

# Foodstuffs Image Classification for People with Color Blindness

Klasifikasi Gambar Bahan Makanan untuk Penderita Buta Warna

Rheza Timothy Tedjo, Alwin M. Sambul, Arie S. M. Lumenta

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

E-mails: [rhezatedjo@gmail.com](mailto:rhezatedjo@gmail.com), [asambul@unsrat.ac.id](mailto:asambul@unsrat.ac.id), [al@unsrat.ac.id](mailto:al@unsrat.ac.id)

Received: 12 January 2022 ; revised: 3 July 2022; accepted: 5 July 2022

**Abstract** — Patients suffering from color blindness have difficulty when buying groceries like foodstuffs, where they were unable to recognize the color difference between a fresh foodstuffs and rotten foodstuffs at a glance. When buying foodstuffs, they usually ask for help from their close relations to shop together to help them choose which foodstuffs to buy. This study aims to build a mobile application that can recognize fresh and rotten foodstuffs to be purchased with the help of machine learning model so that it can help users with color blindness disabilities to correctly choose which foodstuffs to buy. The model used is based on ResNet50 and MobileNet and data augmentation for data balancing, and uses a confusion matrix for evaluating the trained model. The result of this study achieved a model that can classify images taken with a recorded performance of 98% accuracy in the multi-class classification.

**Key words**— application, assistance, color blindness, convolutional neural network, foodstuffs, fruit images, image classification, machine learning, transfer learning.

**Abstrak** — Penderita gangguan buta warna memiliki kesulitan saat membeli bahan makanan, di mana mereka tidak dapat mengenali perbedaan warna bahan makanan yang segar dan busuk dalam sekali lihat. Mereka biasanya meminta pertolongan orang dekat untuk berbelanja bersama-sama dengan mereka agar dapat membantu mereka memilih bahan makanan yang akan dibeli. Tujuan dari penelitian ini adalah membuat aplikasi pada platform *mobile Android* yang dapat mengenali bahan makanan segar dan busuk mana yang akan dibeli dengan bantuan model *machine learning* agar dapat membantu pengguna yang memiliki keterbatasan untuk membedakan warna agar tidak salah memilih bahan makanan yang akan dibeli. Model yang digunakan memiliki dasar arsitektur *ResNet50* dan *MobileNet* dengan melakukan augmentasi untuk melakukan penyeimbangan pada data, dan menggunakan *confusion matrix* untuk mengevaluasi performa model yang dilatih. Hasil dari penelitian ini adalah model yang dapat mengklasifikasikan gambar yang diambil dengan akurasi sebanyak 98% dalam melakukan klasifikasi kelas banyak.

**Kata kunci** — aplikasi, bahan makanan, bantuan, buta warna, convolutional neural network, gambar buah, klasifikasi gambar, machine learning, transfer learning.

## I. PENDAHULUAN

Penderita penglihatan buta warna baik parsial atau buta

warna total memiliki keterbatasan untuk dapat mengenali perbedaan warna. Terdapat kira-kira 300 miliar orang di dunia yang menderita buta warna, dan dari seluruh populasi di Indonesia, sebanyak kl 3.5% dari populasi pria ditemukan menderita gangguan ini [1].

Penderita gangguan ini memiliki kesulitan saat memilih bahan makanan berwarna, mereka tidak dapat mengenali perbedaan warna bahan makanan yang segar dan busuk dalam sekali melihat. Pada suatu forum terbuka, Penderita buta warna menyatakan bahwa mereka biasanya meminta pertolongan orang dekat untuk berbelanja bersama-sama agar dapat memilih bahan makanan yang akan dibeli .

*Smartphone* yang merupakan salah satu alat komunikasi yang sekarang menjadi bagian penting dari kehidupan digital saat ini merupakan salah satu alat yang dapat memiliki kegunaan dalam membantu pasien-pasien yang memiliki gangguan ini.

*Google* dalam membantu pasien yang memiliki kebutuhan khusus telah membuat 3 fitur untuk *Android operating system* pada tahun 2020 yaitu pembuatan *action blocks* yang dibuat untuk penyandang disabilitas kognitif agar dapat lebih mudah mengakses layanan yang sering digunakan pada *smartphone* pengguna tersebut, fitur *instant transcription* yang dibuat untuk membantu penyandang disabilitas pendengaran dengan cara menyajikan teks untuk setiap *audio* yang diterima, dan fitur *sound amplifier* yang merupakan fitur untuk penyandang disabilitas pendengaran dengan cara membuat suara yang lemah menjadi lebih jernih dan lebih keras, tanpa membuat suara yang sudah keras menjadi lebih keras [2], [3]

Penelitian ini menggunakan teknologi *Machine Learning* yang bertujuan untuk mengembangkan sebuah model *CNN deep learning* yang dapat mengenali kondisi bahan makanan yang ingin dibeli oleh pengguna khususnya yang memiliki gangguan penglihatan buta warna sehingga mereka memilih bahan makanan yang segar untuk dibeli.

### A. Penelitian Terkait

Penelitian sebelumnya yang dilakukan oleh Miftahus Sholihin menggunakan sebanyak 150 *dataset* citra insang yang dibagi menjadi 105 *data train* dan 54 *data testing*. Model jaringan yang dibentuk merupakan 3-layer konvolusi dan

menghasilkan sebuah model dengan akurasi *training* sebesar 100% dan akurasi *testing* sebesar 97,7% [4].

Penelitian yang dilakukan oleh Eko Prasetyo melakukan penyesuaian kepada beberapa *pretrained model* yaitu *Xception*, *ResNet50*, *MobileNetV1*, dan *VGG16*. *Dataset* yang digunakan adalah foto mata ikan bandeng yang terbagi menjadi 2 kategori yaitu ikan sangat segar, dan ikan tidak segar. Dari total 154 *dataset* lalu terbagi menjadi 44 data latih dan 15 data validasi ikan sangat segar, dan 48 data latih dan 16 data validasi ikan tidak segar. Setelah mencoba menggunakan *pretrained model* terdapat hasil model *VGG16* dengan akurasi terbaik yaitu sebesar 98% dengan kinerja validasi sebesar 100% [5].

Penelitian yang dilakukan oleh Kun-Li Tsai untuk membuat sebuah sistem pengemasan obat yang dapat mengenali obat dengan menggunakan model *convolutional neural network* yang dilatih dengan dasar arsitektur *GoogLeNet* dengan melakukan augmentasi pada data latih karena data yang tersedia sangatlah kurang. Dalam tahap pra pemrosesan data penelitian pada penelitian ini menggunakan konsep *triplet network* yang dimana terdapat 3 input yang akan diterima model yaitu *anchor sample* dengan *positive sample* dan *negative sample* yang diambil dari *anchor sample*. Hasil dari pelatihan model mencapai hasil latih lebih dari 90%, tetapi memiliki hasil *testing* yang terbilang rendah karena kurangnya variasi data latih [6].

Penelitian yang dilakukan oleh Fangyuan Lei untuk mencoba mengurangi waktu yang diperlukan untuk melatih sebuah model *convolutional neural network* dan sekaligus meningkatkan akurasi hasil latih dengan membuat sebuah model yang dilatih dengan menambahkan *layer batch normalization* yang akan menormalisasikan setiap *feature map* yang didapatkan setelah data dimasukkan dalam *layer* konvolusi pertama. Untuk menguji penelitian yang dilakukan digunakan *dataset MNIST*, *fashion-MNIST* dan *dataset CIFAR10* dalam proses pengujian. Hasil pengujian model mencapai 99% akurasi dalam melakukan klasifikasi pada *dataset MNIST*, 93% akurasi dalam melakukan klasifikasi pada *dataset fashion-MNIST*, dan 86% akurasi dalam melakukan klasifikasi pada *dataset CIFAR10* [7].

Penelitian yang dilakukan oleh Jozef Gorzelany dalam mengembangkan sebuah model *artificial neural network (ANN)* yang dapat memprediksi keterkaitan antara durasi penyimpanan buah *Cranberry* dengan perubahan yang dapat diamati pada buah tersebut yaitu deformasi, energi yang hilang, dan modulus elastisitas dari buah tersebut. Dalam penelitian tersebut dibuat 3 model *ANN* masing-masing untuk menghitung perubahan-perubahan tersebut dimana model *ANNA* yang dikembangkan untuk memprediksi deformasi pada buah, model *ANNb* yang dikembangkan untuk memprediksi energi yang hilang pada buah tersebut dan model *ANNc* untuk memprediksi perubahan modulus elastisitas pada buah tersebut. Hasil pelatihan dari model yaitu model *ANNA* dengan akurasi latih sebanyak 69%, model *ANNb* dengan akurasi latih sebanyak 80%, dan model *ANNc* dengan akurasi latih sebanyak 89% [8].

Penelitian yang dilakukan oleh Ibrahim Kandel untuk mengamati pengaruh parameter jumlah *batch size* dalam pelatihan model *convolutional neural network* dilakukan dengan menggunakan gambar medis hispatologi sebagai *dataset* dalam penelitian ini. Pada penelitian ini dilakukan

pelatihan pada model dengan menggunakan parameter *batch size* yang berbeda untuk melihat perbedaan performa hasil latih oleh karena perbedaan nilai dalam parameter *batch size*. Digunakan 5 nilai parameter *batch size* sebagai pembandingan yaitu *batch size* 16, 32, 64, 128, dan 256 dalam pelatihan. Tercatat hasil dari pelatihan model yang dilatih dengan nilai parameter *batch size* 16 dan 256 memiliki akurasi tertinggi yaitu sebesar 96,6% untuk pelatihan menggunakan *optimizer Adam*, dan model yang dilatih dengan nilai parameter *batch size* 32 dan 256 memiliki akurasi tertinggi yaitu sebesar 95,7% untuk pelatihan menggunakan *optimizer SGD* [9].

## B. Artificial Intelligence (AI)

*Artificial Intelligence (AI)* adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan [10]. Menurutnya untuk mencapai kecerdasan buatan telah dikembangkan tiga metode, yaitu *Fuzzy Logic*, *Evolutionary Computing* dan *Machine Learning*

## C. Machine Learning

*Machine Learning* merupakan proses di mana sebuah komputer dapat mengembangkan kinerja mesin itu sendiri dalam hal menganalisa *file* gambar atau foto dengan menggabungkan data baru secara terus-menerus kedalam sebuah model statistik yang sudah ada [11].

## D. Deep Learning

*Deep Learning* merupakan teknik dalam *neural network (NN)* yang menggunakan teknik tertentu seperti *restricted boltzmann machine (RBM)* untuk mempercepat proses pembelajaran dalam *NN* yang menggunakan lapis yang banyak atau lebih dari 7 lapis [12].

Dengan adanya *deep learning*, waktu yang dibutuhkan untuk melakukan *training* pada sebuah model *machine learning* akan semakin sedikit karena masalah hilangnya gradien pada propagasi balik akan semakin rendah.

## E. Transfer Learning

Transfer Learning merupakan penggunaan data model *machine learning* yang telah dilatih sebelumnya untuk menyelesaikan masalah lain dengan melatih kembali model tersebut dengan *dataset* yang baru [13].

## F. Convolutional Neural Network (CNN)

*Convolutional neural network deep learning* merupakan solusi mutakhir untuk *visual recognition* berskala besar yang mengikuti prinsip dasar seperti meningkatkan *network depth* dan pembuatan *construction highways* [14].

Struktur *convolutional neural network* biasanya terdiri atas ekstraktor fitur (*feature extractor*) yang didalamnya memiliki beberapa *convolutional layer (Conv2D)* dan biasanya diikuti dengan *pooling layer (MaxPooling2D)* serta sebuah pengklasifikasi *softmax* apabila jumlah kelas *output* model yang dilatih memiliki kelas banyak. Tugas dari *convolutional layer* adalah ekstraksi fitur, sedangkan *pooling layer* bertugas mengurangi dimensi data untuk menghemat waktu komputasi. Hasil ekstraksi fitur dari seluruh *convolutional layer* dan *pooling layer* lalu dimasukkan ke lapisan *softmax* untuk proses klasifikasi [15].

### G. Augmentasi

Augmentasi merupakan sebuah cara untuk mengurangi kemungkinan terjadinya *overfitting* pada model yang akan dilatih dengan berfokus pada dasar masalah penyebab terjadinya *overfitting*, yaitu data latih model. Dengan melakukan augmentasi maka akan lebih banyak informasi yang dapat diekstraksi dari data latih tersebut, dengan demikian apabila semakin banyak informasi yang diekstraksi dari data latih maka *error value* saat pelatihan dan validasi dapat lebih berkurang dibandingkan dengan data latih yang tidak diaugmentasi [16].

### H. TensorFlow

*Tensorflow* adalah sebuah *open-source framework* yang dikembangkan oleh *Google researchers* dengan kemampuan untuk beroperasi pada skala besar di dalam lingkungan heterogen. *Tensorflow* awalnya dikembangkan oleh para peneliti dan *engineer* yang bekerja di *Google Brain Team* dalam organisasi riset *Machine Intelligence Google*, untuk keperluan menjalankan *machine learning* dan riset jaringan *deep neural network*. Namun, sistem ini juga cukup umum untuk diterapkan pada aneka *domain* lainnya. *Tensorflow* ini digunakan sebagai *backend* dari *keras*, yang artinya *tensorflow* berjalan di bagian belakang layar bersama dengan *library keras* dari *deep learning* [17]. *Tensorflow* bisa mendukung beberapa aplikasi, dengan fokus pada pelatihan dan inferensi untuk *deep learning* [18].

### I. TensorFlow Lite

*Tensorflow lite* merupakan evolusi dari *tensorflow mobile* yang sebelumnya mendukung deployment pada perangkat bergerak (*mobile device*) dan perangkat tertanam (*embedded device*) [19]. *Tensorflow lite* merupakan seperangkat alat yang memungkinkan proses klasifikasi terjadi pada perangkat (*on-device*) dengan cara membantu pengembang menjalankan model yang telah dilatih sebelumnya pada perangkat bergerak, perangkat tertanam dan *edge devices* yang meliputi *gadgets*, *smartphone*, pemutar musik, dan *wearables* [20].

## II. METODE

### A. Pengumpulan Dataset

Dalam pengembangan model, dilakukan pengumpulan data gambar yang akan digunakan dalam proses pelatihan pada model nanti.

Sumber-sumber dari *dataset* yang digunakan adalah sebagai berikut:

- 1) *Dataset* bernama “*Fresh and Stale Images of Fruits and Vegetables*” pada situs *kaggle* yang setiap gambarnya berukuran 512 x 512 pixel dengan jumlah data di setiap kategori berbeda-beda tetapi terdapat 1693 jumlah gambar untuk kategori apel segar, 1781 jumlah gambar untuk kategori apel busuk, 1581 jumlah gambar untuk kategori pisang segar, 1467 jumlah gambar untuk kategori pisang busuk, 1466 jumlah gambar untuk kategori jeruk segar, 1595 jumlah gambar untuk kategori jeruk busuk, 981 jumlah gambar untuk kategori tomat segar, 982 jumlah gambar untuk kategori tomat busuk, 990 jumlah gambar untuk kategori paprika hijau segar, 901 jumlah gambar untuk kategori paprika hijau busuk, 327 jumlah gambar

untuk kategori paria segar, dan 357 jumlah gambar untuk kategori paria busuk

- 2) Pencarian *Google* dengan menggunakan extension *Fatkun* yang memungkinkan proses mengunduh gambar dengan jumlah yang banyak dengan efisien dengan kata kunci pencarian “*bitter gourd*” yang menghasilkan 144 gambar untuk kategori paria segar, dan kata kunci pencarian “*rotten bitter gourd bitter melon*” yang menghasilkan 41 gambar untuk kategori paria busuk
- 3) Foto lokal yang diambil dari toko *FreshMart* lokal dengan mengambil sekurangnya 10 gambar untuk masing-masing kategori

Tabel I menampilkan jumlah data yang digunakan dari setiap sumber. Data yang telah dikumpulkan akan dibagi menjadi data latih, data validasi dan data uji dengan rasio pembagian yaitu sebanyak 90% untuk data latih, 5% data validasi dan 5% data uji. Hasil pembagian *dataset* dapat dilihat pada Tabel II

### B. Pra Pemrosesan Dataset

Pra pemrosesan atau *preprocessing* pada *dataset* dilakukan dengan tujuan untuk mempersiapkan data sebelum digunakan agar dapat meningkatkan performa model yang akan dilatih nanti.

Pra pemrosesan yang dilakukan pada penelitian ini adalah sebagai berikut:

- 1) *Data Resize*  
Mengubah ukuran awal data gambar *dataset* menjadi ukuran 150 x 150 pixel.
- 2) *Data Rescaling*  
Mengubah nilai pada *range* warna RGB gambar dari 1-255 menjadi nilai 0-1.
- 3) *Augmentasi Data*  
Augmentasi dilakukan pada data latih untuk menyeimbangkan jumlah data latih yang akan digunakan pada model nantinya agar saat proses latih model tidak akan condong atau memiliki *bias* kepada kategori-kategori tertentu yang memiliki jumlah data latih lebih banyak dibandingkan kategori-kategori lain yang memiliki jumlah data latih yang lebih sedikit. Untuk perbandingan jumlah data latih sebelum dan sesudah *Augmentasi* data dapat dilihat pada Tabel III.

### C. Arsitektur Model

Pada penelitian ini dilakukan perbandingan antara model jaringan dengan *5-layer* konvolusi yang dinamakan model *baseline*, dan model yang telah dilatih terlebih dahulu dengan menggunakan metode *transfer learning* dengan menggunakan model *ResNet50* dan *MobileNet* untuk mendapatkan model dengan performa terbaik.

### D. Metric Evaluasi

Evaluasi pada performa model dalam melakukan klasifikasi dapat dihitung menggunakan acuan atau *metric Precision*, *Recall* dan *F1-score*. Nilai acuan tersebut dihitung dengan menggunakan 4 nilai yang akan didefinisikan terlebih dahulu yaitu nilai *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. *TP* didapatkan apabila data positif diklasifikasikan sebagai kelas positif, *TN* didapatkan apabila data positif diklasifikasikan sebagai kelas negatif, *FP* didapatkan apabila data negatif diklasifikasikan

sebagai kelas positif, dan FN didapatkan apabila data negatif diklasifikasikan sebagai kelas negatif.

Nilai akurasi merupakan jumlah data yang diklasifikasi dengan benar, nilai *precision* merupakan nilai perbandingan antara nilai TP dengan jumlah data yang diklasifikasi sebagai kelas positif, nilai *recall* merupakan nilai perbandingan antara nilai TP dengan jumlah data positif, dan *f1-score* merupakan *harmonic mean* dari *precision* dan *recall*. Nilai akurasi dapat dihitung dengan menggunakan persamaan (1), nilai *precision* dapat dihitung dengan menggunakan persamaan (2), nilai *recall* dapat dihitung dengan menggunakan persamaan (3), dan nilai *f1-score* dapat dihitung dengan menggunakan persamaan (4).

### III. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan mengenai hasil pelatihan setiap model menggunakan data yang telah diaugmentasi sebelumnya. Dan dengan melakukan pelatihan pada beberapa model tanpa melakukan augmentasi pada data, kita dapat melihat dampak penyeimbangan data terhadap performa model yang di latih.

#### A. Pembuatan Model

Berdasarkan arsitektur model yang digunakan, pada penelitian ini akan didefinisikan 3 model dengan nama model ResNet50, model MobileNet, dan model *baseline*.

Model ResNet50 dan model MobileNet dibangun dengan menggunakan pendekatan *transfer learning* kepada *pre-trained* model dengan basis arsitektur ResNet50 dan MobileNet sebagai *feature extractor* sebelum masuk pada *fully connected layer* yang memiliki 2-layer yang pada *layer* pertama memiliki 512 neuron dan pada *layer* kedua memiliki 12 neuron sebagai hasil klasifikasi.

Model *baseline* dibuat dengan 5-layer konvolusi yang di ikuti oleh *pooling layer* dan dua *fully connected layer*. Pada *layer* pertama *layer convolutional 2D* memiliki 32 neuron dengan ukuran filter 3 x 3 yang menerima input gambar dengan ukuran 150 x 150 dengan model warna *red green blue*, yang diikuti dengan *layer batch normalization* yang akan menormalisasikan *layer* sebelumnya dengan cara mengurangi rata-rata sebuah bagian gambar dan membagi bagian tersebut dengan *deviation standard* bagian gambar itu sendiri. Setelah *layer batch normalization* terdapat *layer max pooling* dengan ukuran 2 x 2

untuk mengurangi jumlah parameter dengan cara melakukan kompresi pada gambar. Pada *layer* ke dua *layer convolutional 2D* memiliki 64 neuron dan diikuti dengan *layer Max Pooling*, pada *layer* ke tiga *layer convolutional 2D* memiliki 64 neuron dan diikuti dengan *layer Max Pooling*, pada *layer* ke empat *layer convolutional 2D* memiliki 128 neuron dan diikuti dengan *layer Max Pooling*, dan pada *layer* ke lima *layer convolutional 2D* memiliki 128 neuron dan diikuti dengan *layer Max Pooling*.

Sebelum hasil dari *layer* ke lima masuk ke *fully connected layer*, dilakukan flattening yaitu dimana hasil dari *layer* ke 5 *convolutional 2D* yang memiliki ukuran 2 dimensi diubah menjadi sebuah vektor 1 dimensi. Pada *fully connected layer* akan dilakukan klasifikasi dimana dari 512 neuron akan turun menjadi 12 neuron untuk pengklasifikasian.

Berdasarkan arsitektur model yang digunakan, pada penelitian ini dibuat 3 model yang dilatih dengan data yang diaugmentasi dan mengambil 2 model diantaranya untuk dilatih kembali dengan data yang tidak diaugmentasi

Untuk itu akan didefinisikan 5 model yaitu model A yang dilatih menggunakan data yang diaugmentasi dan dengan dasar arsitektur model ResNet50, model B yang dilatih menggunakan data yang diaugmentasi dan dengan dasar arsitektur model MobileNet, model C yang dilatih menggunakan data yang diaugmentasi dan dengan menggunakan dasar arsitektur model *baseline*, model D yang dilatih menggunakan data yang tidak diaugmentasi dan dengan dasar arsitektur model ResNet50, dan model E yang dilatih menggunakan data yang tidak diaugmentasi dan dengan menggunakan dasar arsitektur model *baseline*. Gambar 1 menampilkan arsitektur model ResNet50 sedangkan gambar 2 menampilkan arsitektur model MobileNet dan gambar 3 menampilkan arsitektur model *baseline*.

Dalam proses pelatihan diberikan jumlah *epoch* sebanyak 50 dengan *batch size* sebanyak 32 untuk setiap model, ukuran gambar input sebesar 150 x 150 pixel, menggunakan *optimizer* Adam dan memiliki *learning rate* sebesar 0,001 yang merupakan nilai *default*.

TABEL I  
JUMLAH DATASET BERDASARKAN SUMBER

Kategori	Data Kaggle	Data Fatkun	Data Lokal	Jumlah Data
Apel Segar	1693	-	20	1713
Apel Busuk	1781	-	10	1791
Pisang Segar	1581	-	33	1614
Pisang Busuk	1467	-	10	1477
Jeruk Segar	1466	-	14	1480
Jeruk Busuk	1595	-	12	1607
Tomat Segar	981	-	34	1015
Tomat Busuk	982	-	36	1018
Paprika Segar	990	-	11	1001
Paprika Busuk	901	-	22	923
Paria Segar	327	144	15	486
Paria Busuk	357	41	10	408

TABEL II  
HASIL PEMBAGIAN DATASET

Data	Jumlah
Latih	13190
Validasi	760
Uji	775

TABEL III  
PERBANDINGAN DATA LATIH SEBELUM DAN SETELAH AUGMENTASI

Kategori	Data Latih Sebelum Augmentasi	Data Latih Setelah Augmentasi
Apel segar	1597	1597
Apel busuk	1628	1628
Pisang segar	1455	1455
Pisang busuk	1330	1330
Jeruk segar	1333	1333
Jeruk busuk	1447	1447
Tomat segar	925	1387
Tomat busuk	919	1311
Paprika segar	905	1220
Paprika busuk	832	1292
Paria segar	447	1698
Paria busuk	372	1323

$$accuracy = TP + TN \quad (1)$$

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

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

$$f1\ Score = 2 * \frac{precision * recall}{precision+recall} \quad (4)$$

$$macro\ average = \frac{f1score_a+f1score_b+\dots+f1score_n}{f1score_n} \quad (5)$$

```

Model: "sequential_1"
Layer (type) Output Shape Param #
-----
resnet50 (Functional) (None, 2048) 23587712
flatten_1 (Flatten) (None, 2048) 0
dense_2 (Dense) (None, 512) 1049088
dense_3 (Dense) (None, 12) 6156
-----
Total params: 24,642,956
Trainable params: 1,055,244
Non-trainable params: 23,587,712
    
```

Gambar 1. Arsitektur Model ResNet50

```

Model: "sequential_1"
Layer (type) Output Shape Param #
-----
mobilenet_1.00_224 (Functiona (None, 1024) 3228864
nal)
flatten (Flatten) (None, 1024) 0
dense (Dense) (None, 512) 524800
dense_1 (Dense) (None, 12) 6156
-----
Total params: 3,759,820
Trainable params: 530,956
Non-trainable params: 3,228,864
    
```

Gambar 2. Arsitektur Model MobileNet

TABEL V  
 PRECISION DARI KELIMA MODEL

Kategori	Model A	Model B	Model C	Model D	Model E
Apel Segar	97	98	80	98	98
Pisang Segar	100	99	96	100	99
Paria Segar	100	96	100	100	100
Paprika Segar	91	100	98	100	100
Jeruk Segar	96	99	95	96	91
Tomat Segar	96	93	83	94	88
Apel Busuk	96	99	88	93	94
Pisang Busuk	100	97	97	100	99
Paria Busuk	100	100	75	95	95
Paprika Busuk	100	100	98	100	100
Jeruk Busuk	100	90	89	100	99
Tomat Busuk	100	100	91	100	98

```

Model: "sequential"
Layer (type) Output Shape Param #
-----
conv2d (Conv2D) (None, 148, 148, 32) 896
batch_normalization (BatchN (None, 148, 148, 32) 592
ormalization)
max_pooling2d (MaxPooling2D (None, 74, 74, 32) 0
)
conv2d_1 (Conv2D) (None, 72, 72, 64) 18496
max_pooling2d_1 (MaxPooling (None, 36, 36, 64) 0
2D)
conv2d_2 (Conv2D) (None, 34, 34, 64) 36928
max_pooling2d_2 (MaxPooling (None, 17, 17, 64) 0
2D)
conv2d_3 (Conv2D) (None, 15, 15, 128) 73856
max_pooling2d_3 (MaxPooling (None, 7, 7, 128) 0
2D)
conv2d_4 (Conv2D) (None, 5, 5, 128) 147584
max_pooling2d_4 (MaxPooling (None, 2, 2, 128) 0
2D)
flatten (Flatten) (None, 512) 0
dense (Dense) (None, 512) 262656
dense_1 (Dense) (None, 12) 6156
-----
Total params: 547,164
Trainable params: 546,868
Non-trainable params: 296
    
```

Gambar 3. Arsitektur Model Baseline

TABEL IV  
 TUNING DALAM PELATIHAN MODEL

Batch Size	Epochs	Input Shape	Optimizer	Learning Rate
32	50	150, 150, 3	Adam	0.001

TABEL VI  
 RECALL DARI KELIMA MODEL

Kategori	Model A	Model B	Model C	Model D	Model E
Apel Segar	98	98	92	100	96
Pisang Segar	100	100	99	100	100
Paria Segar	100	100	84	96	100
Paprika Segar	100	100	90	100	100
Jeruk Segar	99	97	80	96	96
Tomat Segar	100	100	98	100	100
Apel Busuk	97	92	81	95	100
Pisang Busuk	100	100	99	100	99
Paria Busuk	100	90	100	100	95
Paprika Busuk	89	100	96	100	98
Jeruk Busuk	94	96	90	78	86
Tomat Busuk	96	94	80	94	86

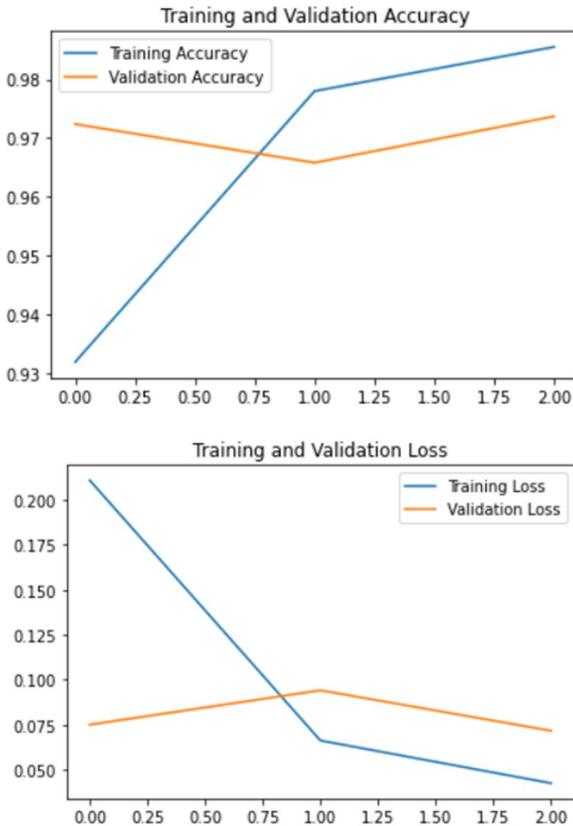
Adapun diberikan fungsi untuk menghentikan proses latihan apabila akurasi model telah mencapai 98% saat jumlah *epoch* belum melewati nilai *epoch* yang ditentukan sebelumnya yaitu 50. Tabel IV menampilkan *tuning* yang digunakan dalam pelatihan kelima model sedangkan pada gambar 4 menampilkan hasil latihan dengan hasil validasi dari model A, gambar 5 menampilkan hasil latihan dengan hasil validasi dari model B, gambar 6 menampilkan hasil latihan dengan hasil validasi dari model C, gambar 7 menampilkan hasil latihan dengan hasil validasi dari model D, dan gambar 8 menampilkan

hasil latih dengan hasil validasi dari model E.

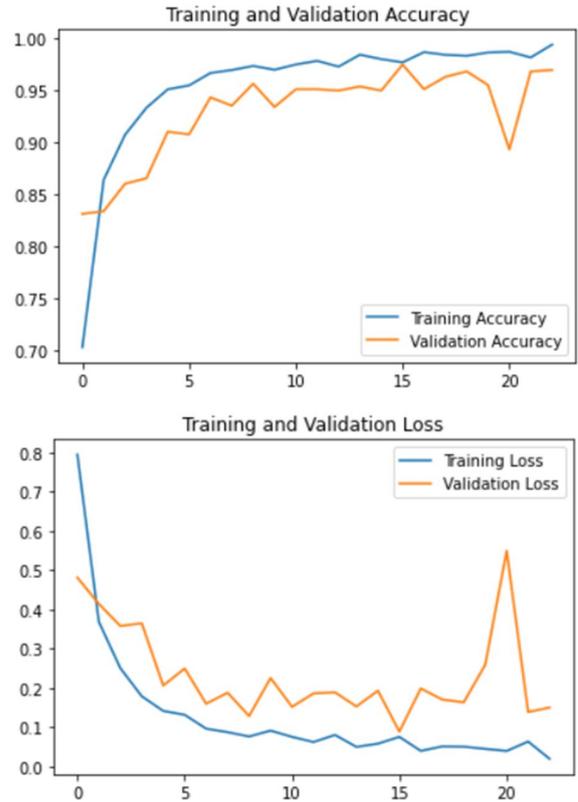
**B. Evaluasi Model**

Agar kita dapat melihat performa model yang telah dilatih dilakukan pengujian terhadap setiap model dengan menggunakan data uji dan hasil prediksi dari setiap model akan dibandingkan menggunakan *confusion matrix*. Tabel VIII menunjukkan *confusion matrix* dari hasil uji pada model A, Tabel IX menunjukkan *confusion matrix* dari hasil uji pada model B, Tabel X menunjukkan *confusion matrix* dari hasil uji pada model C, Tabel XI menunjukkan *confusion matrix* dari hasil uji pada model D, dan Tabel XII menunjukkan *confusion matrix* dari hasil uji pada model E. Performa model akan dinilai dengan menggunakan *metric accuracy*, *precision*, *recall*, dan *f1-score* dari masing-masing model. Tabel V menunjukkan hasil *precision* dari kelima model, tabel VI menunjukkan hasil *recall* dari kelima model, tabel VII menunjukkan hasil *f1-score* dari kelima model, dan tabel XIII menunjukkan hasil akurasi dari kelima model.

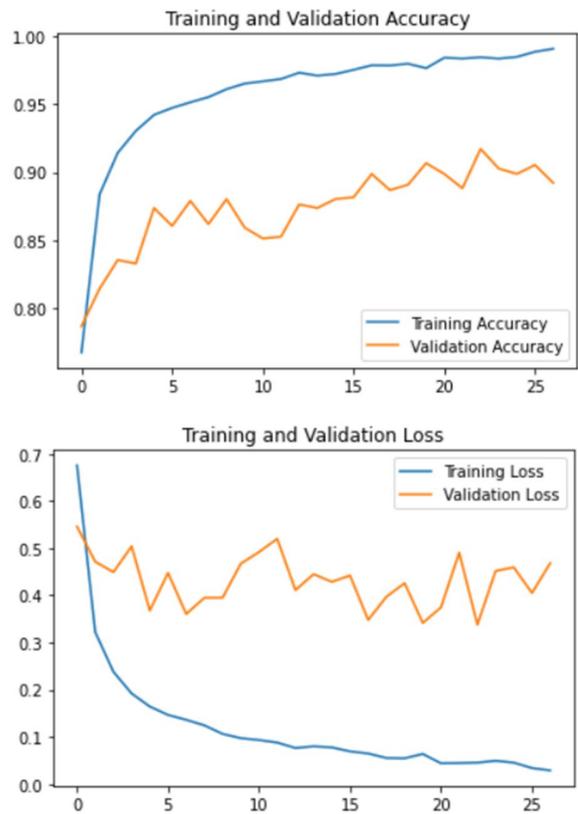
Dalam pemilihan model terbaik, akan diambil model yang memiliki nilai *precision*, *recall* dan *f1-score* tertinggi. Karena pada penelitian ini memiliki hasil *precision*, *recall* dan *f1-score* yang cukup banyak oleh karena jumlah kategori yang akan diklasifikasi juga banyak, maka untuk dapat melihat perbandingan tersebut dengan lebih mudah maka kita dapat menghitung nilai *macro average* yang akan menghitung setiap kelas secara independen lalu mengambil rata-rata dari nilai tersebut.



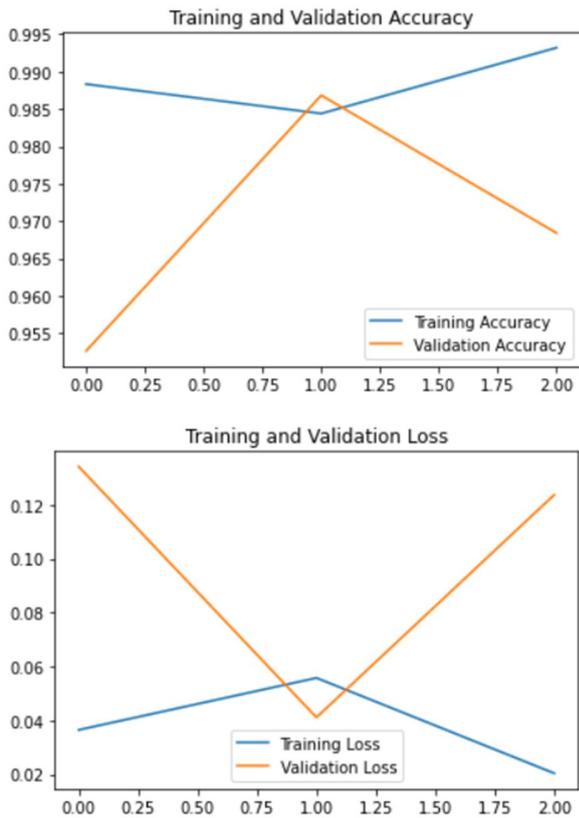
Gambar 4. Grafik Perbandingan Hasil Latih dan Validasi Model A



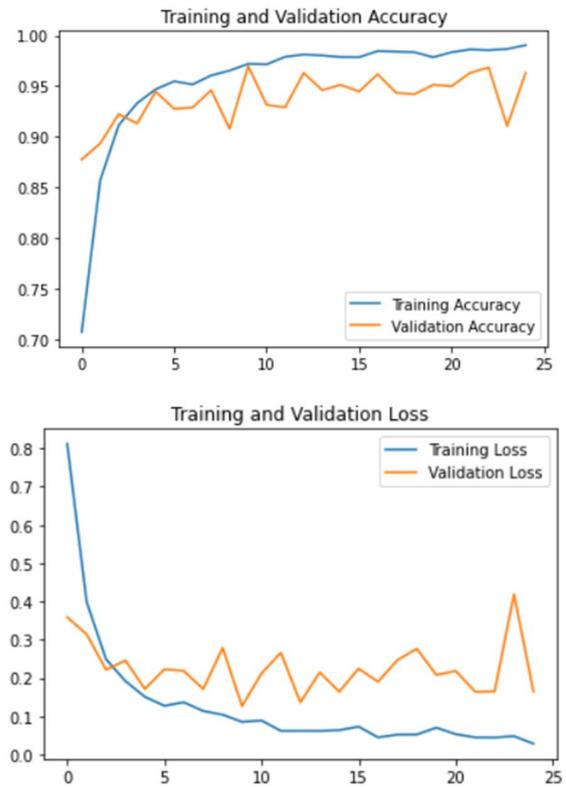
Gambar 5. Grafik Perbandingan Hasil Latih dan Validasi Model B



Gambar 6. Grafik Perbandingan Hasil Latih dan Validasi Model C



Gambar 7. Grafik Perbandingan Hasil Latih dan Validasi Model D



Gambar 8. Grafik Perbandingan Hasil Latih dan Validasi Model E

TABEL VIII  
 CONFUSION MATRIX MODEL A

Kategori	Hasil Prediksi											
	Apel Segar	Pisang Segar	Paria Segar	Paprika Segar	Jeruk Segar	Tomat Segar	Apel Busuk	Pisang Busuk	Paria Busuk	Paprika Busuk	Apel segar	Tomat Busuk
Apel Segar	103	0	0	0	1	0	1	0	0	0	0	0
Pisang Segar	0	80	0	0	0	0	0	0	0	0	0	0
Paria Segar	0	0	25	0	0	0	0	0	0	0	0	0
Paprika Segar	0	0	0	51	0	0	0	0	0	0	0	0
Jeruk Segar	0	0	0	0	73	0	1	0	0	0	0	0
Tomat Segar	0	0	0	0	0	50	0	0	0	0	0	0
Apel Busuk	3	0	0	0	0	0	115	0	0	0	0	0
Pisang Busuk	0	0	0	0	0	0	0	74	0	0	0	0
Paria Busuk	0	0	0	0	0	0	0	0	21	0	0	0
Paprika Busuk	0	0	0	5	0	0	0	0	0	41	0	0
Jeruk Busuk	0	0	0	0	2	0	3	0	0	0	76	0
Tomat Busuk	0	0	0	0	0	2	0	0	0	0	0	48

TABEL IX  
 CONFUSION MATRIX MODEL B



Aktual	Paria Segar	0	0	24	0	0	0	0	0	1	0	0	0
	Paprika Segar	0	0	0	51	0	0	0	0	0	0	0	0
	Jeruk Segar	2	0	0	0	71	0	1	0	0	0	0	0
	Tomat Segar	0	0	0	0	0	50	0	0	0	0	0	0
	Apel Busuk	6	0	0	0	0	0	112	0	0	0	0	0
	Pisang Busuk	0	0	0	0	0	0	0	74	0	0	0	0
	Paria Busuk	0	0	0	0	0	0	0	0	21	0	0	0
	Paprika Busuk	0	0	0	0	0	0	0	0	0	46	0	0
	Jeruk Busuk	7	0	0	0	3	0	8	0	0	0	63	0
	Tomat Busuk	0	0	0	0	0	3	0	0	0	0	0	47

TABEL XII  
CONFUSION MATRIX MODEL E

Kategori	Hasil Prediksi											
	Apel Segar	Pisang Segar	Paria Segar	Paprika Segar	Jeruk Segar	Tomat Segar	Apel Busuk	Pisang Busuk	Paria Busuk	Paprika Busuk	Apel segar	Tomat Busuk
Apel segar	105	0	0	0	0	0	0	0	0	0	0	0
Pisang Segar	0	80	0	0	0	0	0	0	0	0	0	0
Paria Segar	0	0	24	0	0	0	0	0	1	0	0	0
Paprika Segar	0	0	0	51	0	0	0	0	0	0	0	0
Jeruk Segar	2	0	0	0	71	0	1	0	0	0	0	0
Tomat Segar	0	0	0	0	0	50	0	0	0	0	0	0
Apel Busuk	6	0	0	0	0	0	112	0	0	0	0	0
Pisang Busuk	0	0	0	0	0	0	0	74	0	0	0	0
Paria Busuk	0	0	0	0	0	0	0	0	21	0	0	0
Paprika Busuk	0	0	0	0	0	0	0	0	0	46	0	0
Jeruk Busuk	7	0	0	0	3	0	8	0	0	0	63	0
Tomat Busuk	0	0	0	0	0	3	0	0	0	0	0	47

TABEL XIII  
AKURASI DARI KELIMA MODEL

Model	Akurasi
Model A	98
Model B	97
Model C	90
Model D	96
Model E	96

MACRO AVERAGE DARI KELIMA MODEL

Model	F1-score Macro Average
Model A	98
Model B	97
Model C	90
Model D	97
Model E	96

Pada penelitian ini digunakan *metric macro average* dengan nilai *f1-score* sebagai nilai pembanding dikarenakan nilai *f1-score* merupakan hasil kalkulasi dari nilai *precision* dan nilai *recall* dari kelas yang diklasifikasikan. Tabel XIV menampilkan nilai *macro average f1-score* dari kelima model dan kita dapat menghitung nilai *macro average f1-score* dengan menggunakan persamaan (5).

TABEL XIV

Dapat dilihat bahwa nilai *macro average* tertinggi dicapai oleh model A yang dilatih dengan dasar arsitektur ResNet50 dan data yang diaugmentasi sebelumnya dengan hasil *macro average* untuk nilai *precision*, *recall* dan *f1-score* masing-masing sebanyak 98%.

#### IV. KESIMPULAN DAN SARAN

##### A. Kesimpulan

Berdasarkan hasil dari seluruh langkah yang dilakukan pada

penelitian ini, maka kita dapat membuat sebuah kesimpulan bahwa model yang dilatih dengan data yang telah diaugmentasi sehingga memiliki jumlah data yang relatif seimbang menghasilkan performa yang lebih tinggi dibandingkan model yang dilatih dengan dasar arsitektur yang sama dengan data yang tidak diaugmentasi. Hasil dari pelatihan dan evaluasi terhadap ke lima model menunjukkan bahwa model-model tersebut dapat melakukan proses klasifikasi dengan performa yang baik, dimana model A memiliki hasil performa terbaik dibandingkan dengan model lainnya dilihat dari *macro average* dari *metric*, *precision*, *recall*, dan *f1-score* hasil evaluasi kelima model. Pada penelitian terjadi peningkatan performa model baik dari akurasi, *macro average precision*, *macro average recall* dan *macro average f1-score* sesudah menggunakan data yang telah diaugmentasi sebesar 1-2% untuk evaluasi pada model A dan model B.

### B. Saran

Dalam penelitian di bidang *artificial intelligence* khususnya di bidang *deep learning*, diharapkan dapat melakukan penambahan data apabila terdapat data-data baru yang dapat digunakan untuk proses pelatihan model, dan melakukan penyeimbangan terhadap data yang digunakan agar model nantinya tidak condong akurat hanya terhadap kelas-kelas tertentu.

### V. KUTIPAN

- [1] J. A. Barry, S. Mollan, M. A. Burdon, M. Jenkins, and A. K. Denniston, "Development and validation of a questionnaire assessing the quality of life impact of Colour Blindness (CBQoL)," *BMC Ophthalmol.*, vol. 17, no. 1, 2017, doi: 10.1186/s12886-017-0579-z.
- [2] J. S. Han *et al.*, "Knowledge and Expectations of Hearing Aid Apps among Smartphone Users and Hearing Professionals: Cross-sectional Survey," *JMIR mHealth uHealth*, vol. 10, no. 1, 2022, doi: 10.2196/27809.
- [3] G. E. Lancioni *et al.*, "People with intellectual and visual disabilities access basic leisure and communication using a smartphone's Google Assistant and voice recording devices," *Disabil. Rehabil. Assist. Technol.*, 2020, doi: 10.1080/17483107.2020.1836047.
- [4] M. Sholihin, "Identifikasi Kesegaran Ikan Berdasarkan Citra Insang dengan Metode Convolution Neural Network," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 3, 2021, doi: 10.35957/jatisi.v8i3.939.
- [5] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, E. T. Prabowo, and A. I. Ferdiansyah, "Perbandingan Convolution Neural Network Untuk Klasifikasi Kesegaran Ikan Bandeng Pada Citra Mata," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, 2021, doi: 10.25126/jtiik.2021834369.
- [6] K. L. Tsai, B. Y. Liao, Y. M. Hung, G. J. Yu, and Y. C. Wang, "Development of smart pillbox using 3D printing technology and convolutional neural network image recognition," *Sensors Mater.*, vol. 32, no. 5, pp. 1907–1912, May 2020, doi: 10.18494/SAM.2020.2632.
- [7] F. Lei, X. Liu, Q. Dai, and B. W. K. Ling, "Shallow convolutional neural network for image classification," *SN Appl. Sci.*, vol. 2, no. 1, Jan. 2020, doi: 10.1007/s42452-019-1903-4.
- [8] J. Gorzelany, J. Belcar, P. Kuźniar, G. Niedbała, and K. Pentoś, "Modelling of Mechanical Properties of Fresh and Stored Fruit of Large Cranberry Using Multiple Linear Regression and Machine Learning," *Agric.*, vol. 12, no. 2, Feb. 2022, doi: 10.3390/agriculture12020200.
- [9] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.ict.2020.04.010.
- [10] A. Ahmad Hania, "Mengenal Artificial Intelligence, Machine Learning, & Deep Learning," *J. Teknol. Indones.*, vol. 1, no. June, pp. 1–6, 2017, [Online]. Available: <https://amt-it.com/mengenal-perbedaan-artificial-intelligence-machine-learning-deep-learning/>
- [11] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable Machine Learning for Scientific Insights and Discoveries," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2976199.
- [12] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, 2021, doi: 10.1007/s12525-021-00475-2.
- [13] S. K. Behera, A. K. Rath, and P. K. Sethy, "Maturity status classification of papaya fruits based on machine learning and transfer learning approach," *Inf. Process. Agric.*, vol. 8, no. 2, pp. 244–250, Jun. 2021, doi: 10.1016/j.inpa.2020.05.003.
- [14] L. C. Chu *et al.*, "Application of Deep Learning to Pancreatic Cancer Detection: Lessons Learned From Our Initial Experience," *J. Am. Coll. Radiol.*, vol. 16, no. 9, 2019, doi: 10.1016/j.jacr.2019.05.034.
- [15] F. Lu, L. He, S. You, X. Chen, and Z. Hao, "Identifying Surface BRDF from a Single 4-D Light Field Image via Deep Neural Network," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, 2017, doi: 10.1109/JSTSP.2017.2728001.
- [16] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
- [17] T. Halimah Khoirunisa, "Implementasi web service untuk handwriting recognition dengan rest api," pp. 7–16, 2018, [Online]. Available: <http://repository.itelkom-pwt.ac.id/5425/>
- [18] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," 2016.
- [19] O. Alsing, "Mobile Object Detection using TensorFlow Lite and Transfer Learning TT - Objektivgenkänning i mobila enheter med TensorFlow Lite (swe)," 2018.
- [20] Google, "TensorFlow Lite guide," *TensorFlow*, 2020.



Penulis Bernama lengkap Rheza Timothy Tedjo. Lahir di Manado, 10 Mei 2001. Penulis menempuh Pendidikan Sekolah Dasar di SD Katolik 02 Sta. Theresia Manado, lalu melanjutkan Sekolah Menengah Pertama di SMP Katolik PAX Christi Manado dan Sekolah Menengah Atas di SMA Katolik REX Mundi Manado. Kemudian penulis meneruskan pendidikannya di salah satu Universitas yang ada di Manado untuk mengambil gelar Strata-1 yaitu di Universitas Sam Ratulangi dengan Jurusan Teknik Elektro Program Studi Informatika. Dalam memenuhi syarat penulis melakukan ujian untuk pengambilan gelar tersebut dengan judul "Rancang Bangun Aplikasi Pendeteksi Kesegaran Bahan Makanan Berbasis Machine Learning" dengan bimbingan ke dua dosen yaitu; Alwin Melkie Sambul, S.T., M.Eng, Ph.D dan Ir. Arie S. M. Lumenta, ST., MT. Selama perkuliahan penulis tergabung dalam komunitas UNITY sebagai Koordinator Mobile Development.