka Pikir Penelitian

##### 1) Identifikasi Masalah

Tahapan pertama dilakukannya pengidentifikasian masalah terhadap optimalisasi performa model CNN dan penyelesaiannya adalah dengan mencegah *overfitting* menggunakan cara penyetelan parameter. Hasil pemikiran mendasar ini dituangkan menjadi rumusan masalah.

## 2) Studi Literatur

Tahap ini dilakukan dengan cara melakukan pembelajaran mendalam pada literatur jurnal dan makalah terkait cakup permasalahan pada penelitian ini. Setiap manfaat dan batasan kemudian dituangkan untuk mengatur jangkauan target yang akan dicapai serta mengumpulkan setiap teori yang akan diterapkan guna menyelesaikan masalah penelitian untuk melakukan penyetelan parameter.

## 3) Pengumpulan Data

Input data harus dikumpulkan dengan mempertimbangkan sumber data yang legal dan dapat dipertanggungjawabkan. Penggunaan data input penelitian menggunakan dataset yang tersedia secara publik di website Kaggle.

## 4) Pemodelan dan Evaluasi Model

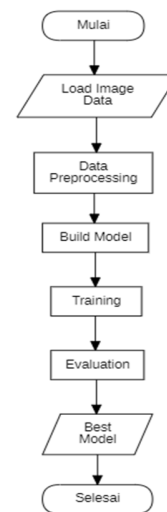
Tahap ini merupakan tahap utama yang mana penelitian akan berfokus untuk membangun model CNN yang optimal melalui penyetelan parameter dengan memperhatikan alur penelitian. Dimulai dari tahap load data gambar, data preprocessing, melakukan modelling, pelatihan, evaluasi, dan menghasilkan satu output model dengan performa terbaik dari percobaan-percobaan eksperimen yang dikonfigurasi.

## 5) Kesimpulan dan Saran

Penelitian akan menjelaskan kesimpulan-kesimpulan dari riset yang dilakukan dan apakah penelitian tersebut dapat menyelesaikan masalah yang telah dirumuskan dan menggapai tujuan yang telah ditargetkan. Kemudian diberikan juga oleh penulis, saran-saran yang akan berguna untuk dipertimbangkan untuk penelitian lanjutan maupun penelitian terkait yang akan memakai hasil yang dicapai sebagai referensi.

## B. Alur Penelitian

Alur Penelitian ini akan dilakukan secara bertahap dengan berfokus melaksanakan setiap langkah penelitian. Metode ini akan dilakukan dengan memuat data gambar, mempersiapkannya, membangun model, melakukan pelatihan, dan mengevaluasi setiap eksperimen yang dilakukan untuk menemukan hasil akhir model terbaik seperti pada gambar.

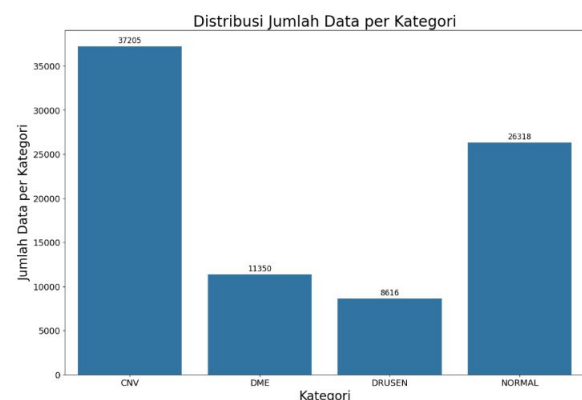


Gambar 4. Flowchart membuat model Convolutional Neural Network

## 1) Load Data Image

Dataset yang digunakan berupa dataset gambar untuk membuat Dataset yang digunakan untuk membuat model CNN menggunakan dataset gambar. Data yang diperoleh ini merupakan dataset publik yang tersedia dari website Kaggle. Data yang terdiri dari empat kategori tiga di antaranya adalah macam macam penyakit dan satu di antaranya adalah kondisi normal. Data gambar yang akan digunakan merupakan dataset gambar “Retinal OCT Images (Optical Coherence Tomography)” yang diambil dari website Kaggle yang menyediakan dataset secara publik untuk diakses dan digunakan.

Dataset gambar ini memuat data gambar sebanyak 83.489 yang sudah dikelompokkan dalam satu folder train\_val\_test, setiap kategori yang sudah ditentukan. Spesifikasi data gambar tersebut ada pada table berikut.



Gambar 5. Distribusi Dataset

Dataset yang digunakan memiliki empat label kondisi retina. CNV (Choroidal Neovascularization), terjadi karena tumbuhnya pembuluh darah tidak normal di bawah retina. DME (Diabetic Macular Edema), adanya pembengkakan di macula tengah retina disebabkan kebocoran cairan pembuluh darah yang rusak. DRUSEN, Titik-titik kecil yang berwarna kuning

terbentuk di bawah retina. NORMAL, Tidak ada penyakit. Berikut ciri gambar setiap kondisi tersebut.



Gambar 6. Ciri-ciri label Dataset OCT Retina

## 2) Data Preprocessing

Pembagian data pelatihan, validasi, dan pengujian pada dataset ini akan menggunakan library *scikit-learn*. Bobotnya akan ditentukan untuk dosis data train yang lebih besar dan untuk data validasi serta data pengujian akan disesuaikan.

### a) Split Data

Pembagian data pelatihan, validasi, dan pengujian pada dataset ini akan menggunakan library *scikit-learn*. Bobotnya akan ditentukan untuk dosis data train yang lebih besar dan untuk data validasi serta data pengujian akan disesuaikan.

### b) Sampling

Sampling dilakukan untuk menanggulangi masalah ketidakseimbangan data latih yang pada dataset, agar model yang dikembangkan tidak akan bias pada label dengan data yang lebih besar. Pada penelitian ini penggunaan metode dilakukan dengan dua cara sampling yaitu *oversampling* dan *undersampling*. *Undersampling* digunakan untuk mengabaikan beberapa data pada label yang memiliki data yang lebih banyak sedangkan *oversampling* dipakai untuk menduplikasi beberapa data pada label yang memiliki data lebih sedikit. Kedua cara sampling tersebut akan dibandingkan mutu dengan cara tanpa penggunaan metode *sampling*.

### c) Augmentasi

Proses ini diperlukan untuk dapat membuat data gambar lebih beragam untuk dapat dilatih sehingga tidak perlu mengumpulkan tambahan data yang baru. Ini bertujuan memodifikasi data latih agar lebih variatif agar kinerja model lebih beradaptasi terhadap perubahan data. Kemudian setelah proses augmentasi, akan dilakukan *shuffle* pada data latih dan data validasi yang selanjutnya menjadi input ke data generator sebagai tahapan akhir persiapan data.

## 3) Build Model

Pada tahap membangun model, setiap parameter yang difokuskan untuk mencegah *overfitting* akan disetel. Nilai-nilai *hyperparameter* akan ditentukan pada proses ini kemudian setelah evaluasi model sudah diamati, maka proses tuning akan dilakukan kembali secara variatif untuk dilakukan pelatihan ulang guna mencapai konfigurasi terbaik pada model. Berikut pembagian tahapan dalam membangun model.

### a) Arsitektur Model

Dalam membuat model CNN, arsitektur model akan menggunakan pre-trained model dan arsitektur yang dibangun dari *scratch* yang akan dikostumisasi dengan menyeting parameter-parameter untuk dapat cocok mempelajari data

gambar yang akan dilatih. Perbandingan Arsitektur CNN akan diterapkan supaya ditemukan arsitektur yang optimal untuk model klasifikasi gambar penelitian.

### b) Compile Model

Tahapan kompilasi model akan berfokus pada inisialisasi *hyperparameter* yang akan ditentukan nilainya ke dalam model yang akan dilatih. Pada tahap ini juga akan dipersiapkan setiap parameter lain yang dipakai untuk menjalankan proses pelatihan model CNN.

## 4) Training

Proses pelatihan adalah langkah berikut di mana model diajar untuk mengenali pola, mengambil keputusan, atau melakukan tugas tertentu dengan memanfaatkan data yang ada. Pelatihan akan terus diiterasi jika hasil nilai metrik akurasi dan *loss* yang dievaluasi dan dikomparasi belum memenuhi standar dimana masih mendapat hasil yang tidak optimal. Model akan telah dilatih akan mencatat setiap nilai akurasi, nilai validasi akurasi, nilai *loss*, dan nilai validasi *loss* pada masa pelatihan.

## 5) Evaluation

Evaluasi akan dilakukan dengan mempertimbangkan skenario pelatihan yang akan dilakukan, kemudian hasilnya akan dijabarkan sesuai kelompok skenario yang diatur. Tahapan ini membuka jalan untuk melakukan pengamatan terhadap model yang telah dilatih sehingga bisa dipilih satu dengan mutu terbaik.

### a) Skenario

Pada tahap ini akan daftar skenario akan dijelaskan beserta hasil terbaik dari setiap kelompok skenario yang telah dicapai. Kemudian proses berikutnya adalah melakukan evaluasi untuk melihat rangkuman penilaian kinerja model terhadap data baru yang akan diuji pada model-model tersebut. Rangkuman pengamatan tersebut akan menentukan keseluruhan hasil evaluasi ini untuk menemukan model dengan performa terbaik.

### b) Hasil performa

Pada tahap ini akan dicantumkan setiap akhir catatan akurasi dan *loss* serta validasinya masing-masing untuk disimpulkan. Fungsi kalkulasi akan digunakan untuk menemukan nilai hasil pengukuran Presisi, *Recall*, *F1-Score*, dan Akurasi. Model dengan konfigurasi terbaik akan dipilih dari setiap eksperimen model yang telah dievaluasi.

## 6) Best Model

Tahapan terakhir dari alur penelitian ini adalah pembahasan hasil performa model terbaik dalam mengoptimalkan performa dan pengaruhnya dalam penelitian.

## III. HASIL DAN PEMBAHASAN

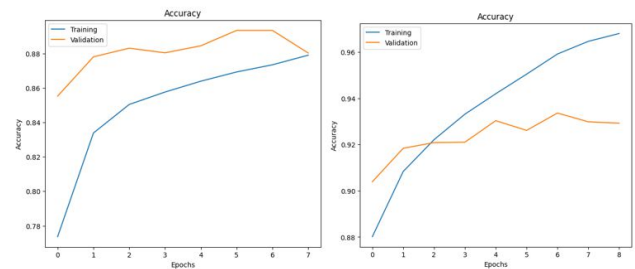
Tahap ini dilakukan dengan mempertimbangkan skenario pelatihan yang akan dilakukan. Tahapan ini membuka jalan untuk melakukan pengamatan terhadap model yang telah dilatih sehingga bisa dipilih satu dengan mutu terbaik.

TABEL I  
SKENARIO PENELITIAN

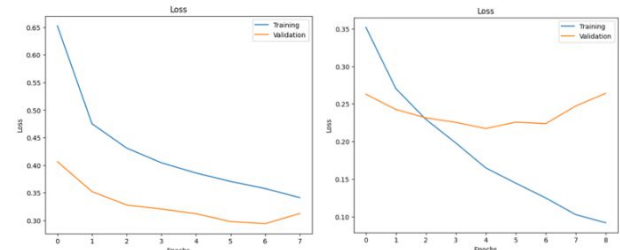
| Experimen                                  | Skenario                                                                                                                                                               |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Skenario Split Data                        |                                                                                                                                                                        |
| Experimen 1                                | Split data 70%:20%:10%                                                                                                                                                 |
| Experimen 2                                | Split data 80%:10%:10%                                                                                                                                                 |
| Experimen 3                                | Split data 70%:30%                                                                                                                                                     |
| Skenario Sampling                          |                                                                                                                                                                        |
| Experimen 4                                | Tanpa Sampling                                                                                                                                                         |
| Experimen 5                                | Oversampling                                                                                                                                                           |
| Experimen 6                                | Undersampling                                                                                                                                                          |
| Skenario Augmentasi dan Rescale            |                                                                                                                                                                        |
| Experimen 7                                | Tanpa Augmentasi Data dan Rescale                                                                                                                                      |
| Experimen 8                                | Rescale = 1./255                                                                                                                                                       |
| Experimen 9                                | Height_shift_range dan height_shift_range = 0.2 dan fill_mode = 'nearest'                                                                                              |
| Skenario Arsitektur Model                  |                                                                                                                                                                        |
| Experimen 10                               | ResNet50 dengan fungsi aktivasi relu (pre-trained)                                                                                                                     |
| Experimen 11                               | ResNet101 dengan fungsi aktivasi relu (pre-trained)                                                                                                                    |
| Experimen 12                               | VGG16 dengan fungsi aktivasi relu (pre-trained)                                                                                                                        |
| Experimen 13                               | VGG19 dengan fungsi aktivasi relu (pre-trained)                                                                                                                        |
| Experimen 14                               | ResNet50 dengan tambahan fungsi aktivasi relu dan smoothswish (scratch)                                                                                                |
| Skenario Kostumisasi Model                 |                                                                                                                                                                        |
| Experimen 15                               | Dense 64 dengan aktivasi 'relu', dropout 0.5                                                                                                                           |
| Experimen 16                               | GlobalAveragePooling2D, dropout 0.5, Dense 64 dengan aktivasi 'relu', dropout 0.5                                                                                      |
| Experimen 17                               | GlobalAveragePooling2D, Dense 128 dengan aktivasi 'relu', dropout 0.5, Dense 64 dengan aktivasi 'relu', dropout 0.5                                                    |
| Experimen 18                               | GlobalAveragePooling2D, Batch Normalization, Dense 128 dengan aktivasi 'relu', dropout 0.5, Dense 64 dengan aktivasi 'relu', dropout 0.5                               |
| Experimen 19                               | GlobalAveragePooling2D, Batch Normalization, Dense 128 dengan aktivasi 'relu', dropout 0.5, Dense 64 dengan aktivasi 'relu', dropout 0.5, dan Fine-tuning unfreeze -10 |
| Experimen 20                               | GlobalAveragePooling2D, Batch Normalization, Dense 128 dengan aktivasi 'relu', dropout 0.5, Dense 64 dengan aktivasi 'relu', dropout 0.5, dan Fine-tuning unfreeze -4  |
| Skenario Epoch setelah Fine-tuning         |                                                                                                                                                                        |
| Experimen 21                               | 5 Epoch pada pelatihan kedua                                                                                                                                           |
| Experimen 22                               | 10 Epoch pada pelatihan kedua                                                                                                                                          |
| Skenario Optimizer                         |                                                                                                                                                                        |
| Experimen 23                               | Adam                                                                                                                                                                   |
| Experimen 24                               | SGD                                                                                                                                                                    |
| Experimen 25                               | RMSProp                                                                                                                                                                |
| Skenario Learning Rate setelah Fine-tuning |                                                                                                                                                                        |
| Experimen 26                               | Laju pembelajaran 0.001 pada pelatihan kedua                                                                                                                           |
| Experimen 27                               | Laju pembelajaran 0.0001 pada pelatihan kedua                                                                                                                          |
| Experimen 28                               | Laju pembelajaran 0.00001 pada pelatihan kedua                                                                                                                         |

Dari semua hasil eksperimen bertahap, ditemukan model terbaik pada eksperimen ke-27. Dalam rangkaian penelitian ini, Eksperimen 27 telah dipilih sebagai model dengan performa terbaik. Model ini menggunakan pembagian set pelatihan dengan 70% dataset untuk pelatihan, 20% dataset untuk validasi, dan 10% dataset untuk pengujian, tanpa melakukan augmentasi atau sampling pada data latih. Arsitektur model mengandalkan ResNet50 yang sudah dilatih sebelumnya (pre-trained) dan ditingkatkan dengan menambahkan dua layer dense, masing-masing dengan 128 dan 64 neuron yang menggunakan fungsi aktivasi ReLU. Dropout sebesar 0.5

diterapkan pada kedua layer dense tambahan untuk mengurangi risiko *overfitting*. GlobalAveragePooling2D digunakan untuk merata-ratakan seluruh fitur map, dan diintegrasikan dengan *Batch Normalization* untuk stabilitas pelatihan yang lebih baik. Pada tahap akhir ini Model telah dilatih kembali untuk mengamati kekonsistenan dan stabilitas model dalam hal performa. Berikut hasil pelatihan yang baru terhadap Ekperimen nomor 27 dan hasil evaluasinya.

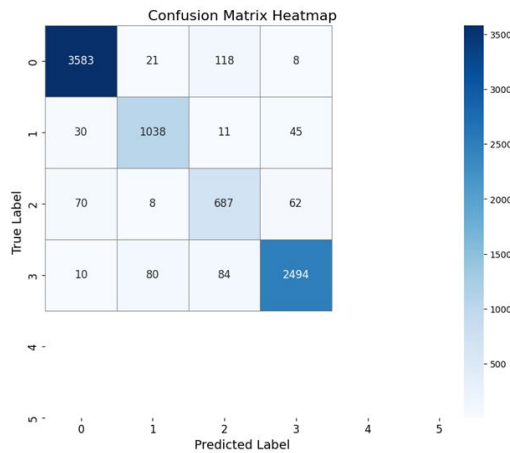


Gambar 7. Perbandingan Akurasi Pelatihan dan Validasi pada Pelatihan Model terbaik sebelum *fine-tuning* (a) dan setelah *fine-tuning* (b).



Gambar 8. Perbandingan Loss Pelatihan dan Validasi pada Pelatihan Model terbaik sebelum *fine-tuning* (a) dan setelah *fine-tuning* (b).

Kedua kurva tersebut memperlihatkan pelatihan model peningkatan akurasi dan penurunan *loss* disertai validasinya. Model menunjukkan fluktuasi selama pelatihan kedua setelah *fine-tuning*, yang terlihat jelas pada epoch ketiga dari sesi pelatihan tersebut. Model checkpoint telah diterapkan sejak awal untuk menyimpan versi terbaik model selama pelatihan. Early stopping dengan parameter patience 2 digunakan karena model tidak langsung berhenti pada iterasi dengan nilai yang lebih baik. Tujuannya adalah untuk menghentikan pelatihan sebelum model mulai *overfitting*, sambil memberi waktu untuk penyesuaian tanpa menurunkan performa. Selain itu, menurunkan nilai learning rate setelah *fine-tuning* bertujuan untuk membuat proses pelatihan lebih halus, membantu model mencapai hasil yang lebih stabil. Dengan pendekatan ini, model dapat lebih baik dalam mengatur parameter untuk meningkatkan akurasi dan hasil akhir.



Gambar 9. Confusion Matrix Model Terbaik

TABEL II  
SKENARIO PENELITIAN

| Label  | Accuracy | Precision | Recall | F1-Score |
|--------|----------|-----------|--------|----------|
| CNV    | 0.93     | 0.97      | 0.96   | 0.97     |
| DME    |          | 0.90      | 0.92   | 0.91     |
| DRUSEN |          | 0.76      | 0.83   | 0.80     |
| NORMAL |          | 0.96      | 0.93   | 0.95     |

Dari hasil confusion matrix dalam gambar 4.5 model berhasil mengklasifikasikan dengan benar masing-masing label dengan jumlah besar.

Tabel II menunjukkan performa model yang sangat baik secara keseluruhan. Dengan akurasi 0.9348, model ini efektif dalam mengklasifikasikan data dengan benar. Presisi dan *Recall* masing-masing mencapai 0.97 dan 0.96, menandakan bahwa model sangat baik dalam mendeteksi kelas positif dan memiliki kesalahan yang minim dalam klasifikasi positif. *F1-Score* yang tinggi di semua kelas, terutama 0.96 untuk kelas 4, menunjukkan keseimbangan yang baik antara presisi dan *Recall*. Meskipun model sedikit kurang efektif dalam *Recall* untuk kelas 3 (0.76), kinerja keseluruhan tetap sangat baik dengan kemampuan mendeteksi kelas lainnya dengan sangat baik.

#### IV. KESIMPULAN DAN SARAN

Berdasarkan pada penelitian tentang Implementasi *Parameter Tuning* dalam model CNN didapatkan kesimpulan sebagai berikut. Model yang menerapkan parameter tuning berkala dan diperbarui selama melakukan eksperimen untuk menemukan hasil variatif telah dapat mengembangkan model performa lebih optimal dan signifikan.

Berdasarkan hasil penelitian dengan menerapkan *Parameter Tuning* yang telah diperoleh maka ada dua saran yang bisa diberikan untuk meningkat penelitian menjadi lebih baik dengan melakukan penelitian lanjutan menggunakan dataset dengan ukuran besar dan stabil setiap label dan mempertimbangkan variasi hyperparameter tuning yang lebih banyak agar model yang kembangkan dapat dimaksimalkan potensinya.

#### V. KUTIPAN

- [1] Nurfitra D. Royani, "IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI PUBLIKASI ILMIAH," 2018.
- [2] A. Anhar and R. A. Putra, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 2, p. 466, Apr. 2023, doi: 10.26760/elkomika.v11i2.466.
- [3] W. A. Firmansyah, U. Hayati, and Y. A. Wijaya, "ANALISA TERJADINYA OVERFITTING DAN UNDERFITTING PADA ALGORITMA NAIVE BAYES DAN DECISION TREE DENGAN TEKNIK CROSS VALIDATION," 2023.
- [4] A. A. Faudisyah, K. Dwi Hartomo, and H. D. Purnomo, "Deteksi Cacat pada Isolasi Trafo Secara Visual menggunakan Algoritma Convolutional Neural Network (CNN)," *Fakultas Teknologi Informasi*, 2023, doi: 10.35870/jti.
- [5] M. K. Rusia and D. K. Singh, "An efficient CNN approach for facial expression recognition with some measures of overfitting," *International Journal of Information Technology (Singapore)*, vol. 13, no. 6, pp. 2419–2430, Dec. 2021, doi: 10.1007/S41870-021-00803-X/METRICS.
- [6] J. Kim and L. Tran, "Retinal Disease Classification from OCT Images Using Deep Learning Algorithms," 2021.
- [7] F. He, T. Liu, and D. Tao, "Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence," 2019.
- [8] V. Verdy, "KLASIFIKASI PENYAKIT MATA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK MODEL RESNET-50," Nov. 2023.
- [9] E. Sutanty, Maukar, D. K. Astuti, and Handayani, "Penerapan Model Arsitektur VGG16 Untuk Klasifikasi Jenis Sampah," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 407–419, Sep. 2023, doi: 10.51454/DECODE.V3I2.331.
- [10] A. B. J. Malyan, R. R. Putra, E. Laila, A. T. Wardhana, M. Fikri, and I. G. T. Isa, "Epoch Tuning Hyperparameter in Fire Image Classification at University Sjakhyakirti," in *Atlantis Press*, 2023, pp. 355–364, doi: 10.2991/978-94-6463-118-0\_37.
- [11] J. A. Wuisan, A. Jacobus, and S. Sompie, "Data Balancing Methods on Radiographic Image Classification on Unbalance Dataset," *Jurnal Teknik Elektro dan Komputer*, vol. 11, no. 1, pp. 1–8, Apr. 2022, doi: 10.35793/JTEK.11.1.2022.37186.
- [12] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed Tools Appl*, vol. 79, no. 19–20, pp. 12777–12815, May 2020, doi: 10.1007/s11042-019-08453-9.
- [13] I. Wulandari, H. Yasin, and T. Widiari, "KLASIFIKASI CITRA DIGITAL BUMBU DAN REMPAH DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)," *JURNAL GAUSSIAN*, 2020, [Online]. Available: <https://ejournal3.undip.ac.id/index.php/gaussian/>
- [14] T. A. E. Putri, T. Widiari, and R. Santoso, "PENERAPAN TUNING HYPERPARAMETER RANDOMSEARCHCV PADA ADAPTIVE BOOSTING UNTUK PREDIKSI KELANGSUNGAN HIDUP PASIEN GAGAL JANTUNG," *Jurnal Gaussian*, vol. 11, no. 3, pp. 397–406, Jan. 2023, doi: 10.14710/j.gauss.11.3.397-406.
- [15] K. Sun, A. Roy, and J. M. Tobin, "Artificial intelligence and machine learning: Definition of terms and current concepts in critical care research," *J Crit Care*, vol. 82, p. 154792, Aug. 2024, doi: 10.1016/J.JCRC.2024.154792.
- [16] M. L. Lailis Syafa'ah, "Penerapan Deep Learning untuk Prediksi Kasus Aktif Covid-19," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 2021.
- [17] C. B. Gonçalves, J. R. Souza, and H. Fernandes, "CNN architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images," *Comput Biol Med*, vol. 142, Mar. 2022, doi: 10.1016/j.combiomed.2021.105205.
- [18] X. Meng, Z. Qiao, C. Wang, and Z. Zhang, "Artificial Regularization Parameter Analysis for the No-Slope-Selection Epitaxial Thin Film Model," *CSIAM Transactions on Applied Mathematics*, vol. 1, no. 3, pp. 441–462, Sep. 2020, doi: 10.4208/csiam-am.2020-0015.
- [19] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark," *Neurocomputing*, Sep. 2021, [Online]. Available: <http://arxiv.org/abs/2109.14545>

- [20] M. D. Putro, J. Litouw, and V. C. Poekoel, “Low-resolution facial emotion recognition on low-cost devices,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 2, p. 2201, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2201-2211.
- [21] R. Andonie and A. C. Florea, “Weighted random search for CNN hyperparameter optimization,” *International Journal of Computers, Communications and Control*, vol. 15, no. 2, pp. 1–11, 2020, doi: 10.15837/IJCCC.2020.2.3868.
- [22] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, “Don’t Decay the Learning Rate, Increase the Batch Size,” *ICLR*, Nov. 2018, [Online]. Available: <http://arxiv.org/abs/1711.00489>
- [23] W. Hastomo, A. S. Bayangkari Karno, N. Kalbuana, A. Meiriki, and Sutarno, “Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1933/1/012050.
- [24] F. Radenović, G. Tolias, and O. Chum, “CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples,” Apr. 2016, [Online]. Available: <http://arxiv.org/abs/1604.02426>
- [25] A. Maharil, “PERBANDINGAN ARSITEKTUR VGG16 DAN RESNET50 UNTUK REKOGNISI TULISAN TANGAN AKSARA LAMPUNG,” *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 3, no. 2, pp. 236–243, 2022, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [26] A. Sai Bharadwaj Reddy and D. Sujitha Juliet, “Transfer Learning with ResNet-50 for Malaria Cell-Image Classification,” *IEEE*, 2019.



**Febrian Glorio Lalamentik** lahir di Matani, pada 18 Agustus 2002. Penulis adalah anak pertama dari empat bersaudara dari pasangan Bill Lalamentik dan Olvi Tulung. Penulis mengikuti pendidikan Sekolah Dasar di SD Inpress Matani lulus pada tahun 2014, kemudian melanjutkan ke jenjang pendidikan Sekolah Menengah Pertama di SMP Advent Tumpaan lulus pada tahun 2017, dan masuk ke Sekolah Menengah Atas di SMA Negeri 1 Amurang lulus pada tahun 2020. Penulis kemudian mengambil pendidikan lanjutan di Universitas Sam Ratulangi dengan masuk menggunakan jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) ke Program Studi S-1 Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik. Selama perkuliahan, penulis aktif dalam komunitas UNSRAT IT COMMUNITY sebagai bagian pengurus komunitas bidang Artificial Intelligence, penulis pernah menjadi Koordinator Posko KKT 138 di Desa Lolah Dua pada tahun 2024, penulis sekaligus menjadi Posdcaster pertama di UKM Pers dan Penyiaran FT UNSRAT, dan terakhir pernah mengikuti Studi Independen Bangkit Academy.