

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/370615767>

# Pengembangan Sistem Pendeteksi Jenis Sayuran dengan Metode CNN Berbasis Android

**Article** in Jurnal Informatika dan Rekayasa Perangkat Lunak · March 2023

DOI: 10.36499/jinrpl.v5i1.7833

---

CITATIONS

0

---

READS

270

2 authors, including:



**Budi Hartono**

Universitas Stikubank Semarang

12 PUBLICATIONS 13 CITATIONS

SEE PROFILE

## Pengembangan Sistem Pendeteksi Jenis Sayuran dengan Metode CNN Berbasis Android

Rere Setiyo Budiawan<sup>1\*</sup>, Budi Hartono<sup>2</sup>

<sup>1,2</sup> Jurusan Teknik Informatika, Fakultas Teknologi Informasi dan Industri,  
Universitas Stikubank

\*Email: setiyorere01@gmail.com

### Abstrak

Sayuran merupakan bahan pangan asal tumbuhan yang dapat dikonsumsi dalam keadaan segar dan memiliki berbagai manfaat bagi kesehatan. Namun, tidak sedikit masyarakat yang tidak mengetahui jenis sayuran dan akan kesulitan ketika mencari sayuran yang diinginkan. Penelitian ini memiliki tujuan agar masyarakat dapat dimudahkan untuk mencari sayuran dengan melakukan klasifikasi. Peneliti mengembangkan model dengan menggunakan metode Convolutional Neural Network (CNN) dengan jumlah 15 datasets dengan total 3000 data gambar. Peneliti melakukan training datasets dengan 3 macam Epoch, diantaranya adalah 20 Epoch, 50 Epoch, dan 100 Epoch. Training tersebut menghasilkan accuracy dan training loss, dengan akurasi tertinggi dimiliki Epoch 50, dan Epoch 100 dan tingkat training loss paling rendah dimiliki Epoch 100 dengan jumlah sebesar 0.609. Namun setelah model dilakukan deployment, hasil akurasi yang didapatkan tidak sebesar dengan pengujian yang dilakukan pada Google Colab. Pengujian dilakukan pada beberapa objek, diantaranya adalah wortel mendapatkan akurasi 69%, kubis mendapatkan akurasi 53%, dan pepaya mendapatkan akurasi 82%. Perbedaan hasil akurasi kemungkinan disebabkan oleh objek yang kurang identik dengan datasets atau bisa juga disebabkan oleh model yang belum sempurna. Walaupun begitu, aplikasi ini sudah dapat digunakan untuk melakukan klasifikasi jenis sayuran.

**Kata kunci:** Android, Convolutional Neural Network (CNN), Epoch, Google Colab, Sayuran

### Abstract

Vegetables are foodstuffs of plant origin that can be consumed fresh and have various health benefits. However, not a few people do not know the types of vegetables and will find it difficult to find the vegetables they want. This research aims to make it easier for people to find vegetables by classifying them. Researchers developed a model using the Convolutional Neural Network (CNN) method with a total of 15 datasets with a total of 3000 image data. Researchers conducted training datasets with 3 types of epochs, including 20 epochs, 50 epochs and 100 epochs. The training produces accuracy and training loss, with the highest accuracy belonging to Epoch 50 and Epoch 100 and the lowest level of training loss is owned by Epoch 100 with a total of 0.609. However, after the model was deployed, the accuracy results obtained were not as high as the tests conducted on Google Colab. Tests were carried out on several objects, including carrots with an accuracy of 69%, cabbage with an accuracy of 53%, and papaya with an accuracy of 82%. The difference in accuracy results may be caused by objects that are less identical to the datasets or can also be caused by imperfect models. Even so, this application can already be used to classify types of vegetables.

**Keywords:** Android, Convolutional Neural Network (CNN), Epoch, Google Colab, Vegetables

### PENDAHULUAN

Perkembangan teknologi di masa sekarang ini memiliki berbagai pengaruh dalam jalannya kehidupan. Banyak dampak positif dan negatif terkait dengan adanya perkembangan teknologi. Peran teknologi banyak membantu masyarakat untuk melakukan segala pekerjaan agar dapat dilakukan dengan mudah dan efektif, sehingga dapat menghemat waktu masyarakat ketika melakukan pekerjaan.

Sayuran merupakan bahan pangan asal tumbuhan yang mengandung kadar air tinggi dan dikonsumsi dalam keadaan segar atau setelah diolah secara minimal (Akbar et al., 2020). Terdapat berbagai jenis sayuran yang memiliki warna, rasa, aroma dan tekstur yang berbeda-beda. Namun secara umum bila ditinjau dari segi gizinya, sayuran merupakan sumber mineral, serat, dan vitamin terutama vitamin A, dan vitamin C (Nurainy, 2018). Sayuran menjadi salah satu bahan makanan

yang sering digunakan dalam masakan. Segala jenis masakan, banyak yang dikombinasikan dengan sayuran untuk membuat cita rasanya menjadi lebih nikmat. Namun tidak sedikit masyarakat dari kalangan muda yang tidak mengetahui akan nama ataupun jenis dari sayuran. Sehingga ketika mereka memiliki kebutuhan untuk mencari sayuran atau mungkin memiliki resep masakan, mereka akan kesulitan untuk mencari sayuran yang dibutuhkan. Karena mereka tidak tau bagaimana bentuk dari sayuran yang dicari tersebut.

Penelitian ini memiliki tujuan agar masyarakat dapat terbantu untuk mengidentifikasi sayuran yang dibutuhkan. Dengan adanya berbagai jenis sayuran yang ada saat ini, maka diperlukan adanya pendekatan teknologi untuk mengklasifikasi jenis sayuran tersebut. Terdapat berbagai teknologi yang dapat digunakan, terutama pada bidang *Machine Learning* yang memiliki berbagai *Neural Networks* yang akan dimanfaatkan untuk membuat model deteksi, guna mengidentifikasi suatu gambar dari jenis sayuran. Diantaranya adalah RNN (*Reccurent Neural Network*), GAN (*Generative Adversial Network*), dan CNN (*Convolutional Neural Network*).

CNN (*Convolutional Neural Network*) merupakan pengembangan dari *Artificial Neural Network* (ANN) yang memanfaatkan *Single Layer Perception* untuk menyelesaikan operasi logika yang kompleks (Pangestu & Bunyamin, 2018). CNN (*Convolutional Neural Network*) memiliki kemampuan klasifikasi yang diperuntukkan untuk pengolahan data gambar dengan proses klasifikasi data yang sudah diberi label dan menggunakan metode supervised learning yang mengelompokkan suatu data ke data yang sudah ada (Harani et al., 2019). CNN (*Convolutional Neural Network*) terdiri dari dua tahapan utama, yaitu *feature learning* dan *classification*. Pada tahapan *feature learning* terdiri dari *convolution layer*, ReLU (*activation function*) dan *pooling layer*. Sedangkan pada tahap *classification* terdiri dari *flatten*, *fully-connected layer*, dan *prediction*. Pada setiap bagian CNN (*Convolutional Neural Network*) terdapat dua proses utama, yaitu *feed-forward* dan *backpropagation* (Yusuf et al., 2019). *Datasets* yang dilakukan *preprocessing* akan memiliki akurasi yang lebih tinggi dibandingkan tanpa dilakukan *preprocessing*

(Alfikri et al., 2022). Model yang dikembangkan nantinya akan dilakukan *deployment* ke Android. Proses *deployment* dilakukan di Android karena Android merupakan salah satu *platform* yang penggunaanya terus meningkat dari tahun ke tahun. Pada Akhir 2020 Indonesia masuk empat besar pengguna smartphone terbesar di dunia setelah China, India, dan Amerika Serikat (Sanjaya, 2022).

## TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian (Yusuf et al., 2019) melakukan klasifikasi emosi berdasarkan pada ciri wajah dengan metode *Convolutional Neural Network* (CNN). Data yang digunakan diperoleh dari *Extended Cohn-Kanade Datasets* (CK+) dari 210 subjek dengan total 327 citra. Hasil pengujian menunjukkan bahwa nilai *learning rate* tertentu melatih arsitektur lebih baik dari yang lain. Akurasi terbaik pada penelitian ini adalah 86.4% dan akurasi rata-rata adalah 80.7%. Perbedaan dengan penelitian yang dilakukan adalah pada *datasets* menggunakan citra lebih banyak, yaitu 3000 citra yang diperoleh dari Kaggle.

Penelitian (Fitriani, 2021) melakukan klasifikasi jenis bunga dengan metode *Convolutional Neural Network* (CNN). *Datasets* yang digunakan memiliki jumlah 140 citra bunga dengan hasil akurasi mencapai 0.5. Perbedaan dengan penelitian yang dilakukan adalah pada hasil akurasi. Penelitian yang dilakukan peneliti memiliki hasil akurasi mencapai 100%.

Penelitian (Yohannes et al., 2020) melakukan klasifikasi jenis buah dan sayuran menggunakan SVM (*Support Vector Machine*). *Datasets* yang digunakan terdiri dari 114 gambar sebagai data *training* dan 30 gambar sebagai data *testing*. Hasil akhir pengujian, fitur *Saliency-HOG* dan *Color Moments* mampu memberikan hasil yang baik dengan akurasi 98.57%, 98.55%, dan 99.120%. Perbedaan dengan penelitian yang dilakukan adalah pada metode klasifikasi. CNN memiliki hasil akurasi yang lebih maksimal dibandingkan dengan SVM.

Penelitian (Erwin et al., 2021) melakukan deteksi hama ulat pada tanaman dengan menggunakan CNN (*Convolutional Neural Network*). *Datasets* diambil menggunakan

modul kamera *Raspberry Pi*, kemudian citra diubah menjadi ukuran 128x128 piksel. Hasil akhir dari penelitian ini memiliki akurasi 89% dari 5110 citra. Perbedaan dengan penelitian yang dilakukan adalah pada penelitian (Erwin dkk., 2021) masih melakukan pengambilan *datasets* secara manual dengan menggunakan modul kamera *Raspberry Pi*. Sedangkan yang dilakukan peneliti menggunakan *datasets* bersumber dari Kaggle untuk jenis sayuran yang akan dilakukan klasifikasi.

## 2.2 Sayuran

Sayuran adalah sebutan untuk bahan pangan yang berasal dari tumbuhan dan memiliki kadar air tinggi yang dapat dikonsumsi dalam keadaan segar. Secara umum sayuran merupakan tanaman hortikultura yang memiliki umur relatif pendek dibandingkan dengan tanaman buah-buahan. Terdapat banyak jenis sayuran yang memiliki warna, rasa, aroma, dan tekstur yang berbeda-beda. Namun, secara umum jika dilihat dari segi gizinya, sayuran merupakan sumber mineral, sumber serat, dan sumber vitamin (Nurainy, 2018).

## 2.3 Convolutional Neural Network (CNN)

*Convolutional Neural Network* yang biasa dikenal dengan sebutan CNN adalah pengembangan dari *Multilayer Perception* (MLP) untuk mengolah data dua dimensi. Sedangkan *Multilayer Perception* (MLP) sendiri, merupakan pengembangan dari *Artificial Neural Network* (ANN) yang memanfaatkan *Single Layer Perception* untuk menyelesaikan operasi logika yang kompleks (Pangestu & Bunyamin, 2018). *Convolutional Neural Network* memiliki kemampuan klasifikasi yang diperuntukkan untuk pengolahan data gambar dengan proses klasifikasi data yang sudah diberi label dan menggunakan metode *supervised learning* yang mengelompokkan suatu data ke data yang sudah ada (Harani et al., 2019).

CNN (*Convolutional Neural Network*) terdiri dari dua tahapan utama, yaitu *feature learning* dan *classification*. Pada tahapan *feature learning* terdiri dari *convolution layer*, ReLU (*activation function*) dan *pooling layer*. Sedangkan pada tahap *classification* terdiri dari *flatten*, *fully-connected layer*, dan *prediction*. Pada setiap bagian CNN (*Convolutional Neural Network*) terdapat dua proses utama, yaitu *feed-*

*forward* dan *backpropagation* (Yusuf et al., 2019). *Feedforward* merupakan tahap pertama yang mengenalkan beberapa *layers* untuk melakukan proses klasifikasi *datasets* yang menggunakan *updated weights* dan *biases* dari proses *backpropagation*. Sedangkan *Backpropagation* merupakan tahap kedua yang memiliki hasil dari proses *feedforward* dan dapat terlacak dari *output layer* sampai *first layer*. Indikasi dari data yang telah terlacak adalah dengan didapatkannya *weights* dan *bias* dengan *value* yang baru (Alfikri et al., 2022).

## 2.4 Android

Android merupakan sistem operasi berbasis Linux yang *open source*, sehingga memungkinkan *developers* untuk dapat membuat aplikasi sesuai keinginan mereka sendiri (Alfikri et al., 2022). Platform ini awalnya didirikan oleh Android Inc. Kemudian diakuisisi oleh Google dan dirilis AOSP (*Android Open Source Project*) di tahun 2007. Pengumuman ini disertai dengan pendanaan dari OHA (*Open Handset Alliance*), sebuah kongsi yang mendedikasikan dirinya untuk mengembangkan dan mendistribusi sistem operasi Android. Platform Android merupakan platform muda dengan proses pengembangan yang cukup cepat. Tiap versi rilis dari sistem operasi Android memiliki nama sesuai dengan urutan *Alphabet* (Gilski & Stefanski, 2015).

## METODE PENELITIAN

Penulis menggunakan beberapa teknik dalam melaksanakan penelitian ini, diantaranya adalah sebagai berikut.

### a. Studi Pustaka

Dalam melakukan penelitian mengenai pengembangan aplikasi pendeteksi jenis sayuran dengan menggunakan metode CNN (*Convolutional Neural Network*) berbasis Android, peneliti membutuhkan berbagai sumber informasi yang relevan, seperti jurnal, artikel, tesis, maupun buku yang terkait. Sumber informasi yang peneliti pelajari, tidak lain adalah yang memiliki keterkaitan dengan pengembangan aplikasi Android Native, Machine Learning, dan CNN (*Convolutional Neural Network*).

### b. Pengumpulan Datasets

Pada penelitian ini, langkah pertama yang dilakukan peneliti adalah melakukan pengumpulan *datasets* yang berupa kumpulan gambar sayuran. *Datasets* tersebut nantinya akan digunakan untuk pengembangan model yang menggunakan metode CNN (*Convolutional Neural Network*). Sumber yang digunakan untuk *datasets* sayuran adalah berasal dari *website* Kaggle, yang merupakan *website* yang memiliki berbagai *open datasets* yang dapat digunakan dengan bebas. *Datasets* yang digunakan peneliti, tepatnya ada pada url berikut

<https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset>.

### c. Training Datasets

Pada bagian ini dilakukan proses training *datasets*, dimana pada proses ini *datasets* dilatih dengan tujuan agar dapat memperoleh akurasi tinggi dari klasifikasi yang dilakukan. *Datasets* dilakukan proses training dengan menggunakan 3 macam Epoch, yaitu adalah 20 Epoch, 50 Epoch, dan 100 Epoch.

```
[ ] datasets = DataLoader.from_folder(ds_path)

[ ] training, rest = datasets.split(0.8)
    validation, testing = rest.split(0.5)
```

Gambar 1. Splitting Process

Dalam Gambar 1, merupakan pembagian proses antara training, validation, dan testing. Proses pertama yaitu adalah melakukan Loading atau memuat *datasets* yang sebelumnya telah diproses ke dalam folder pada variabel *ds\_path*. Setelah selesai dimuat, diatur sebuah proses untuk penggunaan *datasets*. Diantaranya adalah training yang diatur memiliki penggunaan 80% untuk *datasets*, dan sisanya digunakan untuk proses validation, dan testing.

```
[ ] efficientnet_model = model_spec.get("efficientnet_lite1")

[ ] model = image_classifier.create(training,
    epochs=20,
    validation_data=validation,
    use_augmentation=True,
    shuffle=True,
    model_spec=efficientnet_model)
```

Gambar 2. Training Datasets

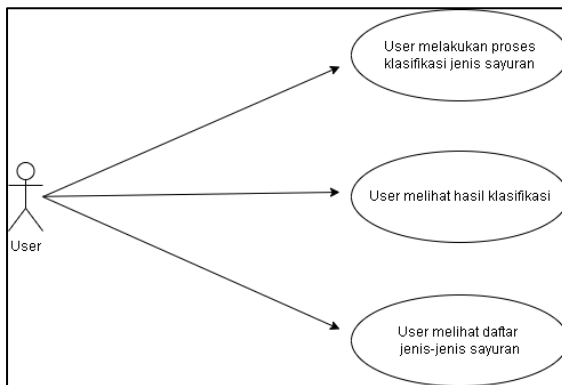
Pada Gambar 2, merupakan proses implementasi yang digunakan untuk training *datasets*. Sebelum proses *training* dilakukan, diatur sebuah *pre-trained model* dengan jenis *efficientnet\_lite1*. Penggunaan *pre-trained model* dilakukan dengan tujuan agar mendapatkan hasil akurasi yang lebih baik. Tahap selanjutnya merupakan proses pembuatan *training*, dengan memanfaatkan fungsi *create* yang memiliki beberapa parameter. Dalam beberapa parameter tersebut diatur sebuah *epoch* dengan jumlah 20, hal ini berarti sistem akan melakukan *training* untuk *datasets* sebanyak 20 kali. Penggunaan *Epoch* sebanyak 20 merupakan percobaan pertama. Percobaan selanjutnya adalah menggunakan *Epoch* 50, dan 100 untuk mengetahui *Epoch* mana yang akan menghasilkan model dengan akurasi tertinggi. Parameter selanjutnya diatur untuk penggunaan *augmentation*, *shuffle*, dan *model\_spec* untuk *pre-trained model*.

### d. Diagram Sistem

Dalam melaksanakan penelitian ini, diperlukan perancangan sistem guna untuk memberi rancangan pada sistem yang akan dibuat. Metode yang digunakan dalam melakukan perancangan sistem aplikasi pendeteksi jenis sayuran berbasis Android adalah dengan menggunakan UML (*Unified Modelling Language*) yang merupakan metode permodelan berorientasi objek. Perancangan sistem pada penelitian ini yaitu menggunakan *Use Case Diagram* dan *Sequence Diagram*.

#### 1. Use Case Diagram

*Use Case Diagram* digunakan untuk menggambarkan fungsi dasar dalam sebuah sistem dengan cara mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya. *Use Case Diagram* terdiri dari beberapa *element*, diantaranya adalah *Actor*, *Use Case*, *Subject Boundary*, *Association Relationship*, *Include Relationship*, *Extend Relationship*, dan *Generalization Relationship* (Indriyani et al., 2019). Dibawah ini merupakan *Use Case Diagram* yang digunakan pada sistem.

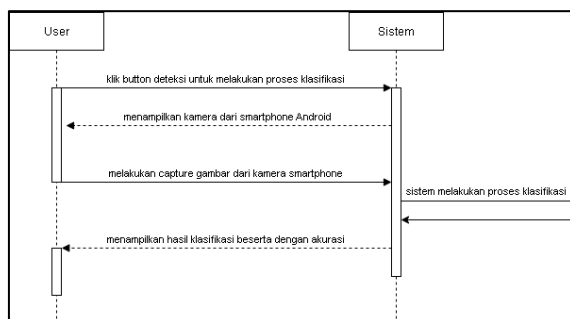


Gambar 3. Use Case Diagram

Gambar 3 menampilkan *Use Case Diagram* dari aplikasi pendeteksi jenis sayuran berbasis Android. *Use Case Diagram* tersebut memiliki 1 aktor, yaitu adalah *user* dan memiliki 3 hak ketika mengakses sistem aplikasi. Hak-hak yang dimiliki oleh *user* diantaranya adalah melakukan proses klasifikasi jenis sayuran, melihat hasil klasifikasi, dan melihat daftar jenis-jenis sayuran.

## 2. Sequence Diagram

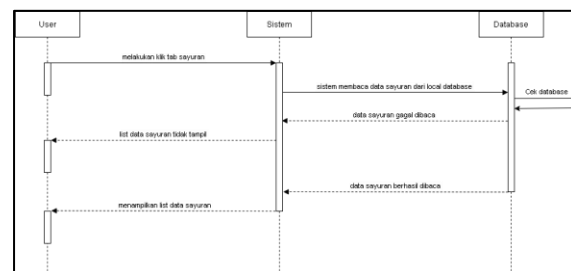
*Sequence Diagram* memiliki tujuan untuk menunjukkan pesan diantara objek untuk *use case* tertentu dari waktu ke waktu. *Sequence Diagram* mengilustrasikan objek-objek yang berpartisipasi di dalam suatu *use case*. Elemen-elemen yang dimiliki oleh *Sequence Diagram* diantaranya adalah *Actor*, *Object*, *Lifeline*, *Execution Occurrence*, *Message*, *Guard Condition*, *Object Destruction*, dan *Frame* (Indriyani dkk., 2019). Gambar 4 merupakan *Sequence Diagram* yang digunakan pada sistem.



Gambar 4. Sequence Diagram Deteksi Jenis Sayuran

Pada Gambar 4, menampilkan *Sequence Diagram* dari sistem pendeteksi jenis

sayuran. Langkah awal dalam *Sequence Diagram* tersebut yaitu *user* melakukan klik *button* deteksi untuk menggunakan sistem klasifikasi jenis sayuran. Kemudian sistem akan menampilkan fitur kamera dari *smartphone* Android, dan *user* harus mengarahkan kamera ke objek sayuran. Pada saat kamera diarahkan ke objek sayuran, *user* harus menekan *button capture* untuk menangkap gambar sayuran. Setelah itu, hasil klasifikasi akan muncul dengan persentase akurasi dan jenis sayuran yang dilakukan deteksi.



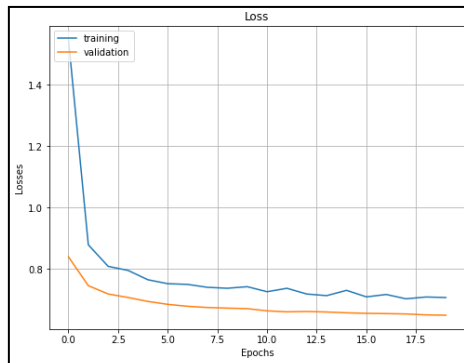
Gambar 5. Sequence Diagram List Daftar Sayuran

Pada Gambar 5, menampilkan *Sequence Diagram* dari halaman sayuran yang mempunyai tampilan list daftar sayuran. Langkah awal dari *Sequence Diagram* tersebut yaitu *user* melakukan klik pada tab sayuran. Kemudian sistem akan membuka halaman sayuran, dan membaca data sayuran dari local database. Terdapat 2 kondisi pada pengecekan database tersebut. Kondisi pertama yaitu jika data sayuran gagal dibaca, maka list data sayuran tidak akan tampil. Sedangkan, kondisi kedua adalah ketika data sayuran berhasil dibaca, sistem akan menampilkan list data sayuran ke halaman sayuran.

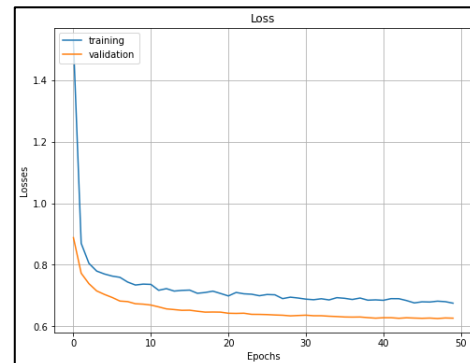
## HASIL DAN PEMBAHASAN

### 4.1 Hasil Pengujian Epoch

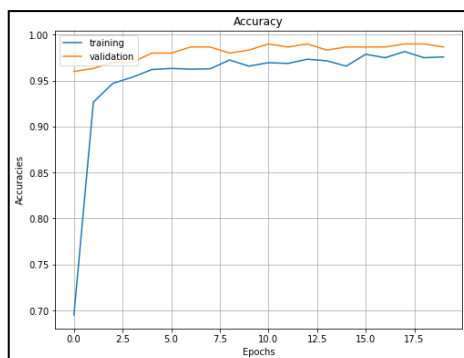
Pada bagian ini dijelaskan mengenai hasil pengujian training datasets dengan *Epoch* 20, 50, dan 100. Pengujian ini dilakukan dengan tujuan untuk mengetahui perbedaan hasil training datasets dengan jumlah *Epoch* yang berbeda. Berikut merupakan hasil dari beberapa training *datasets*.



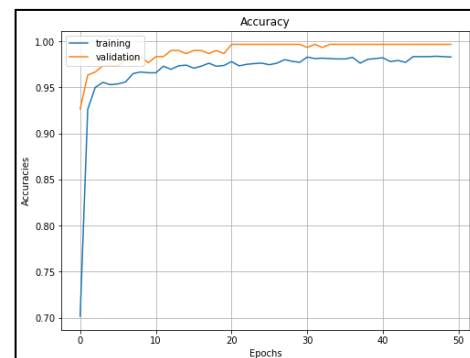
Gambar 6. Training Loss Sistem Epoch 20



Gambar 9. Training Loss Sistem Epoch 50

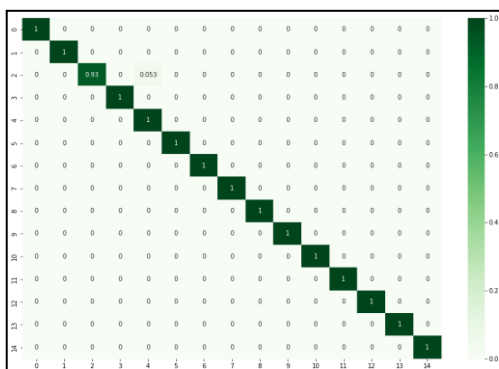


Gambar 7. Training Accuracy Sistem Epoch 20



Gambar 10. Training Accuracy Sistem Epoch 50

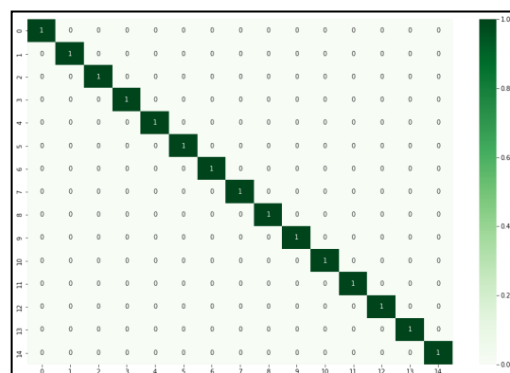
Pada Gambar 6 dan 7, merupakan *training loss* dan *training accuracy* dari sistem Epoch 20. Dari hasil pengujian tersebut, dapat diketahui bahwa sistem mendapatkan *training loss* sebesar 0.635 dan *training accuracy* sebesar 0.996.



Gambar 8. Confusion Matrix Sistem Epoch 20

Pada Gambar 8, dapat diketahui bahwa indeks ke 2 dari *datasets* mendapatkan *training loss* sebesar 0.053 dan *training accuracy* sebesar 0.93. Hal ini berarti bahwa sistem belum dapat melakukan klasifikasi dengan akurat. Maka dari itu dilakukan training kembali dengan menggunakan Epoch yang lebih besar, yaitu Epoch 50.

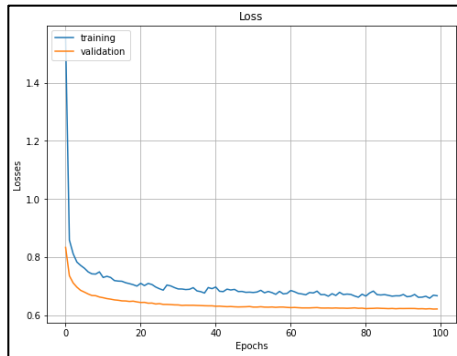
Pada Gambar 9 dan 10, merupakan *training loss* dan *training accuracy* yang dihasilkan dari pengujian sistem Epoch 50. Dari hasil pengujian tersebut, dapat diketahui bahwa sistem mendapatkan *training loss* sebesar 0.621, dan *training accuracy* sebesar 1.000. *Training loss* yang dihasilkan dapat dikatakan berkurang dari pengujian Epoch 20 sebelumnya yang mendapatkan 0.635 *training loss*.



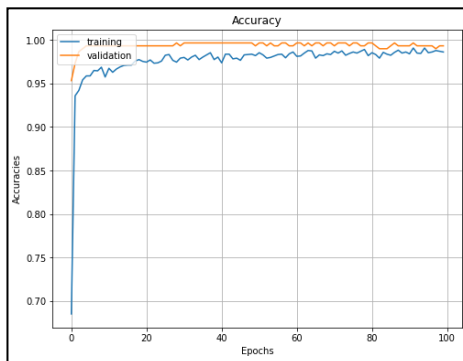
Gambar 11. Confusion Matrix Sistem Epoch 50

Pada Gambar 11, dapat diketahui bahwa semua *datasets* mendapatkan tingkat akurasi sebesar 1 atau setara dengan 100%. Namun, pada hasil pengujian sistem masih mendapatkan *training loss* sebesar 0.621.

Maka dari itu, akan dilanjutkan kembali *training datasets* dengan menggunakan *Epoch* 100.

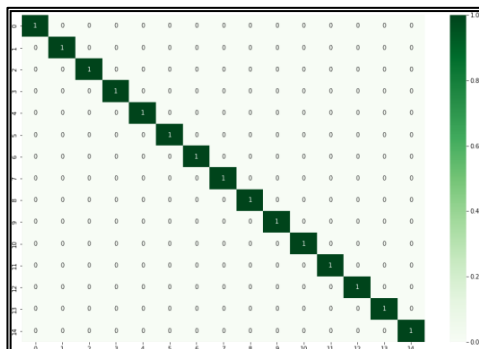


Gambar 12. Training Loss Sistem Epoch 100



Gambar 13. Training Accuracy Sistem Epoch 100

Pada Gambar 12 dan 13, merupakan *training loss* dan *training accuracy* yang dihasilkan dari pengujian sistem *Epoch* 100. Dari hasil pengujian tersebut, dapat diketahui bahwa sistem mendapatkan *training loss* sebesar 0.609 dan *training accuracy* sebesar 1.000. Hasil pengujian ini tergolong cukup bagus jika dibandingkan dengan sebelumnya yang menggunakan *Epoch* 20 dan *Epoch* 100.



Gambar 14. Confusion Matrix Sistem Epoch 100

Pada Gambar 14, dapat diketahui bahwa semua *datasets* mendapatkan akurasi sebesar 1. Hal ini dapat disimpulkan bahwa sistem dapat melakukan klasifikasi dengan akurat. Dengan ini, sistem *Epoch* 100 akan dilakukan *export* menjadi model Tensorflow Lite dan akan dikonsumsi dalam project aplikasi berbasis Android yang dikembangkan.

#### 4.2 Hasil Pengujian dan Analisis

Pada bagian ini, akan dijelaskan mengenai hasil pengujian dan analisis dari pengujian yang telah dilakukan untuk sistem *Epoch* 20, *Epoch* 50, dan *Epoch* 100.

Tabel 1. Keterangan *Datasets*

Jumlah Datasets	Rasio Gambar
3000	224 x 224

Tabel 2. Hasil Pengujian Sistem

Epoch	Accuracy (%)	Training Loss
20	99%	0.635
50	100 %	0.621
100	100 %	0.609

Berdasarkan dari Tabel 1, menjelaskan bahwa dengan menggunakan *datasets* sayuran sebanyak 3000 gambar, dengan rasio dimensi gambar sebesar 224 x 224 piksel, dapat menghasilkan akurasi dari training sampai dengan 100% dan *training loss* sebesar 0.609. Pengujian *training datasets* tersebut dilakukan dengan 3 macam *Epoch*. Diantaranya adalah 20 *Epoch* dengan hasil akurasi sebesar 99% dan menghasilkan *training loss* sebesar 0.635. Kemudian dengan 50 *Epoch* menghasilkan akurasi sebesar 100% dan *training loss* sampai dengan 0.621. Terakhir adalah dengan menggunakan 100 *Epoch*, menghasilkan tingkat akurasi sampai dengan 100% dan *training loss* sebesar 0.609.

Kesimpulan dari beberapa pengujian *Epoch* yang telah dilakukan adalah dengan perbedaan jumlah *Epoch* ketika dilakukan training, sangat mempengaruhi hasil akurasi yang dihasilkan oleh sistem. Semakin besar *Epoch* yang digunakan ketika *training*, maka semakin besar pula tingkat akurasi yang



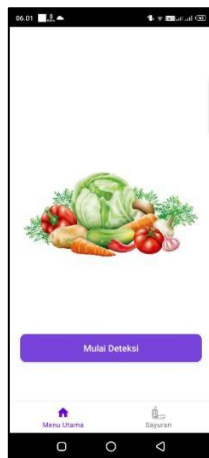
dihasilkan. Begitu pula dengan training loss, semakin besar *Epoch* yang digunakan, maka semakin kecil tingkat training loss yang dihasilkan. Model yang baik, harus memiliki tingkat akurasi yang tinggi, dan training loss yang rendah agar dapat melakukan klasifikasi dengan benar ketika dilakukan *deployment* pada suatu *platform*.

### 4.3 Hasil Pengembangan Aplikasi Android

Selain hasil pengembangan model *Convolutional Neural Network* (CNN), pengembangan aplikasi Android juga telah berhasil dilakukan dengan *deployment* model yang telah dikembangkan. Berikut merupakan tampilan user interface yang telah dihasilkan dari proses pengembangan sistem Android.

#### 4.3.1 Home Screen

Dibawah ini merupakan hasil dari pengembangan *Home Screen* untuk aplikasi pendeteksi jenis sayuran berbasis Android yang dikembangkan peneliti.



**Gambar 15.** Tampilan *Home Screen*

Pada Gambar 15, menampilkan tampilan dari *Home Screen* yang telah dikembangkan. *Home Screen* memiliki satu *ImageView* yang menampilkan gambar sayuran untuk merepresentasikan aplikasi yang dipakai merupakan aplikasi dengan tema sayuran. Halaman ini juga mempunyai button dengan teks “Mulai Deteksi” yang berfungsi untuk melakukan navigasi ke halaman *Camera Screen* untuk dilakukan proses klasifikasi jenis sayuran. Kemudian, di bagian bawah terdapat *bottom tab navigation* dengan button “Menu Utama”

dan “Sayuran”. Bagian button “Menu Utama” berfungsi untuk melakukan navigasi ke halaman *Home Screen* yang mana merupakan halaman yang terbuka saat ini, sedangkan button “Sayuran” berfungsi untuk membuka halaman *Vegetables Screen*.

#### 4.3.2 Camera Screen

Pada Gambar 16 merupakan hasil dari pengembangan *Camera Screen* untuk aplikasi pendeteksi jenis sayuran berbasis Android yang dikembangkan peneliti.



**Gambar 16.** Tampilan *Camera Screen* : Test Wortel

Pada Gambar 16 menampilkan tampilan dari *Camera Screen*. Halaman ini umumnya menampilkan tampilan dari kamera *smartphone*, namun juga memiliki fungsi untuk melakukan klasifikasi jenis sayuran secara langsung. Terdapat satu button berada di bagian bawah halaman yang memiliki fungsi untuk melakukan rotasi ke kamera belakang, dan begitu pula sebaliknya. Dalam Gambar 16, juga menampilkan bahwa sistem telah melakukan proses klasifikasi pada sayuran dengan jenis wortel. Pengujian klasifikasi pertama dilakukan dengan wortel tersebut, dan menghasilkan akurasi sebesar 69%. Hal ini berarti wortel yang ditampilkan tidak akurat 100% dengan yang ada pada datasets, dan hanya akurat 69% dari datasets yang ada.



**Gambar 17.** Tampilan *Camera Screen* : Test Kubis

Pada Gambar 17 menampilkan tampilan dari *Camera Screen* dengan pengujian klasifikasi untuk jenis sayuran “Kubis”. Pengujian tersebut mendapatkan hasil akurasi sebesar 53% akurat. Hasil akurasi yang tidak sempurna ini, kemungkinan dapat disebabkan oleh objek yang tidak sama persis dengan yang ada pada *datasets*. Namun, dengan akurasi sebesar 53% dapat menunjukkan bahwa objek yang dilakukan klasifikasi adalah objek sayuran dengan jenis “Kubis”.



**Gambar 18.** Tampilan *Camera Screen* : Test Pepaya

Pada Gambar 18, menampilkan tampilan dari *Camera Screen* yang sedang melakukan klasifikasi untuk jenis sayuran “Pepaya”. Pengujian pada jenis sayuran “Pepaya” ini menghasilkan akurasi sebesar 82%, walaupun objek “Pepaya” yang dilakukan klasifikasi hanya setengah saja. Hal ini dapat diartikan bahwa objek “Pepaya” tersebut dapat

dilakukan klasifikasi dengan baik, walaupun tidak 100% akurat.

#### 4.3.1 Vegetables Screen

Pada Gambar 19 merupakan hasil dari pengembangan *Vegetables Screen* untuk aplikasi pendeteksi jenis sayuran berbasis Android yang dikembangkan peneliti.



**Gambar 19.** Tampilan *Vegetables Screen*

Pada Gambar 19, menampilkan tampilan dari *Vegetables Screen*. Data yang digunakan dari *Vegetables Screen* berasal *Room Database* yang merupakan *database local* yang digunakan peneliti. *Room Database* tersebut melakukan *loading* data dari *dummy json* yang telah dibuat oleh peneliti. *Json* tersebut dilakukan *import* ke project Android, dan dilakukan *loading* ke *Room Database* ketika *database* diinisialisasi untuk pertama kali.

#### SIMPULAN

Penelitian pengembangan aplikasi pendeteksi jenis sayuran berbasis Android berhasil dilakukan dengan menggunakan metode *Convolutiona Neural Network* (CNN). Pengembangan model deteksi dilakukan dengan menggunakan *Pre-Trained Model* dengan tujuan agar dapat mendapatkan hasil akurasi yang cukup tinggi. Jenis *Pre-Trained Model* yang digunakan peneliti adalah *Efficient-Net Lite1* dan mendapatkan hasil akurasi yang cukup tinggi dengan *Training Loss* yang cukup rendah. Diantaranya adalah 20 *Epoch* dengan hasil akurasi 99% dan *Training Loss* 0.635, 50 *Epoch* dengan hasil akurasi 100% dan *Training*

*Loss* 0.621, dan yang terakhir adalah 100 *Epoch* dengan hasil akurasi 100% dan *Training Loss* 0.609. Akurasi tertinggi dari hasil *training* yang dilakukan adalah pada *Epoch* 50 dan *Epoch* 100. Namun, tingkat *Training Loss* yang dimiliki *Epoch* 100 lebih kecil dibandingkan dengan *Epoch* 50, maka dari itu yang digunakan untuk implementasi ke Android adalah *Epoch* 100.

Setelah dilakukan *deployment* ke Android, akurasi yang dihasilkan tidak setinggi saat dilakukan *testing* di Google Colab. Beberapa pengujian telah dilakukan setelah dilakukan *deployment* ke Android. Diantaranya adalah pengujian pada objek wortel mendapatkan akurasi 69%, pengujian pada objek kubis mendapatkan akurasi 53%, dan yang terakhir adalah pengujian pada objek pepaya mendapatkan akurasi 82%. Setiap pengujian mendapatkan hasil yang berbeda-beda dan kemungkinan disebabkan oleh berbagai penyebab. Seperti objek yang tidak identik dengan yang ada pada *datasets*, atau model yang belum sempurna. Walaupun masih belum sempurna, aplikasi pendeteksi jenis sayuran yang dikembangkan peneliti sudah dapat digunakan untuk melakukan klasifikasi jenis sayuran dengan hasil yang belum maksimal. Aplikasi ini, diharapkan dapat membantu masyarakat yang pengetahuannya kurang mengenai jenis-jenis sayuran agar dapat dimanfaatkan ketika sedang membutuhkan.

## DAFTAR PUSTAKA

- Akbar, J.M. *et al.* (2020) *MENGGUNAKAN DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK*, Fakultas Teknologi Industri. Available at: <https://jurnal.wicida.ac.id/index.php/sebatik/article/view/1134> (Accessed: 5 February 2023).
- Nurainy, F. (2018) *PENGETAHUAN BAHAN NABATI I: SAYUR-SAYURAN, BUAH-BUAHAN, KACANG-KACANGAN, SEREALIA DAN UMBI-UMBIAN*. Lampung. Available at: <http://repository.lppm.unila.ac.id/id/eprint/9850> (Accessed: 5 February 2023).
- Pangestu, M.A. and Bunyamin, H. (2018) 'Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model', 4, pp. 2443–2229. Available at: <https://doi.org/10.28932/jutisi.v4i2.828>
- Harani, N.H., Prianto, C. and Hasanah, M. (2019) *Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python*, Jurnal Teknik Informatika. Available at: <https://ejurnal.poltekpos.ac.id/index.php/informatika/article/view/658> (Accessed: 5 February 2023).
- Yusuf, A., Wihandika, C.R. and Dewi, C. (2019) *Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network*. Available at: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6732> (Accessed: 5 February 2023).
- Alfikri, H.R. *et al.* (2022) *PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID*. Available at: <https://doi.org/https://doi.org/10.33365/jti.v16i2.1752>.
- Sanjaya, S. (2022) *APLIKASI PENGENALAN TINGKAT KEMATANGAN BUAH TOMAT MENGGUNAKAN FITUR WARNA HSV BERBASIS ANDROID*, Jurnal TEKNOINFO. Available at: <https://doi.org/https://doi.org/10.33365/jti.v16i1.1489>.
- Fitriani (2021) *FITRIANI-KLASIFIKASI JENIS BUNGA DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) KLASIFIKASI JENIS BUNGA DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) (CLASSIFICATION OF INTEREST TYPES USING CONVOLUTIONAL NEURAL NETWORK (CNN))*. Available at: <https://doi.org/https://doi.org/10.46764/teknime dia.v2i2.39>.
- Yohannes, Y., Pribadi, R.M. and Chandra, L. (2020) 'Klasifikasi Jenis Buah dan Sayuran Menggunakan SVM Dengan Fitur Saliency-HOG dan Color Moments', *ELKHA*, 12(2), pp. 125–131. Available at: <https://doi.org/http://dx.doi.org/10.26418/elkha.v12i2.42160>.
- Erwin, S.R.M., Putrada, G.A. and Triawan, A.M. (2021) *Deteksi Hama Ulat Pada Tanaman Selada Berbasis Aquaponic Menggunakan CNN (Convolutional Neural Network)*. Available at: <https://towardsdatascience.com/>.

Gilski, P. and Stefanski, J. (2015) *Android OS: A Review*. Available at: [www.temjournal.com](http://www.temjournal.com).

Indriyani, F. *et al.* (2019) 'Analisa Perancangan Sistem Informasi', *Repository BSI* [Preprint]. Available at: <https://repository.bsi.ac.id/index.php/repo/viewitem/22858> (Accessed: 5 February 2023).

