

**KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN  
METODE KNN (*K-NEAREST NEIGHBOR*) DI KOPERASI USAHA TANI  
GAYO KECAMATAN BEBESEB ACEH TENGAH**

**SKRIPSI**

**DISUSUN OLEH:**

**KHAIRUNNISA**

**2009010042**



**UMSU**  
Unggul | Cerdas | Terpercaya

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2024**

**KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN  
METODE KNN (*K-NEAREST NEIGHBOR*) DI KOPERASI USAHA TANI  
GAYO KECAMATAN BEBESEB ACEH TENGAH**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas  
Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah  
Sumatera Utara**

**KHAIRUNNISA**

**NPM. 2009010042**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN  
2024**

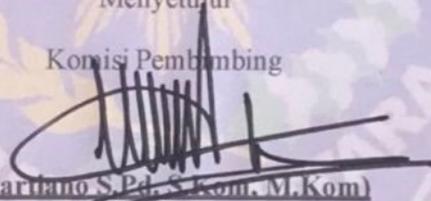
LEMBAR PENGESAHAN

Judul Skripsi : KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA  
MENGUNAKAN METODE KNN (*K-NEAREST  
NEIGHBOR*) DI KOPERASI USAHA TANI GAYO  
KECAMATAN BEBESEB ACEH TENGAH

Nama Mahasiswa : KHAIRUNNISA  
NPM : 2009010042  
Program Studi : SISTEM INFORMASI

Menyetujui

Komisi Pembimbing

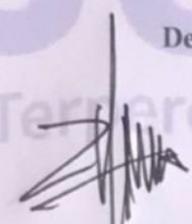
  
(Marthano S. Pd., S.Kom., M.Kom.)

NIDN. 0128029302

Ketua Program Studi

  
(Marthano S. Pd., S.Kom., M.Kom.)  
NIDN. 0128029302

Dekan

  
(Dr. Al-Khowarizmi, S.Kom., M.Kom.)  
NIDN. 0127099201

## PERNYATAAN ORISINALITAS

KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN METODE KNN  
(*K-NEAREST NEIGHBOR*) DI KOPERASI USAHA TANI GAYO KECAMATAN  
BEBESEB ACEH TENGAH

### SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Mei 2024

Yang membuat pernyataan



KHAIRUNNISA

Npm. 2009010042

**PERNYATAAN PERSETUJUAN  
PUBLIKASIKARYA ILMIAH UNTUK  
KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : KHAIRUNNISA  
NPM : 2009010042  
Program Studi : Sistem Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

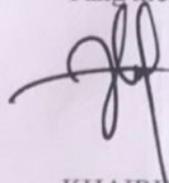
**KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN METODE  
KNN (*K-NEAREST NEIGHBOR*) DI KOPERASI USAHA TANI GAYO KECAMATAN  
BEBESEB ACEH TENGAH**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, **30** Mei 2024

Yang membuat pernyataan



KHAIRUNNISA  
NPM. 2009010042

## DAFTAR RIWAYAT HIDUP

### DATA PRIBADI

Nama Lengkap : KHAIRUNNISA Tempat  
dan Tanggal Lahir : Medan, 29 Oktober 2002

Alamat Rumah : Jl. Puskesmas I Gg.Ridho  
Telepon/Faks/HP : 082390581455

E-mail : knisa4806@gmail.com

Instansi Tempat Kerja : Belum Bekerja

Alamat Kantor : -

### DATA PENDIDIKAN

SD : SD KEMALA BHAYANGKARI 01 MEDAN TAMAT: 2014

SMP : SMP MUHAMMADIYAH 01 MEDAN TAMAT: 2017

SMA : SMA NEGERI 6 MEDAN TAMAT: 2020

## KATA PENGANTAR



Puji syukur atas kehadiran Allah SWT, berkat limpahan rahmat, hidayah dan karuniannya, penulis bisa menyelesaikan skripsi dengan judul “**KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN METODE KNN (*K-NEAREST NEIGHBOR*) DI KOPERASI USAHA TANI GAYO KECAMATAN BEBESEB ACEH TENGAH**”. Skripsi ini adalah salah satu dari beberapa persyaratan untuk menyelesaikan pendidikan dan memperoleh gelar sarjana pada program studi S1 Sistem Informasi di Universitas Muhammadiyah Sumatera Utara.

Penyusunan skripsi ini tidak lepas dari bimbingan, bantuan, arahan dan dukungan dari berbagai pihak terkait. Oleh karena itu pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-khowarizmi, S.Kom.M.Kom selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi
3. Bapak Martiano, Martiano, S.Pd., S.Kom., M.Kom selaku Ketua Program Studi Sistem Informasi
4. Ibu Yoshida Sary, S.E., S.Kom., M.Kom selaku Sekretaris Program Studi Sistem Informasi
5. Bapak Martiano., M.Kom Selaku dosen pembimbing
6. Bapak Drs Mufti. Terima kasih telah berjuang untuk kehidupan penulis, terima kasih telah percaya atas semua keputusan yang telah penulis ambil untuk melanjutkan mimpinya, serta cinta, doa dan motivasi yang selalu membuat penulis percaya bahwa penulis mampu menyelesaikan skripsi ini hingga akhir.

7. Pintu surgaku, ibu Dra. Sudarsini Mustahil penulis mampu melewati semua permasalahan yang penulis alami selama ini jika tanpa doa, ridha dan dukungan dari beliau. Terimakasih ibu, atas kasih sayang dan semangat yang tiada henti ketika penulis merasa putus asa dan tidak mampu, ibu menjadi penguat dan pengingat paling hebat.
8. Yang tersayang abang zulfadli Syarif A.Md., S.Kom dan Khairul Fatah S.T. Terima kasih telah menjadi penyemangat dan penghibur penulis dikala penulis jenuh menghadapi permasalahan selama ini.
9. Kepada Cuaks. Sahabat satu satunya penulis selama menempuh pendidikan di dunia perkuliahan. Terima kasih telah menemani penulis dalam keadaan apapun selama ini, terima kasih karna telah bertahan hingga akhir, terima kasih atas pengalaman hebat selama ini.
10. Dan yang terakhir, kepada diri sendiri, Khairunnisa. Terima kasih karna telah bertahan dan tetap memilih berusaha selama ini. Terima kasih karena memutuskan tidak menyerah sesulit apapun proses penyusunan skripsi ini. Terimakasih kamu hebat bisa sampai di titik ini. Berbahagialah selalu dimanapun berada.

## ABSTRAK

Penelitian ini mengkaji klasifikasi cacat pada biji kopi Arabika menggunakan metode *K-Nearest Neighbor* (KNN) di Koperasi Usaha Tani Gayo, Aceh Tengah. Kopi Arabika, dengan nilai ekonomis tinggi, sering kali dinilai secara subyektif oleh petani menggunakan indra manusia, yang kurang efektif. Oleh karena itu, penelitian ini menggunakan metode KNN untuk meningkatkan akurasi klasifikasi cacat biji kopi. Metode KNN, yang merupakan algoritma supervised, mengklasifikasikan objek berdasarkan kategori tetangga terdekatnya. Penelitian ini menggunakan citra digital berwarna yang diolah dengan *web-tools Teachable Machine* dan *dataset MNIST*. Dataset ini dibagi menjadi tiga bagian: pelatihan, validasi, dan pengujian. Gambar cacat biji kopi diklasifikasikan ke dalam 16 kelas, seperti *Full Sour Bean*, *Full Black Bean*, dan lainnya. Evaluasi menunjukkan bahwa model KNN memiliki akurasi tinggi dalam klasifikasi cacat biji kopi, meskipun membutuhkan waktu komputasi yang signifikan. Hasil penelitian diimplementasikan dalam aplikasi *mobile* berbasis *Flutter* dan bahasa pemrograman *Dart*, yang mempermudah proses klasifikasi cacat biji kopi Arabika di Koperasi Tani Gayo, meningkatkan kualitas dan efisiensi penentuan biji kopi.

**Kata Kunci:** Klasifikasi cacat, biji kopi Arabika, *K-Nearest Neighbor* (KNN), citra digital, *Teachable Machine*, *dataset MNIST*.

## ***ABSTRACT***

*This study examines the classification of defects in Arabica coffee beans using the K-Nearest Neighbor (KNN) method at the Koperasi Usaha Tani Gayo, Aceh Tengah. Arabica coffee, with its high economic value, is often subjectively evaluated by farmers using their senses, which is less effective. Therefore, this research employs the KNN method to enhance the accuracy of defect classification in coffee beans. The KNN method, a supervised algorithm, classifies objects based on the categories of their nearest neighbors. This study utilizes color digital images processed with web-tools Teachable Machine and the MNIST dataset. The dataset is divided into three parts: training, validation, and testing. Defect images of coffee beans are classified into 16 classes, such as Full Sour Bean, Full Black Bean, and others. Evaluation shows that the KNN model has high accuracy in classifying coffee bean defects, although it requires significant computational time. The research results are implemented in a mobile application based on Flutter and the Dart programming language, simplifying the process of classifying Arabica coffee bean defects at the Koperasi Tani Gayo, enhancing the quality and efficiency of coffee bean determination.*

**Keywords:** *Defect classification, Arabica coffee beans, K-Nearest Neighbor (KNN), digital image, Teachable Machine, MNIST dataset.*

## DAFTAR ISI

<b>DAFTAR GAMBAR</b> .....	xiv
<b>DAFTAR TABEL</b> .....	xvi
<b>BAB I</b> .....	1
<b>PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
<b>BAB II</b> .....	4
<b>LANDASAN TEORI</b> .....	4
2.1 Pengertian Sistem .....	4
2.2 <i>Machine Learning</i> .....	4
2.3 <i>Deep Learning</i> .....	5
2.4 <i>Teachable machine</i> .....	6
2.5 Citra Digital.....	6
2.6 <i>Android</i> .....	7
2.7 Metode KNN ( <i>K-Nearest Neighbor</i> ).....	8
2.8 Kopi.....	10
2.9 UML ( <i>Unified Modeling Language</i> ).....	11
2.10 <i>Flutter</i> .....	15
2.11 <i>Confusion Matrix</i> .....	16
2.12 Citra Digital.....	17
<b>BAB III</b> .....	16
<b>METODOLOGI PENELITIAN</b> .....	16
3.1 Jenis Penelitian .....	16
3.2 Tempat dan Waktu Penelitian .....	16
3.3 Teknik Pengumpulan Data .....	17
3.4 Teknik Analisis Data .....	18

3.5 Kerangka Berpikir .....	25
3.6 <i>Use Case Diagram</i> .....	26
3.7 <i>Activity Diagram</i> .....	26
<b>BAB IV .....</b>	<b>28</b>
<b>HASIL DAN PEMBAHASAN .....</b>	<b>28</b>
4.1 Tahap Persiapan <i>Dataset</i> .....	28
4.2 Pengaruh <i>Dataset</i> Terhadap Model .....	30
4.3 Pengaruh <i>Loss</i> Terhadap Model .....	33
4.4 Analisis Model <i>FI-Score, Accuracy, Precision, dan Recall</i> .....	37
4.4.1 Analisis <i>Object Classification Class 1 Full Sour Bean</i> .....	37
4.4.2 Analisis <i>Object Classification Class 2 Full Black Bean</i> .....	51
4.4.3 Analisis <i>Object Classification Class 3 Dried Cherry Pod</i> .....	53
4.4.4 Analisis <i>Object Classification Class 4 Fungus Damage Bean</i> .....	55
4.4.5 Analisis <i>Object Classification Class 5 Busuk Pucuk</i> .....	57
4.4.6 Analisis <i>Object Classification Class 6 Severe Insect Damage</i> .....	59
4.4.7 Analisis <i>Object Classification Class 7 Partial Black Bean</i> .....	61
4.4.8 Analisis <i>Object Classification Class 8 Partial Sour Bean</i> .....	63
4.4.9 Analisis <i>Object Classification Class 9 Parchment Pergamino Bean</i> ...	65
4.4.10 Analisis <i>Object Classification Class 10 Floater Bean</i> .....	67
4.4.10 Analisis <i>Object Classification Class 11 Immature Unripe Bean</i> .....	69
4.4.10 Analisis <i>Object Classification Class 12 Whitered Bean</i> .....	71
4.4.10 Analisis <i>Object Classification Class 13 Shell</i> .....	73
4.4.10 Analisis <i>Object Classification Class 14 Broken Chipped Cut</i> .....	75
4.4.10 Analisis <i>Object Classification Class 15 Hulk Hush</i> .....	77
4.4.10 Analisis <i>Object Classification Class 16 Slight Insect Damage</i> .....	79
4.5 Analisis Data Gambar dengan <i>K-Nearest Neighbor</i> .....	81
4.5.1 Pengumpulan dan Pra-pemrosesan Data Gambar .....	81
4.5.2 Pembangunan Model <i>K-Nearest Neighbor</i> .....	81
4.5.3 Visualisasi Hasil Analisis .....	82
4.6 Menghitung <i>Accuracy</i> dengan <i>K-Nearest Neighbor</i> .....	86
4.7 Menampilkan Tampilan Aplikasi .....	88

<b>BAB V</b> .....	<b>90</b>
<b>PENUTUP</b> .....	<b>90</b>
5.1 Kesimpulan.....	90
5.2 Saran.....	91
<b>DAFTAR PUSAKA</b> .....	<b>92</b>
<b>LAMPIRAN</b> .....	<b>94</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>Confusion Matrix</i> .....	16
Gambar 3.5 Kerangka Berpikir.....	25
Gambar 3.6 <i>Use Case Diagram</i> .....	26
Gambar 3.7 <i>Activity Diagram</i> .....	27
Gambar 4.1 Tampilan <i>Web Teachable Machine</i> .....	29
Gambar 4.2 Hasil <i>Training Dataset per Epoch</i> .....	30
Gambar 4.3 Hasil <i>Loss Training</i> .....	34
Gambar 4.4 Performa Model pada <i>Object Classification Class 1 Full Sour Bean</i>	38
Gambar 4.5 Performa Model pada <i>Object Classification Class 2 Full Black Bean</i> .....	52
Gambar 4.6 Performa Model pada <i>Object Classification Class 3 Dried Cherry Pod</i> .....	54
Gambar 4.7 Performa Model pada <i>Object Classification Class 4 Fungus Damage Bean</i> .....	56
Gambar 4.8 Performa Model pada <i>Object Classification Class 5 Foreign Matter</i> .....	58
Gambar 4.9 Performa Model pada <i>Object Classification Class 6 Severe Insect Damage</i> .....	60
Gambar 4.10 Performa Model pada <i>Object Classification Class 7 Partial Black Bean</i> .....	62
Gambar 4.11 Performa Model pada <i>Object Classification Class 8 Partial Sour Bean</i> .....	63
Gambar 4.12 Performa Model pada <i>Object Classification Class 9 Parchment Pergamino Bean</i> .....	66
Gambar 4.13 Performa Model pada <i>Object Classification Class 10 Floater Bean</i> .....	68
Gambar 4.14 Performa Model pada <i>Object Classification Class 11 Immature Unripe Bean</i> .....	70
Gambar 4.15 Performa Model pada <i>Object Classification Class 12 Whitered Bean</i>	

.....	72
Gambar 4.16 Performa Model pada <i>Object Classification Class 13 Shell</i> .....	74
Gambar 4.17 Performa Model pada <i>Object Classification Class 14 Broken Chipped Cut</i> .....	76
Gambar 4.18 Performa Model pada <i>Object Classification Class 15 Hulk Hush</i> ...	78
Gambar 4.19 Performa Model pada <i>Object Classification Class 16 Slight Insect Damage</i> .....	80
Gambar 4.20 Pemrosesan Data Gambar Menggunakan RGB .....	83
Gambar 4.21 <i>Coding-an</i> Untuk Menentukan Akurasi .....	87
Gambar 4.22 Tampilan Aplikasi Pendeteksi Penyakit Kopi .....	88

## DAFTAR TABEL

Tabel 2.9 UML .....	12
Tabel 3.4 Jenis-jenis <i>Defect</i> pada Kopi .....	18
Tabel 4.1 <i>Object Classification Class 1 Full Sour Bean</i> .....	37
Tabel 4.2 <i>Object Classification Class 2 Full Black Bean</i> .....	51
Tabel 4.3 <i>Object Classification Class 3 Died Cherry Pod</i> .....	53
Tabel 4.4 <i>Object Classification Class 4 Fungus Damage Bean</i> .....	55
Tabel 4.5 <i>Object Classification Class 5 Foreign Matter</i> .....	57
Tabel 4.6 <i>Object Classification Class 6 Severe Insect Damage</i> .....	59
Tabel 4.7 <i>Object Classification Class 7 Partial Black Bean</i> .....	61
Tabel 4.8 <i>Object Classification Class 8 Partial Sour Bean</i> .....	63
Tabel 4.9 <i>Object Classification Class 9 Parchment Pergamino Bean</i> .....	65
Tabel 4.10 <i>Object Classification Class 10 Floater Bean</i> .....	67
Tabel 4.11 <i>Object Classification Class 11 Immature Unripe Bean</i> .....	69
Tabel 4.12 <i>Object Classification Class 12 Whitered Bean</i> .....	71
Tabel 4.13 <i>Object Classification Class 13 Shell</i> .....	73
Tabel 4.14 <i>Object Classification Class 14 Broken Chipped Cut</i> .....	75
Tabel 4.15 <i>Object Classification Class 15 Hulk Hush</i> .....	77
Tabel 4.16 <i>Object Classification Class 16 Slight Insect Damage</i> .....	79

# **BAB I**

## **PENDAHULUAN**

### **1.1 LATAR BELAKANG**

Indonesia merupakan salah satu negara yang menyandang jenis kopi yang cukup terkenal dengan citarasa dan aroma yang ciri khas. Sebagian besar kopi yang diproduksi di Indonesia terdiri dari kopi arabika dan kopi robusta. Kopi Arabika tumbuh di daerah yang memiliki tanah dengan daratan yang tinggi, sementara kopi Robusta ditanam di daerah rendah. Kopi Arabika merupakan salah satu varian paling terkenal karena memiliki rasa yang unik dan biji Arabika juga suatu produk biji kopi global memiliki rasa khas, sehingga dapat menyegarkan badan (Ficri, et al.2022).

Kopi adalah salah satu komoditas perkebunan yang sangat dibutuhkan oleh para petani kopi dan memiliki nilai ekonomi yang cukup besar. Kopi sangat populer di kalangan masyarakat, baik lokal maupun internasional. Kopi memiliki peran penting sebagai penyumbang devisa bagi negara. Selain itu, kopi juga menjadi sumber pendapatan utama bagi para petani kopi di Indonesia serta membuka peluang kerja bagi masyarakat. Kabupaten Aceh Tengah dan Bener Meriah merupakan wilayah dengan produksi kopi tertinggi di Provinsi Aceh. Luas lahan tanaman kopi di kedua kabupaten ini mencapai 80% atau sekitar 96 ribu hektar. Perkebunan kopi yang ada secara keseluruhan adalah milik masyarakat, yang melibatkan sebanyak 77 ribu kepala keluarga. (Noratun, et.al.2017).

Koperasi Usaha Tani Gayo merupakan sebuah entitas bisnis yang berfokus pada proses serta ekspor komoditas kualitas kopi tidak semata-mata ditentukan oleh jenisnya.

Kopi juga ditentukan berdasarkan kulit dari kopi (Amelia, et al. 2023). Kurang telitinya para petani dalam mengidentifikasi kualitas biji Kopi Arabika dengan menggunakan panca Indra seperti tangan, mata, aroma, sentuhan dan penciuman. Maka kurang efektif cara pemilihan biji kopi arabika terbaik.

Citra digital untuk menemukan suatu matriks yang di mana indeks baris dan kolomnya menunjukkan suatu lokasi pada citra tersebut. Elemen-elemen dalam matriks ini (elemen pada suatu gambar) merepresentasikan tingkat kecerahan pada lokasi yang bersangkutan. Dalam penelitian ini, digunakan gambar berwarna. Gambar berwarna terdiri dari tiga warna dasar, yaitu Merah, Hijau, dan Biru.

Dengan ini kita bisa Menggunakan pendekatan KNN (K-Nearest Neighbor) yang merupakan salah satu teknik dari algoritma pembelajaran terawasi (supervised learning), di mana hasil dari query instance dikelompokkan berdasarkan kategori mayoritas dari K tetangga terdekat. Algoritma ini dirancang untuk mengelompokkan objek-objek anyar berdasarkan karakteristik dan contoh yang diambil dari data pelatihan. Proses pengklasifikasian dilakukan dengan menentukan tetangga terdekat dari sampel data uji yang baru. Kedekatan atau jarak tetangga dihitung menggunakan jarak Euclidean. (Ficri, et.al, 2022).

Pada penelitian ini kita dapat mengklasifikasi pemilihan defect pada biji kopi arabika. Dengan ini pada mempermudah dalam mengklasifikasi defect pada biji Kopi Arabika Berdasarkan latar belakang tersebut, penulis merasa tertarik untuk melaksanakan sebuah penelitian. dengan judul “Mengklasifikasi defect pada biji kopi arabika dalam

menerapkan metode KNN (*K-Nearest Neighbor*) Di Koperasi Tani Gayo Kac.Beberes Aceh Tengah”

## 1.2 Rumusan Masalah

Rumusan permasalahan dalam penelitian ini adalah sebagai berikut :

1. Bagaimana mengukur kinerja dalam mengklasifikasi defect pada biji kopi arabika?
2. Bagaimana keakurasi Algoritma KNN (*K-Nearest Neighbor*) dapat mengklasifikasi defect pada biji kopi arabika ?

## 1.3 Batasan Masalah

Untuk menghindari permasalahan dari pembahasan maka penulisan membutuhkan batasan masalah dapat dijelaskan sebagai berikut :

1. Penelitian ini hanya mengklasifikasi 16 Defect pada biji kopi arabika.
2. Perancangan sistem ini menggunakan metode KNN dalam mengklasifikasi biji kopi arabika.

## 1.4 Tujuan Penelitian

Adapun beberapa tujuan dari penulisan penelitian ini adalah sebagai berikut :

1. Untuk mengetahui hasil kinerja dari metode KNN (*K-Nearest Neighbor*) dalam mengklasifikasi defect pada biji kopi arabika.
2. Untuk pengukuran kkn

## 1.5 Manfaat Penelitian

Berikut adalah beberapa manfaat yang dapat diperoleh dari penelitian ini :

1. Untuk mempermudah petani dalam menentukan defect pada kopi arabika

2. Dapat menerapkan algoritma KNN menjadi proses pemilihan kualitas biji kopi arabika terbaik.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem**

Definisi metode dalam pelaksanaan suatu kegiatan adalah sebuah kerangka kerja. Ini digambarkan sebagai sekumpulan alat yang saling terhubung dan berkolaborasi. Sistem juga dapat diartikan sebagai sekumpulan organ atau struktur tubuh yang memiliki fungsi tertentu, atau sebagai kumpulan perangkat lunak dan perangkat keras komputer yang digunakan secara bersama untuk mencapai tujuan spesifik. Sistem merupakan suatu kumpulan elemen yang saling terhubung dan berinteraksi dalam suatu kesatuan untuk melaksanakan suatu proses demi mencapai tujuan utama. (Dewi M.Ariska, et,al. 2024).

#### **2.2 *Machine Learning***

*Machine Learning* adalah pendekatan yang menggunakan komputer, di mana perangkat tersebut dilengkapi kemampuan untuk belajar dari data tanpa perlu diprogram sebelumnya. Algoritma *Machine Learning* mengembangkan model matematis berdasar pada kumpulan data yang disebut "data pelatihan" untuk menghasilkan prediksi atau keputusan tanpa perlu diprogram secara langsung untuk melaksanakan tugas tersebut. (Pratama, 2020).

Pada umumnya, hal ini terbagi menjadi tiga kategori, yaitu :

- a) *Supervised Learning*

Model di atas ini merupakan teknik dimana diajarkan dengan pengetahuan sehingga dapat memprediksi dimasa mendatang atau dengan kata lain mengajarkan dan melatih model dengan beberapa data.

b) *Unsupervised Learning*

Model di atas ini merupakan pembelajaran tanpa pengawasan untuk menentukan informasi yang mungkin tidak dapat dilihat oleh pandangan manusia.

c) *Reinforcement Learning*

Model di atas ini merupakan Teknik pemodelan dari pengalaman yang diperoleh, sebagaimana manusia yang juga belajar melalui pengalaman. dapat diuraikan menjadi "kerugian" atau "kerusakan pada suatu benda atau kondisi.

### **2.3 Deep Learning**

Deep Learning merupakan salah satu cabang dari Machine Learning yang memanfaatkan data dalam jumlah besar untuk menyelesaikan berbagai masalah dengan menggunakan kopi yang telah rusak. Metode Deep Learning menawarkan struktur yang sangat efektif untuk pembelajaran terawasi. Deep learning merupakan kumpulan algoritma dalam pembelajaran mesin yang berupaya untuk memahami data pada berbagai tingkatan, sesuai dengan variasi tingkat abstraksinya.

Berikut adalah beberapa tipe Deep Learning :

a) *Deep Learning*

untuk proses belajar mengenai tanda-tanda pengawasan. Jenis deep learning ini diterapkan ketika label untuk variabel target tidak ada, dan analisis pola dilakukan dengan menghitung hubungan yang lebih kuat dari unit yang dianalisis.

b) *Hybrid Deep Networks*

Deep learning gabungan mengadopsi metode yang bertujuan untuk mencapai hasil yang memuaskan, baik melalui pembelajaran terawasi untuk menganalisis pola maupun dengan penerapan pembelajaran tidak terawasi.

## **2.4 Teachable Machine**

*Teachable machine* adalah sebuah perangkat yang memungkinkan pengguna untuk membuat model klasifikasi dengan cara yang sederhana, yang dapat dimanfaatkan dalam pengembangan aplikasi pembelajaran mesin. *Teachable machine* yang disediakan oleh Google memanfaatkan sistem pembelajaran untuk menganalisis data tanpa perlu pengaturan yang jelas dalam pemrogramannya. *Teachable Machine* adalah alat yang dapat menangani berbagai jenis input, seperti gambar, suara, dan juga gerakan. Tujuannya adalah untuk mempermudah dalam mengklasifikasi model.

Pelatihan data adalah suatu proses yang memerlukan banyak sumber daya, berupa koleksi informasi mengenai objek biji kopi yang mengalami kerusakan. Satu objek biji kopi dari tanaman perlu memiliki berbagai variasi bentuk citra. Tujuan dari hal ini adalah untuk meningkatkan ketepatan hasil klasifikasi.

## **2.5 Citra Digital**

Citra merupakan gambaran dua dimensi yang menciptakan representasi fisik dari objek tiga dimensi. Representasi dalam bentuk visual dapat bervariasi, mulai dari gambar hitam-putih dalam sebuah foto yang statis hingga gambar berwarna yang dinamis yang ditampilkan di televisi. (Asti Riani Putri, 2020). Adapun beberapa citra digital yaitu

gambar digital yang berfungsi sebagai gambar analog dan gambar yang berfungsi sebagai gambar digital.

Citra Analog adalah sebuah contoh dari suatu gambar dalam membentuk sinyal yang tidak dapat dipecahkan, dan bernilai tidak terbatas. Dikarenakan computer hanya dapat menkung gambar digital sebelum dapat ditangan oleh computer.

Citra Digital adalah jenis citra yang terbentuk dari sinyal diskrit yang tidak saling terhubung, di mana nilai-nilai dalam citra digital ini diwakili oleh angka biner. Citra digital merupakan gambaran yang dapat dihasilkan oleh komputer.

Pengolahan citra digital adalah bidang studi yang fokus pada metode untuk mengolah gambar. Citra yang dimaksud mencakup foto statis (gambar tetap) maupun gambar bergerak seperti rekaman video. Sementara itu, pengertian dari digital adalah proses pengolahan gambar yang dilakukan menggunakan komputer secara digital. RGB adalah akronim dari Merah-Hijau-Biru, yang terdiri dari tiga warna dasar yang biasanya dijadikan sebagai referensi untuk warna dasar dalam sistem RGB.

## ***2.6 Android***

*Android* merupakan sistem operasi untuk perangkat mobile yang dirancang oleh Google, dengan dasar modifikasi dari versi kernel Linux. Sistem ini ditujukan khusus untuk perangkat dengan layar sentuh, seperti ponsel pintar dan tablet (Open Handset Alliance, 2012). Seiring dengan perkembangannya, Android tidak hanya terbatas pada perangkat mobile, tetapi Google juga telah menciptakan Android TV untuk televisi, Android Auto untuk kendaraan, dan Android Wear untuk smartwatches.

Antarmuka Android biasanya melibatkan interaksi langsung melalui gerakan sentuh, seperti menggeser, mengetuk, dan mencubit untuk mengatur objek di layar, di samping itu terdapat juga keyboard virtual untuk pengetikan teks. Android juga kompatibel dengan perangkat internal seperti akselerometer, giroskop, dan sensor kedekatan yang digunakan oleh sejumlah aplikasi untuk merespons aksi pengguna, seperti mengubah tampilan layar dari mode potret menjadi mode lanskap saat perangkat diputar. Aplikasi, yang sering disebut sebagai "apps" di Android, berfungsi untuk meningkatkan kemampuan perangkat. Aplikasi Android umumnya dibuat dengan menggunakan bahasa pemrograman Java dan memanfaatkan Android Software Development Kit (SDK) (Android Developers, 2017).

## **2.7 Metode KNN (*K-Nearest Neighbor*)**

Metode *K-Nearest Neighbor* (KNN) adalah suatu teknik untuk mengklasifikasikan objek berdasarkan data pelatihan yang memiliki jarak terdekat dengan objek yang sedang dianalisis. *Nearest neighbor* adalah metode yang digunakan untuk menemukan kasus dengan cara mengukur kesamaan antara kasus baru dan kasus yang sudah ada, dengan mempertimbangkan nilai serta bobot yang serupa. Sasaran dari metode KNN (*K-Nearest Neighbor*) adalah untuk mengklasifikasikan objek baru dengan menggunakan atribut serta sampel pelatihan.

Prinsip dasar K-Nearest Neighbor (KNN) adalah mengidentifikasi jarak terdekat antara data yang sedang dianalisis dengan K tetangga terdekat yang ada dalam dataset pelatihan. Metode ini tergolong dalam kategori klasifikasi nonparametrik. Metode KNN cukup mudah dipahami, beroperasi dengan cara mengukur jarak terdekat antara instance

yang dicari dan sampel pelatihan untuk mengidentifikasi KNN-nya. (Aziz Ali Mahendra, et.al. 2023).

K-Nearest Neighbour, yang lebih dikenal sebagai KNN, merupakan bagian dari kelompok pembelajaran berbasis instansi. Metode ini dilakukan dengan mengelompokkan sekumpulan objek dari data pelatihan yang memiliki jarak paling mendekati dengan data yang baru atau yang sedang diuji. Algoritma KNN adalah sebuah metode yang digunakan untuk melakukan pengelompokan pada objek tertentu dengan mempertimbangkan data pelatihan yang memiliki kesamaan tertinggi dengan objek tersebut. (Taufiq, et.al. 2022). Berikut adalah rumus dari algoritma KNN (*K-Nearest Neighbour*) :

$$d_i = \sqrt{\sum_{i=1}^n (x_{ij} - p_i)^2} \quad (1)$$

Keterangan :

$d_i$  = Jarak Sempel

$x_{ij}$  = Data Sempel

$p_j$  = Data Input Variabel ke-j

$n$  = Jumlah sampel

Cara-cara untuk menghitung menggunakan metode *K-Nearest Neighbor* sebagai berikut:

- a) Menentukan parameter dengan jumlah tetangga terdekat.
- b) Menghitung kuadrat dari jarak setiap objek terhadap data sampel yang telah disediakan dengan menggunakan rumus di atas.

- c) Selanjutnya mengelompokkan objek-objek itu berdasarkan jarak terdekat.
- d) Mengumpulkan kategori klasifikasi.
- e) Dengan memanfaatkan kategori *K-Nearest Neighbor* yang paling domain maka dapat diprediksi nilai query yang sudah dihitung dapat di prediksi.

Berikut adalah beberapa dari keunggulan dari metode *K-Nearest Neighbor* adalah sebagai berikut :

- a. KNN memiliki kelebihan pada data training data set yang memiliki banyak noisy
- b. Berkinerja baik terhadap jumlah data training yang besar besar.

Beberapa kelemahan dari metode *K-Nearest Neighbor* adalah kewajiban untuk menentukan nilai *K* serta dalam memilih atribut yang paling relevan. Pemilihan Fitur adalah tahap dalam memilih sebagian dari ciri-ciri asli dengan menghapus fitur yang tidak berkaitan. Pemilihan fitur akan meningkatkan efisiensi. Rephrase: Pemilihan fitur dapat meningkatkan efektivitas. Menitikberatkan pada pencarian data yang relevan adalah tujuan utama dari pemilihan fitur. Fitur yang tidak penting atau fitur yang terlalu banyak dapat berdampak besar pada hasil, sehingga pemilihan fitur perlu mampu mengenali data tersebut.(Fatma Tolana, at.el, 2023)

## 2.8 Kopi

Kopi adalah salah satu komoditas tropis penting yang telah diperdagangkan secara global, menyumbang sekitar 50% dari keseluruhan ekspor komoditas tropis. Kepopuleran dan minat global terhadap kopi terutama disebabkan oleh cita rasanya yang khas, yang juga dipengaruhi oleh aspek sejarah, tradisi, sosial, dan pertimbangan ekonomi. (Mahmud Saputra, et.al. 2020). Di Indonesia kopi termasuk komoditas

pengekspor yang relatif besar. Kebun kopi di Indonesia memainkan peranan vital dalam ekonomi negara, sekaligus memberikan peluang kerja bagi masyarakat sekitar. adapun jenis Kopi yang digemari oleh masyarakat yaitu jenis Kopi Arabika.

Kopi Arabika Gayo adalah salah satu varietas kopi terbaik yang berasal dari Indonesia dan telah dikenal secara internasional. Jenis kopi ini memiliki rasa yang sangat istimewa dan memiliki nilai ekonomis yang cukup tinggi di pasar global. Kopi Arabika Gayo memiliki ciri khas yang menonjol, di mana kandungan kimia dalam bijinya dipengaruhi oleh elevasi lokasi penanamannya. Biji kopi Arabika Gayo yang tumbuh pada ketinggian antara 1000 hingga 1400 meter di atas permukaan laut. (Irma Zarwinda, et.al. 2023).

Kopi arabika (*Coffea arabica*) tergolong dalam keluarga Rubiaceae dan merupakan spesies dari genus *Coffea*. Kopi ini banyak ditemukan di wilayah pegunungan di Indonesia. Salah satu jenis kopi yang paling terkenal di Indonesia adalah Kopi Arabika Gayo. Kopi ini berasal dari Kabupaten Bener Meriah, yang terletak di dataran tinggi Gayo. Kabupaten ini merupakan hasil pemekaran dari Kabupaten Aceh Tengah. Kopi ini memiliki aroma yang terdiri dari berbagai unsur, termasuk caramel, vanilla, buah, dan nuansa smoky, yang menciptakan karakter aroma kopi yang unik dan istimewa. Kopi Arabika Gayo memiliki karakteristik rasa, aroma, dan tubuh yang kuat, dengan sentuhan asam yang dihasilkan dari buah-buahan, serta sedikit rasa pedas dan manis-pahit yang menciptakan cita rasa unik dari kopi Gayo. Tidak hanya disukai di Sumatera, kopi Gayo juga populer di kalangan masyarakat global.

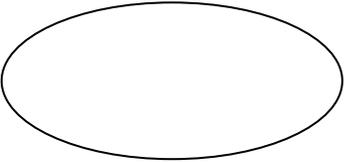
## **2.9 UML (*Unified Modeling Language*)**

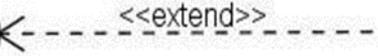
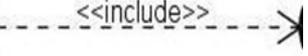
*Unified Modeling Language* (UML) merupakan sebuah bahasa yang telah distandarisasi untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. Pemanfaatan UML oleh seorang pengembang sistem memungkinkan untuk membuat diagram yang menggambarkan pandangan tentang suatu sistem dengan format yang baku, mudah dipahami, serta menyediakan cara untuk mempermudah komunikasi dengan pengguna. (Fadli Kamil, 2023).

#### 1) Use Case Diagram

Use case diagram memiliki kegunaan yaitu untuk menggambarkan interaksi antara actor (stakeholder atau pengguna) dengan sistem. Diagram penggunaan adalah salah satu tipe diagram dalam Bahasa Pemodelan Terpadu (UML) yang berfungsi untuk menunjukkan interaksi antara aktor. (stakeholder atau pengguna) dengan sistem yang akan dikembangkan. Diagram ini membantu mengidentifikasi dan memvisualisasikan scenario penggunaan yang berbeda, serta menjelaskan bagaimana sistem akan merespons Tindakan atau permintaan dari para actor tersebut.

Tabel 2.1 UML

No	Simbol	Deskripsi
1	Use Case 	Kemampuan yang ditawarkan oleh sistem berfungsi sebagai unit-unit yang saling berkomunikasi melalui pertukaran pesan antara unit atau aktor.

2	<p>Aktor</p> 	<p>Prosedur atau sistem yang berbeda dapat saling berinteraksi dengan sistem informasi yang telah ada.</p>
3	<p>Association</p> 	<p>Interaksi antara aktor melalui use case yang terlibat dalam use case yang memiliki hubungan dengan aktor.</p>
4	<p>Extend</p> 	<p>Penggunaan kasus tambahan di mana penggunaan kasus yang ditambahkan dapat berdiri sendiri meskipun tanpa adanya penggunaan kasus lainnya.</p>
5	<p>Include</p> 	<p>Pemanggilan use case oleh use case lain</p>
6	<p>Generalization</p> 	<p>Keterkaitan antara generalisasi dan spesialisasi (dari yang umum ke yang khusus) di antara dua skenario penggunaan.</p>

## 2) Activity Diagram

Diagram Aktivitas adalah salah satu jenis diagram yang ada dalam UML yang berfungsi untuk menggambarkan alur proses atau kegiatan dalam suatu sistem.

Diagram ini berperan dalam menggambarkan tindakan dan proses dari satu kegiatan ke kegiatan yang lain, termasuk pemilihan cabang, peluang, dan aktivitas paralel. Activity Diagram membantu dalam memvisualisaikan bagaimana objek berinteraksi dalam bentuk aktivitas, serta bagaimana alur eksekusi dari satu aktivitas ke aktivitas lainnya.

### 3) *Sequence* Diagram

Diagram urutan digunakan untuk menggambarkan aliran pesan atau interaksi antara objek dalam sistem seiring berjalannya waktu. Diagram ini membantu dalam memahami urutan Tindakan scenario tertentu. Berikut beberapa elemen utama dalam sequence diagram :

- a. Objek adalah entitas dalam sistem yang berpartisipasi dalam interaksi. Ini bisa berupa contoh kelas atau aktor yang terlibat dalam skenario. Objek digambarkan sebagai blok vertikal dengan nama objek di atasnya.
- b. Lifeline adalah garis putus-putus yang menghubungkan objek dengan pesan atau pemanggilan metode. Garis hidup menggambarkan masa hidup objek dalam skenario dan menunjukkan berapa lama objek aktif dalam interaksi.
- c. Pesan adalah komunikasi yang dikirim antar objek. Pesan mewakili panggilan metode atau pesan yang diteruskan dari satu objek ke objek lainnya. Pesan digambarkan sebagai anak panah dengan anak panah menunjuk ke objek penerima.
- d. Aktivasi menggambarkan periode waktu ketika objek menerima atau merespons pesan. Garis vertikal pada garis hidup objek yang dimaksud menunjukkan aktivasi.

- e. Return Message adalah pesan balasan yang dikirim oleh objek penerima sebagai respons terhadap pesan asli. Ini menunjukkan nilai kembalian dari metode yang dipanggil.

## **2.10 Flutter**

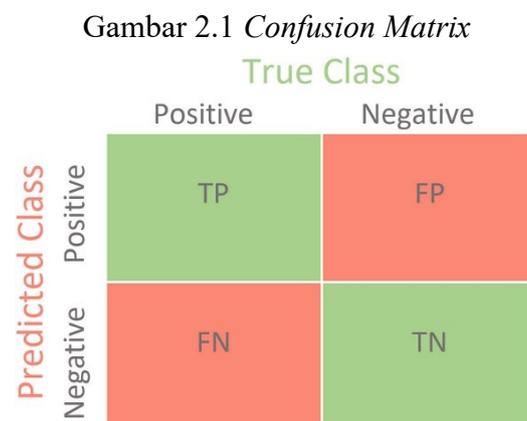
*Flutter* adalah sebuah platform yang dikembangkan oleh Google untuk pengembangan aplikasi mobile, yang memungkinkan pembuatan aplikasi di iOS, Android, aplikasi web, serta aplikasi desktop (Kurosaki, 2020). Berbagai keunggulan *flutter* antara lain adalah fitur hot reload, banyaknya pilihan widget, serta desain yang menarik (Giordano, 2019). Flutter bertujuan untuk menjadi inovator dalam pembuatan aplikasi mobile. *Flutter* menawarkan seluruh perangkat yang diperlukan oleh para pengembang untuk menciptakan aplikasi yang menakjubkan tanpa mengorbankan kinerja dan kemampuan untuk berkembang.

*Flutter* pada dasarnya memiliki sejumlah konsep yang berfokus pada performa aplikasi dan pengalaman pengguna. *Flutter* memanfaatkan bahasa pemrograman Dart sebagai sarana yang mendukung efisiensi pengembang selama tahap pengembangan, serta untuk menciptakan aplikasi yang sudah dioptimalkan untuk diterbitkan (Biessek, 2019). *Flutter* adalah proyek sumber terbuka yang dioperasikan di GitHub dengan kontribusi dari Google serta anggota komunitas. *Flutter* memanfaatkan Dart, yaitu bahasa pemrograman berorientasi objek yang kontemporer, yang dapat diubah menjadi kode asli ARM serta kode *JavaScript*. *Flutter* memanfaatkan mesin rendering Skia 2D yang kompatibel dengan berbagai jenis platform perangkat keras dan perangkat lunak. Teknologi ini juga digunakan oleh Google Chrome, Chrome OS, Android, Mozilla Firefox, Firefox OS, dan lain-lain. Flutter memanfaatkan bahasa pemrograman Dart

untuk merancang antarmuka pengguna (*user interface*). *Flutter* memiliki sifat deklaratif, artinya *Flutter* merancang antarmuka pengguna (UI) untuk mewakili keadaan (*state*) aplikasi. Saat keadaan (data) berubah, antarmuka pengguna (UI) akan mengalami perubahan, dan *Flutter* akan membuat *instance* baru menggunakan *widget*. (Napoli, 2019).

## 2.11 Confusion Matrix

*Confusion matrix* adalah alat evaluasi yang sering digunakan dalam *machine learning* berbasis *supervised learning* untuk menilai kinerja model klasifikasi. Metode ini memberikan gambaran tentang tingkat kekeliruan model dalam mengklasifikasikan data ke dalam berbagai kelas, tanpa bergantung pada algoritma spesifik yang digunakan (Rezky Firmansyah, 2021).



Tujuan utama dari *confusion matrix* adalah untuk menghitung akurasi dalam konteks data mining. Dengan menggunakan *confusion matrix*, kita dapat memperoleh metrik penting seperti akurasi, presisi, dan *recall*. Akurasi mengukur persentase data yang diklasifikasikan dengan benar dibandingkan dengan total data yang diuji. Presisi, atau

*confidence*, menunjukkan proporsi dari hasil prediksi positif yang benar-benar positif dalam data sebenarnya. Sementara itu, *recall*, atau *sensitivity*, mengukur proporsi dari kasus positif yang benar-benar teridentifikasi sebagai positif oleh model (Rezky Firmansyah, 2021).

## 2.12 Citra Digital

Citra umumnya diartikan sebagai gambar. Dalam pengertian yang lebih spesifik, citra adalah deskripsi visual tentang suatu benda. Jenis citra dapat berupa foto individu, ilustrasi, hasil pemindaian sinar-X, serta gambar dari satelit. Secara umum, gambaran dapat dikategorikan menjadi dua tipe, yaitu gambar analog dan gambar digital. Citra analog merupakan gambar yang ditemukan pada media fisik, seperti kertas, contohnya foto mahasiswa di kartu identitas atau hasil sinar-X. Citra digital merupakan gambaran yang disajikan dalam format data digital dan dapat diolah dengan bantuan komputer. Gambar digital terdiri dari banyak piksel, di mana setiap piksel memiliki posisi tertentu. Citra digital dapat dijelaskan sebagai suatu fungsi dua dimensi  $f(x,y)$ , di mana  $x$  dan  $y$  berperan sebagai koordinat di atas bidang datar. Fungsi  $f$  untuk setiap pasangan koordinat  $(x,y)$  mencerminkan intensitas atau tingkat keabuan (grey level) dari gambar di titik tersebut. Gambar digital memiliki nilai yang terbatas (finite) dan terpisah pada sumbu  $x$ ,  $y$ , serta  $f$ . Citra digital terdiri dari sejumlah komponen yang dikenal sebagai elemen gambar, elemen citra, pels, atau piksel. (Mulyani, 2023). Dalam pengolahan citra digital terdapat dua metode. Pada salah satu metode, baik input maupun output adalah gambar. Pada metode lainnya, inputnya adalah gambar sedangkan outputnya berupa beberapa fitur dari gambar tersebut (Sharma et al, 2022).

## BAB III

### METODE PENELITIAN

#### 3.1 Jenis Penelitian

Tipe penelitian yang diterapkan adalah metode kuantitatif. Penelitian kuantitatif merupakan suatu cara untuk memperoleh pengetahuan dengan menggunakan data yang berbentuk angka sebagai instrumen untuk menganalisis penjelasan. Di mana penelitian ini mengumpulkan data dengan mensurvei untuk menentukan presentase penentuan dari kualitas biji kopi arabika.

#### 3.2 Tempat dan Waktu Penelitian

##### 3.2.1 Tempat Penelitian

Lokasi untuk pelaksanaan penelitian ini adalah Koperasi Usaha Tani Gayo yang terletak di jalan keramat mufakar N0.202 kecamatan beberes kabupaten aceh tengah.

##### 3.2.2 Waktu Penelitian

Adapun waktu pelaksanaan penelitian :

No	Aktivitas	Bulan					
		Desember	Januari	Februari	Maret	April	Mei
1.	Pengajuan Judul						
2.	Penelitian Pra-Riset						
3.	Penyusunan Proposal						
4.	Pengumpulan data						

5.	Seminar proposal						
6.	Riset						
7.	Penyusunan skripsi						
8.	Bimbingan skripsi						
9.	Pembuatan Sistem						
10.	Sidang Meja Hijau						

### 3.3 Teknik Pengumpulan Data

Adapun proses pengumpulan data penelitian pada koperasi usaha tani gayo sebagai berikut :

a) Studi Pustaka

Studi Pustaka merupakan proses mengumpulkan informasi dari berbagai jurnal yang relevan dengan pembahasan penelitian untuk mendukung pemecahan permasalahan penelitian.

b) Observasi

Observasi merupakan langkah dalam proses mengunjungi lokasi penelitian untuk mengumpulkan informasi atau data yang diperlukan. penelitian ini pengumpulan data juga menggunakan sensor untuk menentukan nilai dataset. Sensor yang digunakan yaitu sensor kelembapan, sensor warna, sensor bau dan sensor berat.

c) Wawancara

Wawancara adalah suatu metode pengumpulan informasi yang dilakukan dengan mengajukan pertanyaan kepada beberapa petani kopi terkait dengan kualitas terbaik biji kopi arabika.

### 3.4 Teknik Analisis Data

Penulis menerapkan teknik analisis data melalui metode KNN dengan langkah-langkah sebagai berikut :

- 1) Menentukan parameter dengan jumlah tetangga terdekat.
- 2) Menghitung kuadrat jarak setiap objek terhadap data sampel yang telah disediakan dengan memanfaatkan rumus di atas.
- 3) Selanjutnya mengelompokkan objek-objek tersebut berdasarkan jarak terdekat.
- 4) Mengumpulkan kategori klasifikasi KNN.
- 5) Dengan memanfaatkan kategori K-Nearest Neighbor yang paling berpengaruh, maka nilai query yang sudah dihitung dapat diprediksi..

Jenis-jenis defect pada kopi arabika sebagai berikut ;

No	Nama dan gambar	Keterangan
1	<p><b>Full Black Bean (Kacang Hitam Penuh)</b></p> 	Ini bisa ditandai dengan warna hitam secara keseluruhan pada green bean

2	<b>Full Sour Bean (Kacang Asam Penuh)</b> 	Green bean berwarna kemerahan atau kuning kecokelatan.
3	<b>Dried Cherry/Pod (Ceri/Polong Kering)</b> 	Green bean masih terlindungi oleh kulit ceri kopi. Tidak terpisah sempurna.

4	<p><b>Fungus Damage Bean (jamur Merusak Kacang)</b></p> 	<p>Green bean berwarna kekuningan dan kecokelatan dan bertekstur berbeda yang diakibatkan jamur.</p>
5	<p><b>Foreign Matter (Benda Asing)</b></p> 	<p>Benda-benda yang ikut masuk ke green bean selain dari unsur dan bagian kopi seperti batu, sampah dan lain-lain.</p>
6	<p><b>Severe Insect Damage Bean (Kacang Kerusakan Serangga)</b></p> 	<p>Ini ditandai dengan fisik biji green bean yang rusak dan bolong lebih dari 3 titik. Hal ini terjadi dikarenakan serangan serangga.</p>

7	<b>Partial Black Bean (Kacang Hitam Sebagian)</b> 	Setengahnya berwarna hitam kecokelatan.
8	<b>Partial Sour Bean (Kacang Asam Parsial)</b> 	Berwarna kemerahan dan kekuningan di setengah bagian saja. Dikarenakan setelah proses panen biji merah tidak langsung diproses.
9	<b>Parchment/pergamino Bean</b> 	Biji hijau berwarna pucat karena diselimuti oleh dried parchment.

10	<b>Floater Bean (Kacang Kopong)</b> 	Warnanya lebih terang dan ini adalah biji kopi yang biasanya kurang padat dan ditemukan mengambang pada proses pasca-panen.
11	<b>Immature/Unripe Bean (Kacang Belum Matang)</b> 	Ini adalah biji hijau yang underdevelope dan tidak terproses dengan sempurna. Biasanya ditandai dengan fisik yang lebih kecil.
12	<b>Withered Bean (Kacang Layu)</b> 	Terdapat kerutan pada permukaan green bean.

13	<b>Shell</b> 	Bentuknya seperti cangkang dan memiliki rongga lebar di permukannya.
14	<b>Broken/Chipped/Cut</b> <b>(Rusak/Terkupas/Terpotong)</b> 	Biasanya ini adalah biji kopi yang tidak utuh. Biasanya terjadi karena mesin atau cacat akibat proses dan ulah manusia.
15	<b>Hull/Huks</b> 	Bagian-bagian kopi selain biji yang masuk ke dalam kopi. Biasanya kulit ceri dan lainnya.

16	<p><b>Slight Insect Damage (Kerusakan Ringan Akibat Serangga)</b></p> 	<p>Kerusakan yang dialami karena serangan serangga tapi kurang dari tiga lobang.</p>
----	---	--

Tabel 3.4 Jenis-Jenis Defect pada Kopi Arabika

Setiap poin defect (kecacatan) pada biji kopi arabika sebagai berikut :

No	Jenis	Nilai Cacat
1	Full Black	1
2	Full Sour	1
3	Dried Cherry/Pad	1
4	Fungus Damaged	1
5	Foreign Matter	1
6	Severe Insect Damage	5
7	Partial Black	3
8	Partial Sour Bean	3
9	Parchment/Pargamino	5
10	Floater	5

11	Immature/Uniripe	5
12	Withered	5
13	Shell	5
14	Broken	5
15	Hull/husk	5
16	Slight Insect Damage	10

Penulis menggunakan teknik analisis data dengan metode KNN (K-Nearest Neighbor). Tahapan-tahapan dalam analisis data adalah sebagai berikut:

1. Menentukan Parameter (Jumlah Tetangga Paling Dekat)

KNN membutuhkan parameter K, yang merupakan jumlah tetangga terdekat yang akan digunakan untuk klasifikasi atau regresi. Nilai K dipilih berdasarkan eksperimen dan validasi untuk mendapatkan performa terbaik.

2. Menghitung Jarak Euclidean

Jarak antara dua titik data  $X = (X_1, X_2, \dots, X_n)$  dan  $Y = (Y_1, Y_2, \dots, Y_n)$  dapat dihitung menggunakan rumus jarak Euclidean sebagai berikut:

$$d(X, Y) = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \dots + (X_n - Y_n)^2}$$

Dimana  $d(X, Y)$  adalah jarak antara titik X dan Y, dan  $X_i$  serta  $Y_i$  adalah komponen-komponen dari titik data X dan Y masing-masing.

3. Mengurutkan Objek Berdasarkan Jarak

Setelah menghitung jarak antara titik data sampel dengan semua titik data dalam dataset, langkah selanjutnya adalah mengurutkan semua titik data berdasarkan jarak dari yang terkecil ke terbesar.

#### 4. Menentukan Klasifikasi Berdasarkan K Tetangga Terdekat

Dari data yang telah diurutkan, ambil K titik data terdekat. Klasifikasi dilakukan dengan memilih kategori yang paling banyak muncul di antara K tetangga terdekat tersebut. Misalnya, jika dari 5 tetangga terdekat, 3 di antaranya adalah biji kopi berkualitas baik, maka sampel akan diklasifikasikan sebagai biji kopi berkualitas baik.

#### 5. Validasi Model

Validasi dilakukan untuk menguji keakuratan model yang telah dibangun. Teknik yang umum digunakan adalah *k-fold cross validation*. Proses ini melibatkan pembagian dataset menjadi k subset, kemudian model dilatih menggunakan k-1 subset dan diuji menggunakan subset yang tersisa. Proses ini diulang sebanyak k kali dengan setiap subset digunakan sekali sebagai data uji. Keakuratan rata-rata dari semua iterasi dihitung untuk mendapatkan gambaran performa model.

#### 6. *Confusion Matrix*

*Confusion Matrix* adalah alat analisis yang dipakai untuk menilai efektivitas algoritma pengklasifikasian. Matriks ini menggambarkan jumlah prediksi yang tepat dan keliru yang dihasilkan oleh model jika dibandingkan dengan nilai sebenarnya dalam dataset. Berikut adalah bentuk umum dari *confusion matrix*:

Tabel 3.4 *Confusion Matrix*

Aktual	Prediksi	
	+	-
+	<i>True positives (A)</i>	<i>False negatives (B)</i>
-	<i>False positives (C)</i>	<i>True negatives (D)</i>

Untuk menghitung tingkat akurasi pada matriks kebingungan, Anda dapat menggunakan rumus berikut ini :

$$Akurasi = \frac{(A + D)}{(A + B + C + D)}$$

Presisi (*Precision*) adalah rasio antara item relevan yang dipilih dengan total item yang dipilih. Dengan kata lain, presisi mengukur sejauh mana jawaban yang diberikan cocok dengan permintaan informasi. Presisi dapat dihitung menggunakan rumus berikut:

$$Akurasi = \frac{A}{(C + A)}$$

*Recall* adalah perbandingan antara jumlah item yang relevan yang terpilih dengan total item relevan yang ada. Recall dapat dihitung menggunakan rumus yang berikut ini :

$$Akurasi = \frac{A}{(A + D)}$$

Presisi dan *Recall* dapat dinilai menggunakan perhitungan persentase (dalam rentang 1-100%) atau dengan angka antara 0 hingga 1. F1 Score adalah suatu ukuran yang menghitung rata-rata antara presisi dan *recall*

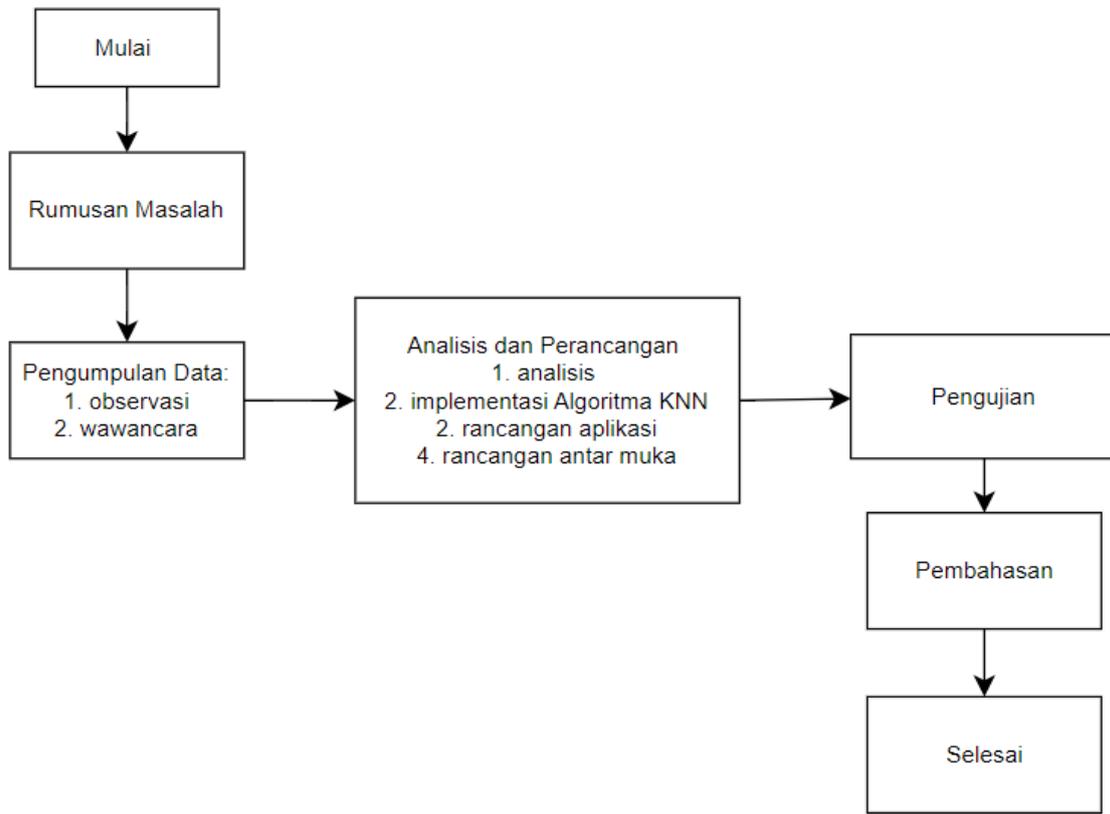
$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall}$$

Akurasi memiliki tingkat nilai diagnosa yaitu:

- a. rentang 0. 90 hingga 100 dapat digolongkan sebagai klasifikasi yang sangat baik.
- b. Akurasi yang berada pada rentang 0. 80 hingga 090 dapat diklasifikasikan sebagai klasifikasi yang baik..
- c. Akurasi yang berkisar antara 0. 70 hingga 080 dapat dianggap sebagai klasifikasi yang cukup baik..
- d. Akurasi yang berkisar antara 0. 60 hingga 070 dapat dianggap sebagai klasifikasi yang kurang baik..
- e. Akurasi yang berada dalam rentang 0. 50 hingga 00 dapat dianggap sebagai kegagalan..

### **3.5 Kerangka Berpikir**

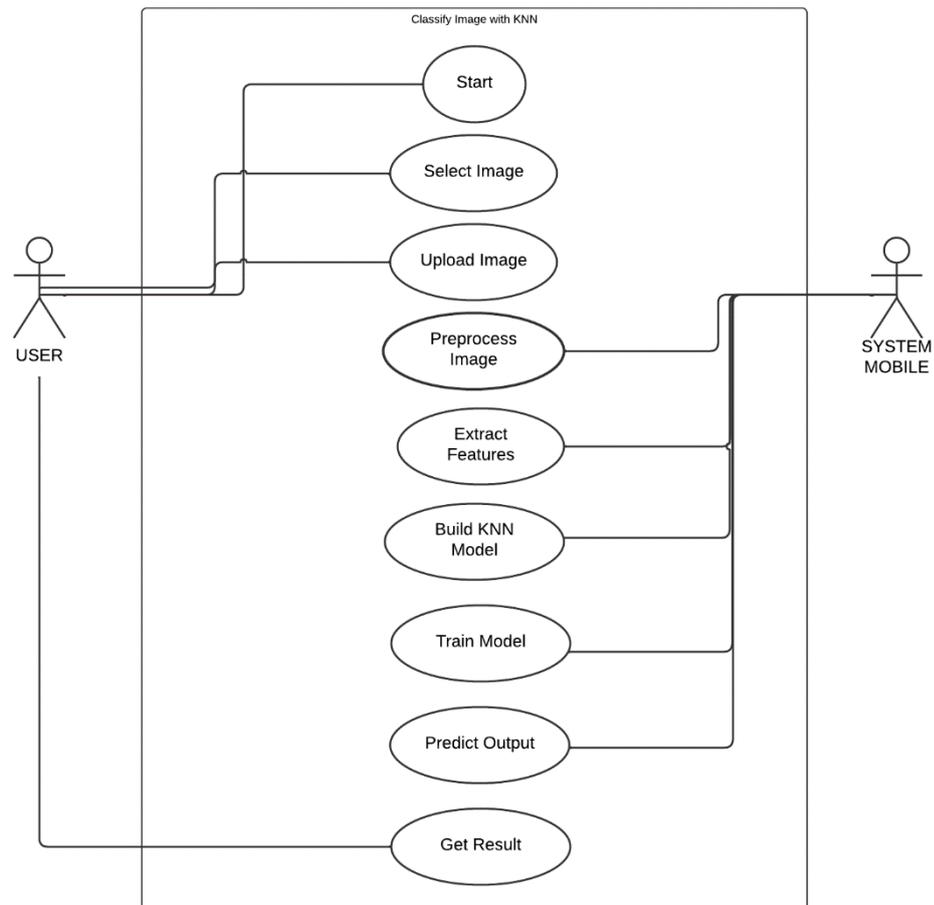
Adapun kerangka berpikir dari peneitian ini adalah sebagai berikut :



Gambar 3.5 Kerangka Berpikir

Gambar diatas merupakan kerangka berpikir dari penelitian yang saya lakukan. Dimulai dari rumusan masalah kemudian mengumpulkan data dengan cara observasi dan melakukan wawancara kepada pihak yang terkait. Setelah itu melakukan analisis dan merancang sistem. Setelah sistem dibangun maka tahap selanjut nya melakukan pengujian sistem. Kemudian mendapatkan hasil dan pembahasan tentang sistem tersebut dan selesai.

### 3.6 Use Case Diagram



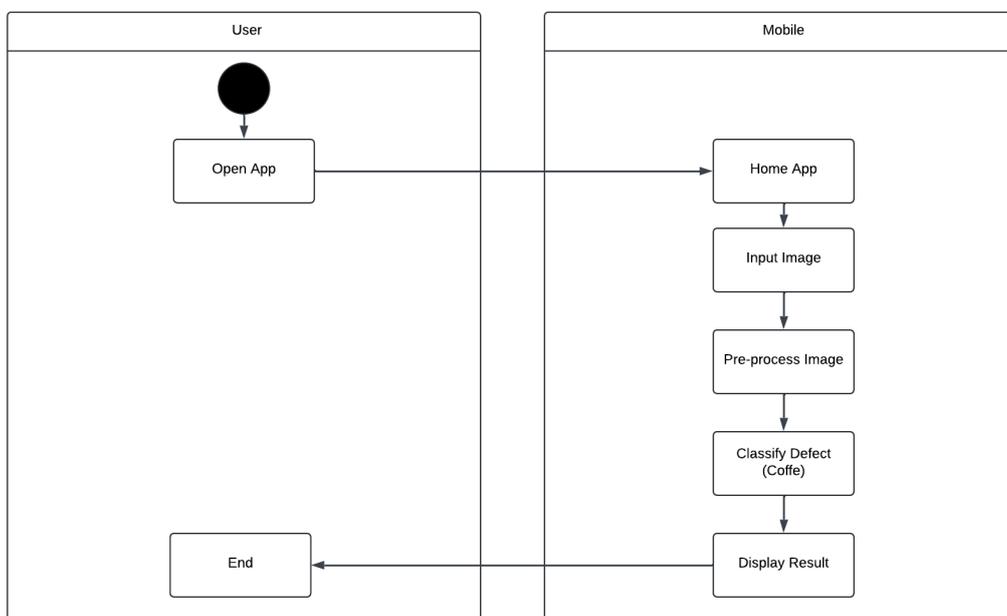
Gambar 3.6 Use Case Diagram

Diagram use case tersebut menggambarkan alur kerja klasifikasi gambar menggunakan algoritma K-Nearest Neighbors (KNN). Dimulai dari interaksi pengguna yang memilih dan mengunggah gambar ke sistem, gambar tersebut kemudian diproses melalui tahap prapemrosesan untuk persiapan ekstraksi fitur. Fitur-fitur yang diekstraksi dari gambar ini menjadi input untuk membangun model KNN. Model tersebut kemudian dilatih menggunakan data berlabel untuk mempelajari hubungan antara fitur dan labelnya. Setelah model siap, sistem dapat memprediksi

keluaran untuk gambar-gambar baru yang belum pernah dilihat sebelumnya, dan hasil klasifikasi ini ditampilkan kepada pengguna. Selain itu, hasilnya juga dapat diakses melalui platform mobile, menunjukkan fleksibilitas dalam penyampaian hasil.

### 3.7 Activity Diagram

Adapun *activity diagram* dari sistem ini adalah sebagai berikut : Gambar berikut merupakan *activity diagram* halaman sistem, setelah itu sistem akan mengarah ke halaman selanjutnya yaitu halaman utama



Gambar 3.7 Activity Diagram

Gambar diatas merupakan aktivitas diagram masuk ke halaman beranda aplikasi setelah itu masuk ke halaman selanjutnya. *Activity diagram* tersebut menggambarkan proses kerja aplikasi *mobile* untuk klasifikasi cacat pada biji kopi. Proses dimulai ketika pengguna membuka aplikasi, yang kemudian menampilkan halaman utama. Pengguna

menginput gambar biji kopi, yang selanjutnya diproses oleh sistem melalui tahap prapemrosesan untuk menyiapkan data. Setelah itu, aplikasi mengklasifikasikan cacat pada biji kopi berdasarkan gambar yang diinput dan menampilkan hasil klasifikasi kepada pengguna. Aktivitas berakhir ketika pengguna menutup aplikasi setelah melihat hasil. Diagram ini memperlihatkan alur interaksi antara pengguna dan sistem mobile, mulai dari awal hingga akhir penggunaan aplikasi.

## BAB IV

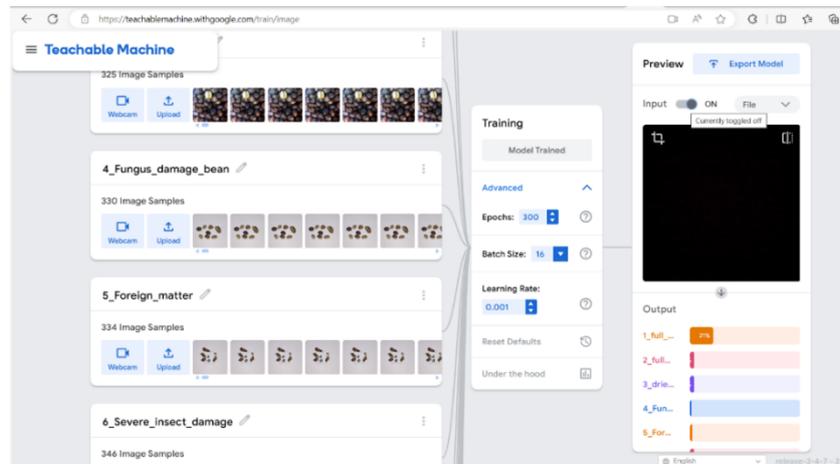
### HASIL DAN PEMBAHASAN

#### 4.1 Tahap Persiapan *Dataset*

Pada tahap ini data yang berupa gambar akan dilakukan proses *training* agar dapat menghasilkan model yang sesuai. Proses ini menggunakan sebuah *web-tools teachable machine* yakni sebuah aplikasi berbasis web yang digunakan untuk membuat model klasifikasi. Dan dataset yang digunakan adalah MNIST, yang terdiri dari gambar digit tulisan tangan berukuran 28x28 piksel. Kumpulan data ini terbagi menjadi tiga kategori: set pelatihan, set validasi, dan set pengujian. Data pelatihan dipakai untuk melatih model, sementara data validasi berfungsi untuk menguji model selama sesi pelatihan. Sedangkan data pengujian digunakan untuk menilai sejauh mana kinerja model setelah pelatihan selesai.

Gambar yang akan diklasifikasi berupa gambar *defect* pada kopi yang terbagi menjadi 16 class *Full Sour Bean, Full Black Bean, Dried Cherry Pod, Fungus Damage Bean, Foreign Matter, Severe Insect Damage, Partian Black Bean, Partial Sour Bean, Parchment Pergamino Bean, Floater Bean, Immature Unripe Bean, Whitered Bean, Shell, Broken Chipped Cut, Hull Husk, Slight Insect Damage*. Pengelompokan cacat pada kopi yang dikategorikan pada bentuk *defect*. Sehingga dapat membedakan objek gambar yang

akan dihasilkan. Berikut ini merupakan tampilan *web tools teachable machine* pada Gambar 4.1.



Gambar 4.1 Tampilan *Web Teachable machine*

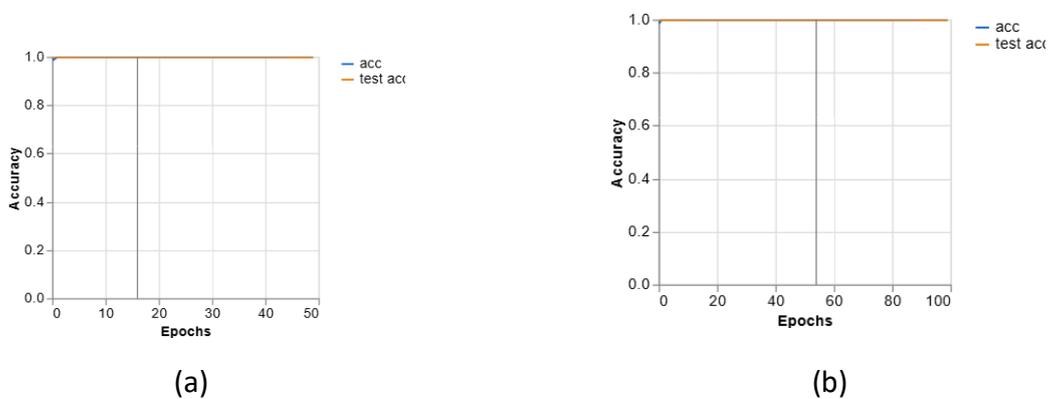
Berdasarkan Gambar 4.1 merupakan tampilan dari hasil *input dataset* cacat pada kopi yang akan di-*training* menggunakan *web tools teachable machine* pada proses ini *dataset* di-*training* dengan mempertimbangkan jenis *dataset*, *batch size* dan juga *learning rate*.

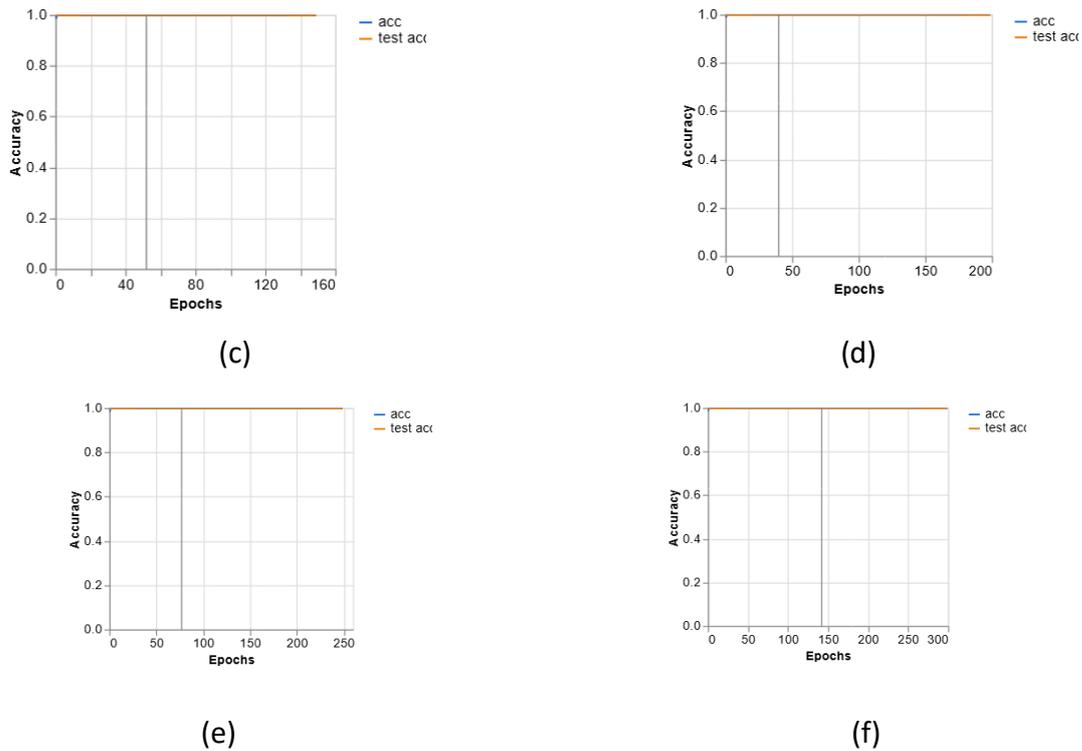
*Dataset* merupakan *hyperparameter* yang menentukan berapa kali algoritma *deep learning* bekerja melewati seluruh *dataset*, yang artinya satu *dataset* tercapai ketika semua batch telah dilewatkan melalui *neural network* satu kali. *Batch size* merupakan jumlah sampel *dataset* yang akan dilalui dalam satu waktu.

Pada *batch size* ini dapat menentukan jumlah sampel yang akan dikerjakan. *Learnig rate* merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*. Pada hal ini *learning rate* menjadi parameter seberapa cepat proses *training* yang akan dilakukan.

## 4.2 Pengaruh *Dataset* Terhadap Model

Pada tahap pengujian yang dilakukan dengan mengetahui baiknya sebuah model *neural network* yang dibuat dengan membandingkan *dataset* terhadap model sehingga dapat menjadi bahan acuan dalam menentukan tingkat akurasi model yang diinginkan. Pada penelitian ini digunakan enam kategori *epoch* yang berbeda yaitu 50, 100, 150, 200, 250, 300 dengan batch size 16 dengan *dataset* yang sama. Variasi *epoch* dilakukan untuk menemukan model *learning* terbaik dan ideal pada kasus penelitian cacat kopi ini. *Epoch* yang rendah berpotensi untuk mengalami kegagalan dalam pembelajaran model, sementara pada kasus *epoch* yang lebih tinggi berpotensi terjadinya *overfitting*. *Epoch* standar dalam *machine learning* adalah 100 iterasi, tetapi pada pembelajaran yang lebih berat penambahan hingga 500 *epoch* dapat dipertimbangkan. Hasil dari variasi *epoch* akan ditunjukkan dalam bentuk hasil *training* berupa grafik *train accuracy* dan *train testing* yang biasa disebut dengan *validation*. Model yang baik merupakan model yang mencapai tingkat konvergensi yang ditandai dengan tidak adanya perubahan nilai pada parameter akurasi mendekati 1. Hasil dari visualisasi proses *training* dengan variasi *epoch* ditunjukkan pada Gambar 4.2.





Gambar 4.2 Hasil Training *Dataset* per *Epoch*, (a) epoch 50, (b) epoch 100, (c) epoch 150, (d) epoch 200, (e) epoch 250, (f) epoch 300.

Pada *epoch* 50, model menunjukkan akurasi pelatihan yang tinggi, yang terlihat dari garis berwarna biru yang mencapai nilai 100%. Akurasi pengujiannya juga mencapai nilai yang sama, yaitu 100%. Hal ini menunjukkan bahwa tidak ada disparitas antara akurasi pelatihan dan pengujian pada *epoch* ini, menandakan bahwa model mampu menggeneralisasi pengetahuannya dengan baik sejak awal pelatihan. Model telah berhasil mengklasifikasikan data pelatihan dan pengujian dengan akurasi yang sempurna.

Pada *epoch* 100, tren serupa terlihat di mana akurasi pelatihan dan pengujian keduanya tetap pada nilai 100%. Tidak ada indikasi *overfitting* karena tidak ada perbedaan antara akurasi pelatihan dan pengujian. Ini menunjukkan bahwa model tetap stabil dan mampu mempertahankan performa optimal dalam mengklasifikasikan data pelatihan dan pengujian tanpa adanya penurunan performa atau generalisasi yang buruk.

Pada *epoch* 150, situasi tetap sama dengan akurasi pelatihan dan pengujian yang tetap pada nilai 100%. Ini menunjukkan bahwa model mampu mempertahankan performa optimal dalam pengklasifikasian data baik pada pelatihan maupun pengujian. Konsistensi ini menandakan bahwa model telah mencapai titik dimana ia dapat secara efektif belajar dari data pelatihan dan menerapkan pengetahuannya pada data pengujian dengan hasil yang sempurna.

Pada *epoch* 200, tidak ada perubahan dalam akurasi pelatihan dan pengujian yang keduanya tetap pada nilai 100%. Ini menandakan bahwa model telah mencapai kematangan dalam proses pembelajarannya dengan performa yang optimal. Pada tahap ini, model menunjukkan kemampuan generalisasi yang sangat baik, dengan perbedaan minimal antara akurasi pelatihan dan pengujian. Hal ini menunjukkan bahwa model telah stabil dan tidak memerlukan penyesuaian lebih lanjut untuk meningkatkan performanya.

Pada *epoch* 250 dan 300, model masih mempertahankan akurasi pelatihan dan pengujian pada nilai 100%. Ini menunjukkan bahwa model telah mencapai batas optimal dalam proses pelatihan, di mana penambahan epoch lebih lanjut tidak memberikan peningkatan performa yang signifikan. Model menunjukkan performa yang stabil dan akurat dalam klasifikasinya, dengan kemampuan generalisasi yang konsisten dan efisien.

Berdasarkan analisis performa di atas, model menunjukkan performa optimal dari awal hingga *epoch* 300, dengan akurasi pelatihan dan pengujian yang konsisten pada nilai 100%. Ini menandakan bahwa model mampu menggeneralisasi pengetahuan dari data pelatihan ke data pengujian dengan sangat baik tanpa adanya tanda-tanda *overfitting* atau

*underfitting*. Model telah mencapai efisiensi optimal dalam proses pembelajarannya sejak epoch awal.

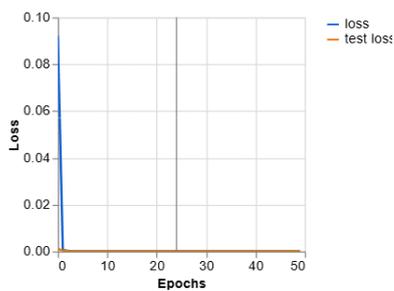
Gambar dari tabel (a) sampai (f) menunjukkan bahwa dari *epoch* 50 hingga 300, akurasi pelatihan dan pengujian selalu berada di angka 100%, yang berarti model secara konstan memberikan performa terbaiknya sepanjang pelatihan. Tidak ada perubahan signifikan atau disparitas yang terlihat, mengindikasikan bahwa model ini stabil dan sangat akurat dalam klasifikasinya sejak awal pelatihan. Hal ini menunjukkan bahwa model ini telah terlatih dengan baik dan mampu mengatasi kompleksitas data dengan efektif tanpa kehilangan kemampuan untuk melakukan generalisasi yang baik pada data yang belum pernah dilihat sebelumnya.

Secara keseluruhan, model ini menunjukkan performa yang sangat baik dan stabil sepanjang proses pelatihan, dengan kemampuan untuk mengklasifikasikan data pelatihan dan pengujian dengan akurasi sempurna. Ini mencerminkan kualitas yang sangat baik dari algoritma dan data yang digunakan dalam pelatihan model ini.

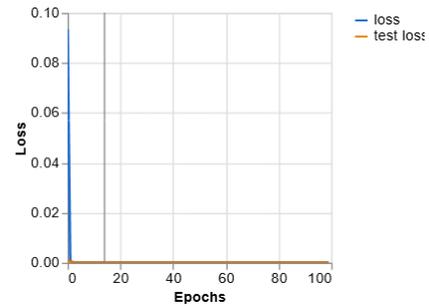
### **4.3 Pengaruh *Loss* Terhadap Model**

Pada pengujian yang dilakukan dengan mengetahui nilai *loss*, yakni nilai dari perhitungan *loss function* dari *training dataset* dan prediksi dari model. Pengujian ini akan terfokus pada konvergensi nilai *loss* model yang telah dilakukan proses training dengan *setting epoch* sebesar 50, 100, 150, 200, 250, 300. Konvergensi *loss* dapat diindikasikan dengan tidak adanya perubahan nilai *loss* secara fluktuatif dan nilai mendekati nilai 0.

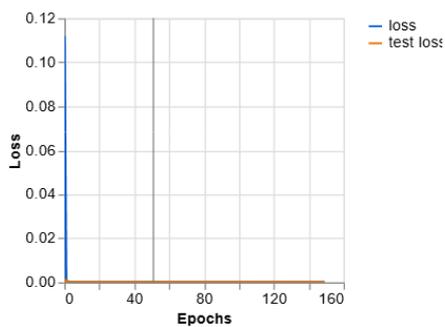
Variasi rendah pada nilai *epoch* akan memberikan waktu *training* yang lebih cepat, tetapi kemungkinan gagal *learning* yang cukup tinggi. Pada kasus variasi yang lebih tinggi dapat memberikan waktu untuk melakukan proses *training*, tetapi dengan membutuhkan waktu yang lebih lama. Pada Gambar 4.3 merupakan visualisasi hasil *training* dengan hasil *loss training*.



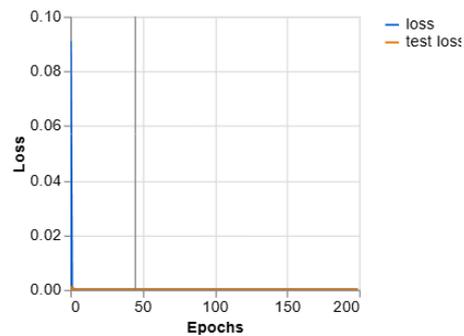
(a)



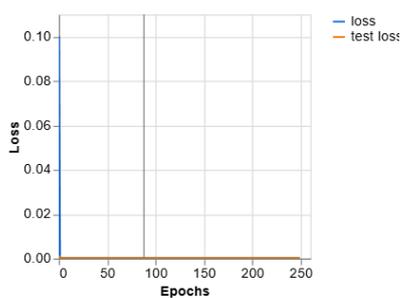
(b)



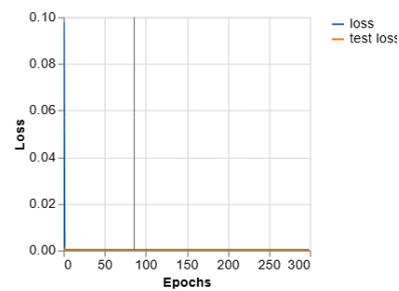
(c)



(d)



(d)



(e)

Gambar 4.3 Hasil *Loss Training*, (a) *Epoch* 50, (b) *Epoch* 100, (c) *Epoch* 150, (d) *Epoch* 200, (e) *Epoch* 250, (f) *Epoch* 300

Pada *epoch* 50 (Gambar (a)), *loss* pelatihan dan pengujian mulai dari nilai yang relatif tinggi tetapi cepat menurun mendekati nol. Dari visualisasi ini, dapat dilihat bahwa model belajar dengan cepat untuk meminimalkan kesalahan selama fase awal pelatihan. Pada titik ini, *loss* pelatihan (bergaris biru) dan *loss* pengujian (bergaris oranye) berada di sekitar nilai yang sangat rendah, menandakan bahwa model mampu mempelajari pola dari data pelatihan dengan baik dan mengaplikasikannya ke data pengujian dengan kesalahan yang minimal.

Pada *epoch* 100 (Gambar (b)), *loss* pelatihan dan pengujian tetap rendah setelah penurunan tajam di awal pelatihan. Nilai *loss* yang rendah secara konsisten menunjukkan bahwa model tidak hanya menghafal data pelatihan tetapi juga mampu menggeneralisasi pola yang ditemukan pada data baru dengan efisien. Tidak ada indikasi *overfitting* atau *underfitting* terlihat pada tahap ini karena nilai *loss* pelatihan dan pengujian keduanya sangat rendah dan mendekati nol.

Pada *epoch* 150 (Gambar (c)), tren yang sama terlihat, dengan *loss* pelatihan dan pengujian tetap pada nilai yang sangat rendah. Model menunjukkan stabilitas dalam kinerjanya, dengan tidak ada peningkatan signifikan dalam nilai *loss* setelah periode awal pelatihan. Ini menandakan bahwa model telah belajar untuk meminimalkan kesalahan dan mempertahankan performa optimal secara konsisten.

Pada *epoch* 200 (Gambar (d)), *loss* pelatihan dan pengujian masih berada pada nilai yang sangat rendah. Model telah mencapai kestabilan dalam proses pembelajarannya dan

menunjukkan kemampuan yang baik dalam meminimalkan kesalahan. Penurunan yang cepat di awal dan stabilitas yang terlihat setelahnya menunjukkan bahwa model mampu menangkap dan mempelajari pola data dengan cepat dan efektif.

Pada *epoch* 250 (Gambar (e)), tidak ada perubahan signifikan dalam nilai *loss* pelatihan dan pengujian. Kedua nilai *loss* tetap pada angka yang sangat rendah, menunjukkan bahwa model telah mencapai titik di mana penambahan *epoch* lebih lanjut tidak memberikan keuntungan signifikan dalam mengurangi kesalahan. Model telah mencapai performa optimal dalam hal meminimalkan kesalahan pada data pelatihan dan pengujian.

Pada *epoch* 300 (Gambar (f)), *loss* pelatihan dan pengujian masih tetap rendah. Ini menunjukkan bahwa model mempertahankan performa yang sangat baik dalam mengklasifikasikan data pelatihan dan pengujian dengan kesalahan yang minimal. Stabilitas ini menandakan bahwa model telah mencapai efisiensi optimal dalam proses pembelajarannya dan penambahan *epoch* lebih lanjut tidak akan memberikan peningkatan signifikan dalam performa.

Secara keseluruhan, dari *epoch* 50 hingga 300, model menunjukkan kemampuan yang sangat baik dalam meminimalkan *loss* baik pada data pelatihan maupun pengujian. Penurunan cepat pada nilai *loss* di awal diikuti oleh stabilitas yang konsisten menunjukkan bahwa model mampu belajar dengan cepat dan mempertahankan performa optimal tanpa adanya tanda-tanda *overfitting* atau *underfitting*. Model ini stabil dan sangat efektif dalam proses pembelajaran dan generalisasi, menunjukkan bahwa algoritma dan data yang digunakan sangat mendukung proses pelatihan yang efisien.

#### 4.4 Analisis Model *F1-Score*, *Accuracy*, *Precision*, dan *Recall*

Pada pembahasan ini analisis dari masing-masing penelitian ini dilakukan menggunakan *teachable machine*. Parameter yang digunakan sebagai analisis model ini ialah *F1-Score*, *accuracy*, *precision*, dan *recall* dari tiap-tiap *dataset*, selain itu terdapat juga parameter lainnya seperti *class* yang dibagi menjadi 16 yakni *Full Sour Bean*, *Full Black Bean*, *Dried Cherry Pod*, *Fungus Damage Bean*, *Foreign Matter*, *Severe Insect Damage*, *Partian Black Bean*, *Partial Sour Bean*, *Parchment Pergamino Bean*, *Floater Bean*, *Immature Unripe Bean*, *Whitered Bean*, *Shell*, *Broken Chipped Cut*, *Hull Husk*, *Slight Insect Damage*.

##### 4.4.1 Analisis *Object Classification Class 1 Full Sour Bean*

Pada penelitian ini menggunakan klasifikasi gambar pada kopi yang mempunyai *defect* yaitu *Full Sour Bean*. Hasil penelitian akan ditunjukkan menggunakan *confussion matrix* per objek dari klasifikasi gambarnya. Penelitian ini mengacu pada parameter dari *object classification class 1 Full Sour Bean*, oleh sebab itu parameter hasil dari *confussion matrix* didapatkan dari hasil training pada *web tools teachable machine* berdasarkan tabel 4.1 dibawah ini.

Tabel 4.1 *Object Classification Class 1 Full Sour Bean*

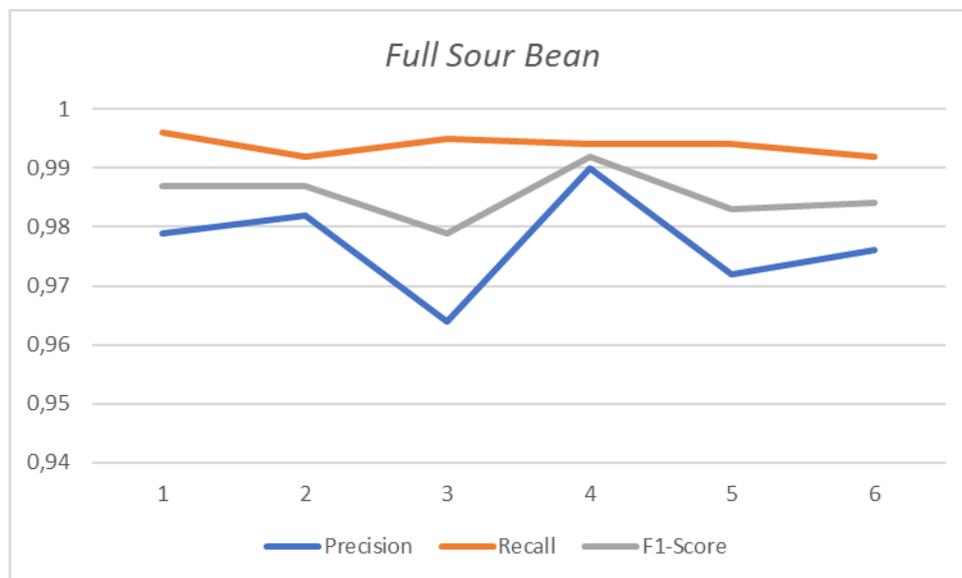
<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	97,85	0,979	0,996	0,987
100	98,26	0,982	0,992	0,987
150	96,48	0,964	0,995	0,979
200	98,94	0,990	0,994	0,992
250	97,30	0,972	0,994	0,983
300	97,70	0,976	0,992	0,984
$\bar{X}$	<b>97,75</b>	<b>0,977</b>	<b>0,993</b>	<b>0,968</b>

Berdasarkan Tabel 4.1 merupakan hasil pengukuran dari parameter *object*

*classification Full Sour Bean* untuk mengetahui *precision*, *recall* dan *f1-score*.

Berdasarkan tabel 4.1 tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian *Full Sour Bean* sebesar 97,75%

Evaluasi *F1-score* nilai rata-rata nya sebesar 0,968 menunjukkan standar performa yang cukup tinggi. Nilai rata-rata *precision* dan *recall* yang merupakan variabel yang berpengaruh terhadap *F1-score* itu sendiri sebesar 0,977 dan 0,993. Berdasarkan data Berdasarkan Tabel 4.1 dapat ditentukan grafik dari perbandingan sebagai berikut.



Gambar 4.4 Performa Model pada *Object Classification Class 1 Full Sour Bean*

Gambar 4.4 merupakan grafik performa dengan diperlihatkan *f1-score* dan juga *accuracy*. Berdasarkan Gambar 4.4 ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan memiliki rata-rata *accuracy* sebesar 0,9795, nilai rata-rata *precision* sebesar 0,977, nilai rata-rata *recall* sebesar 0,993,

dan nilai rata-rata *f1-score* sebesar 0,968, dari nilai rata-rata yang didapatkan pada penelitian ini, disimpulkan bahwa model yang diperoleh menggunakan *web-tools teachable machine* memiliki hasil *learning* dan *prediction* yang cukup baik, serta memiliki kehandalan dalam mendeteksi objek menggunakan kamera android, namun dalam proses pengambilan *dataset*-nya terdapat *noise* cahaya yang masuk tetapi bisa digunakan dalam mengenali objek.

Untuk menilai apakah nilai yang kita dapatkan benar, mari kita menghitung nilai dari *precision*, *recall* dan *f1 score* terlebih dahulu. Untuk perhitungan *epoch* 50 sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.979$$

$$TP = 0.979 \times (TP + FP)$$

$$TP = 0.979TP + 0.979FP$$

$$TP - 0.979TP = 0.979FP$$

$$0.021TP = 0.979FP$$

$$FP = \frac{0.021TP}{0.979}$$

$$FP = 0.0214TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.996$$

$$TP = 0.996 \times (TP + FN)$$

$$TP = 0.996TP + 0.996FN$$

$$TP - 0.996TP = 0.996FN$$

$$0.004TP = 0.996FN$$

$$FN = \frac{0.004TP}{0.996}$$

$$FN = 0.0040TP$$

### 1. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.979 \cdot 0.996}{0.979 + 0.996}$$

$$F1 - Score = 2 \cdot \frac{0.975564}{1.975}$$

$$F1 - Score = 2 \cdot 0.4933$$

$$F1 - Score = 0.9866$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	21
Negative	4	975

**Precision :**

$$Precision = \frac{1000}{1000 + 21} = \frac{1000}{1021} = 0.979$$

**Recall :**

$$Recall = \frac{1000}{1000 + 4} = \frac{1000}{1004} = 0.996$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.979 \cdot 0.996}{0.979 + 0.996} = 0.987$$

Untuk perhitungan *epoch* 100 dapat dilihat sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.982$$

$$TP = 0.982 \times (TP + FP)$$

$$TP = 0.982TP + 0.982FP$$

$$TP - 0.982TP = 0.982FP$$

$$0.018TP = 0.979FP$$

$$FP = \frac{0.018TP}{0.979}$$

$$FP = 0.0184TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.992$$

$$TP = 0.992 \times (TP + FN)$$

$$TP = 0.992TP + 0.992FN$$

$$TP - 0.992TP = 0.992FN$$

$$0.008TP = 0.992FN$$

$$FN = \frac{0.008TP}{0.992}$$

$$FN = 0.0081TP$$

## 2. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.982 \cdot 0.992}{0.982 + 0.992}$$

$$F1 - Score = 2 \cdot \frac{0.973464}{1.974}$$

$$F1 - Score = 2 \cdot 0.4933$$

$$F1 - Score = 0.9866$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	18
Negative	8	974

**Precision :**

$$Precision = \frac{1000}{1000 + 18} = \frac{1000}{1018} = 0.982$$

**Recall :**

$$Recall = \frac{1000}{1000 + 8} = \frac{1000}{1008} = 0.992$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.982 \cdot 0.992}{0.982 + 0.992} = 0.987$$

Untuk perhitungan pada *epoch* 150 dapat dilihat sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.964$$

$$TP = 0.964 \times (TP + FP)$$

$$TP = 0.964TP + 0.964FP$$

$$TP - 0.964TP = 0.964FP$$

$$0.036TP = 0.964FP$$

$$FP = \frac{0.036TP}{0.964}$$

$$FP = 0.0374TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.995$$

$$TP = 0.995 \times (TP + FN)$$

$$TP = 0.995TP + 0.995FN$$

$$TP - 0.995TP = 0.995FN$$

$$0.005TP = 0.995FN$$

$$FN = \frac{0.005TP}{0.995}$$

$$FN = 0.0050TP$$

### 3. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.964 \cdot 0.995}{0.964 + 0.995}$$

$$F1 - Score = 2 \cdot \frac{0.973464}{1.959}$$

$$F1 - Score = 2 \cdot 0.4896$$

$$F1 - Score = 0.979$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	37
Negative	5	458

**Precision :**

$$Precision = \frac{1000}{1000 + 37} = \frac{1000}{1037} = 0.964$$

**Recall :**

$$Recall = \frac{1000}{1000 + 5} = \frac{1000}{1005} = 0.995$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.964 \cdot 0.995}{0.964 + 0.995} = 0.979$$

Untuk perhitungan pada *epoch* 200 dapat dilihat sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.990$$

$$TP = 0.990 \times (TP + FP)$$

$$TP = 0.990TP + 0.990FP$$

$$TP - 0.990TP = 0.990FP$$

$$0.010TP = 0.990FP$$

$$FP = \frac{0.010TP}{0.990}$$

$$FP = 0.0101TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.994$$

$$TP = 0.994 \times (TP + FN)$$

$$TP = 0.994TP + 0.994FN$$

$$TP - 0.994TP = 0.994FN$$

$$0.006TP = 0.994FN$$

$$FN = \frac{0.006TP}{0.994}$$

$$FN = 0.0060TP$$

#### 4. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.990 \cdot 0.994}{0.990 + 0.994}$$

$$F1 - Score = 2 \cdot \frac{0.98406}{1.984}$$

$$F1 - Score = 2 \cdot 0.4969$$

$$F1 - Score = 0.9938$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	10
Negative	6	984

**Precision :**

$$Precision = \frac{1000}{1000 + 10} = \frac{1000}{1010} = 0.990$$

**Recall :**

$$Recall = \frac{1000}{1000 + 6} = \frac{1000}{1006} = 0.994$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.990 \cdot 0.994}{0.990 + 0.994} = 0.992$$

Untuk perhitungan pada *epoch* 250 dapat dilihat sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.972$$

$$TP = 0.972 \times (TP + FP)$$

$$TP = 0.972TP + 0.972FP$$

$$TP - 0.972TP = 0.972FP$$

$$0.028TP = 0.972FP$$

$$FP = \frac{0.028TP}{0.972}$$

$$FP = 0.0288TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.994$$

$$TP = 0.994 \times (TP + FN)$$

$$TP = 0.994TP + 0.994FN$$

$$TP - 0.994TP = 0.994FN$$

$$0.006TP = 0.994FN$$

$$FN = \frac{0.006TP}{0.994}$$

$$FN = 0.0060TP$$

## 2. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.972 \cdot 0.994}{0.972 + 0.994}$$

$$F1 - Score = 2 \cdot \frac{0.964728}{1.984}$$

$$F1 - Score = 2 \cdot 0.4901$$

$$F1 - Score = 0.9802$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	29
Negative	6	1465

**Precision :**

$$Precision = \frac{1000}{1000 + 29} = \frac{1000}{1029} = 0.972$$

**Recall :**

$$Recall = \frac{1000}{1000 + 6} = \frac{1000}{1006} = 0.994$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.972 \cdot 0.994}{0.972 + 0.994} = 0.983$$

Untuk perhitungan pada *epoch* 300 dapat dilihat sebagai berikut:

1. Menyusun Sistem Persamaan

- **Precision (P):**

$$Precision = \frac{TP}{TP + FP} = 0.976$$

$$TP = 0.976 \times (TP + FP)$$

$$TP = 0.976TP + 0.976FP$$

$$TP - 0.976TP = 0.976FP$$

$$0.024TP = 0.976FP$$

$$FP = \frac{0.024TP}{0.976}$$

$$FP = 0.0246TP$$

- **Recall (R):**

$$Recall = \frac{TP}{TP + FN} = 0.992$$

$$TP = 0.992 \times (TP + FN)$$

$$TP = 0.992TP + 0.992FN$$

$$TP - 0.992TP = 0.992FN$$

$$0.008TP = 0.992FN$$

$$FN = \frac{0.008TP}{0.992}$$

$$FN = 0.0081TP$$

### 3. Verifikasi *F1-Score*

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$F1 - Score = 2 \cdot \frac{0.976 \cdot 0.992}{0.976 + 0.992}$$

$$F1 - Score = 2 \cdot \frac{0.970752}{1.968}$$

$$F1 - Score = 2 \cdot 0.4926$$

$$F1 - Score = 0.9852$$

Untuk Confusion Matrix yang kita dapatkan sebelumnya adalah sebagai berikut :

Predicted Actual	Positive	Negative
Positive	1000	25
Negative	8	1967

**Precision :**

$$Precision = \frac{1000}{1000 + 25} = \frac{1000}{1025} = 0.976$$

**Recall :**

$$Recall = \frac{1000}{1000 + 8} = \frac{1000}{1008} = 0.992$$

**F1-Score :**

$$F1 - Score = 2 \cdot \frac{0.976 \cdot 0.992}{0.976 + 0.992} = 0.984$$

#### 4.4.2 Analisis Object Classification Class 2 Full Black Bean

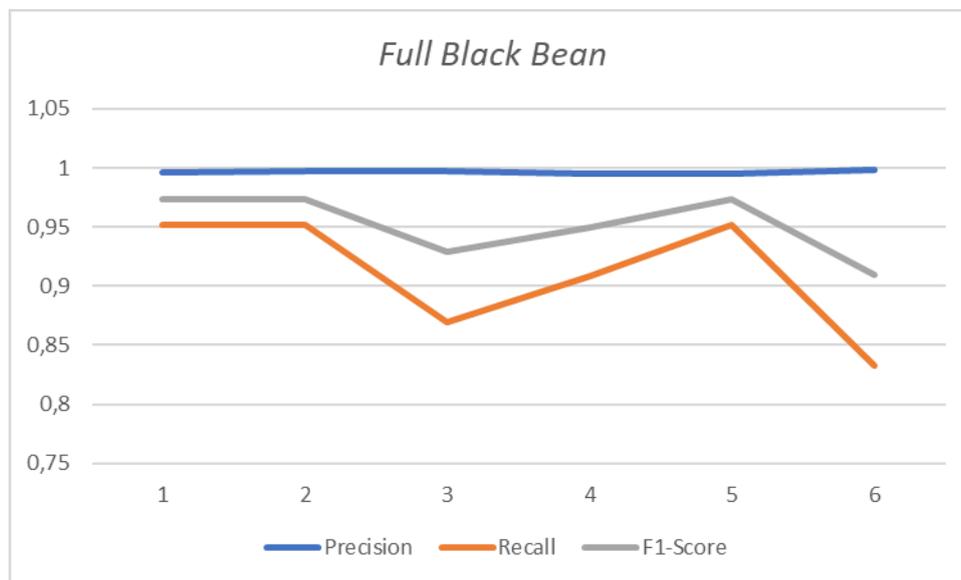
Pada analisis selanjutnya yang mengacu pada parameter dari *object classification class 2 Full Black Bean*. Pada pembahasan ini menggunakan parameter hasil dari *confussion matrix* yang didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.2 di bawah ini.

Tabel 4.2 *Object Classification Class 2 Full Black Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	99,63	0,996	0,952	0,974
100	99,76	0,997	0,952	0,974
150	99,79	0,997	0,869	0,929
200	99,58	0,995	0,908	0,950
250	99,53	0,995	0,952	0,973
300	99,81	0,998	0,833	0,909
$\bar{X}$	<b>99,85</b>	<b>0,996</b>	<b>0,911</b>	<b>0,951</b>

Berdasarkan Tabel 4.2 merupakan hasil pengukuran dari parameter *object classification Full Black Bean* untuk mengetahui *precision*, *recall* dan *f1-score*.

Berdasarkan Tabel 4.2 tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada sampel *Full Black Bean* sebesar 99,85%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,951, untuk nilai rata-rata *precision* dan *recall* yang merupakan nilai harmonik dari *f1-score* itu sendiri sebesar 0,996 dan 0,911. Berdasarkan data Berdasarkan Tabel 4.2 dapat ditentukan grafik dari perbandingan yang ditunjukkan pada Gambar 4.5 berikut.



Gambar 4.5 Performa Model pada *Object Classification Class 2 Full Black Bean*

Pada Gambar 4.5 merupakan grafik performa dengan perbandingan *f1-score* dan juga *accuracy*. Berdasarkan Gambar 4.5 hasil dari performa model dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*. Dari keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9985, nilai rata-rata *precision* sebesar 0,996, nilai rata-rata recall sebesar 0,911, dan nilai rata-rata *f1-score* sebesar 0,951.

Berdasarkan nilai rata-rata yang didapatkan pada percobaan ini dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik. Serta memiliki keandalan dalam mendeteksi objek menggunakan kamera android, namun dalam proses pengambilan *dataset* nya terdapat noise cahaya yang masuk tetapi bisa digunakan dalam mengenali objek.

#### 4.4.3 Analisis *Object Classification Class 3 Dried Cherry Pod*

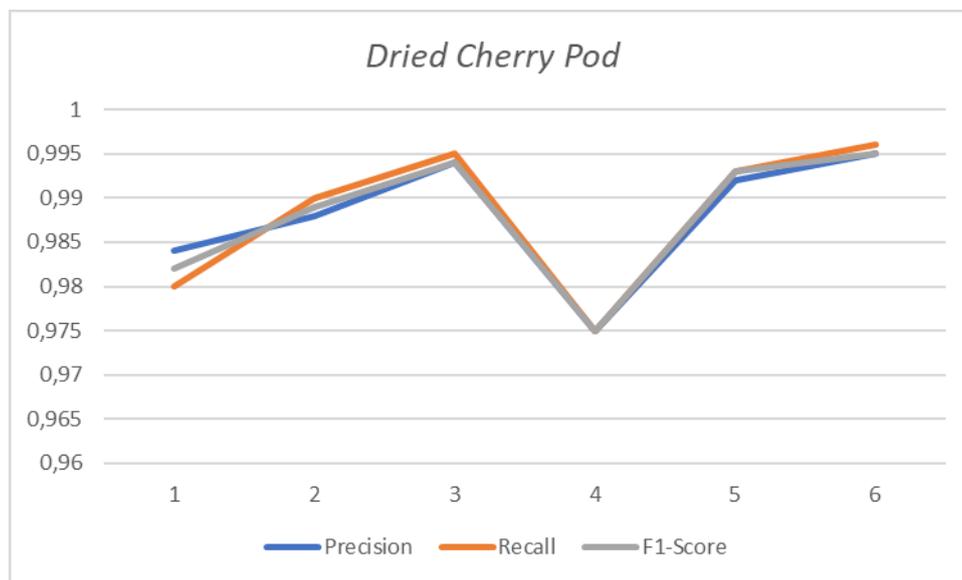
Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 3 Dried Cherry Pod*, pada parameter ini hampir sama dengan analisis sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.3 dibawah ini.

Tabel 4.3 *Object Classification Class 3 Dried Cherry Pod*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	98,39	0,984	0,980	0,982
100	98,79	0,988	0,990	0,989
150	99,47	0,994	0,995	0,994
200	97,47	0,975	0,975	0,975
250	99,31	0,992	0,993	0,993
300	99,55	0,995	0,996	0,995
<b>X</b>	<b>98,83</b>	<b>0,988</b>	<b>0,988</b>	<b>0,988</b>

Berdasarkan Tabel 4.3 merupakan hasil dari pengukuran dari parameter *object classification* pada class 3 untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada

model bagian sample *Dried Cherry Pod* sebesar 98,83%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,988, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,988 dan 0,988. Berdasarkan data Berdasarkan Tabel 4.3 dapat ditentukan grafik dari perbandingan pada Gambar 4.6.



Gambar 4.6 Performa Model pada *Object Classification Class 3 Dried Cherry Pod*

Gambar 4.6 merupakan grafik performa dengan perbandingan *f1-score* dan juga *accuracy* pada *object classification class 3*. Berdasarkan gambar di atas hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan nilai *accuracy*.

Berdasarkan dari keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9883, nilai rata-rata *precision* sebesar 0,988, nilai rata-rata *recall* sebesar 0,988, dan nilai rata-rata *f1-score* sebesar 0,988.

Nilai yang didapatkan pada penelitian ini dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik. Keandalan dalam mendeteksi objek menggunakan kamera Android memiliki reliabilitas yang cukup, tetapi faktor pencahayaan dapat menjadi penurunan keandalan algoritma.

#### 4.4.4 Analisis *Object Classification Class 4 Fungus Damage Bean*

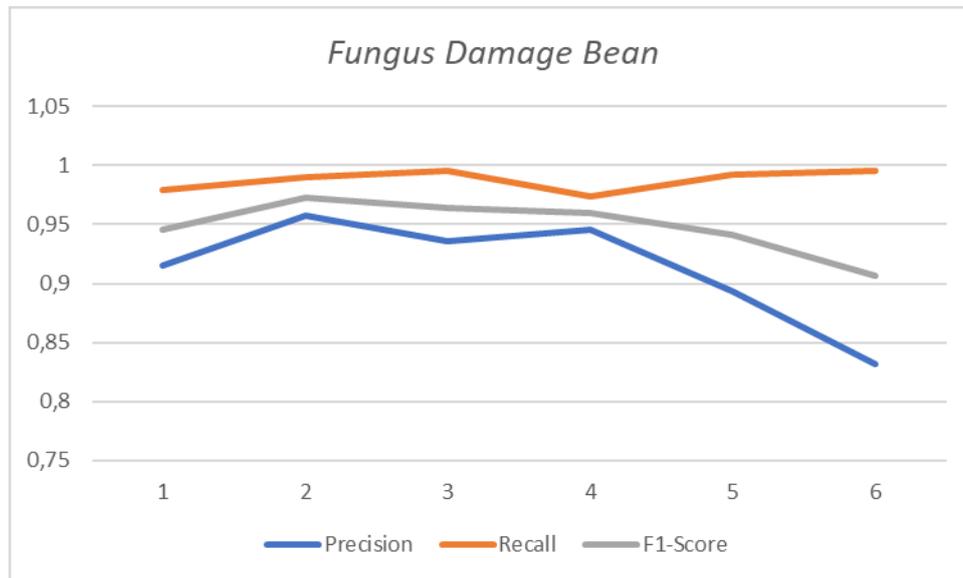
Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 4 Fungus Damage Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.4 dibawah ini.

Tabel 4.4 *Object Classification Class 4 Fungus Damage Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	91,47	0,915	0,979	0,946
100	95,79	0,957	0,990	0,973
150	93,64	0,936	0,995	0,964
200	94,65	0,946	0,974	0,960
250	89,46	0,894	0,992	0,941
300	83,18	0,832	0,995	0,906
$\bar{X}$	<b>91,43</b>	<b>0,913</b>	<b>0,987</b>	<b>0,948</b>

Berdasarkan Tabel 4.4 merupakan hasil dari pengukuran dari parameter object classification pada class 4 untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk kaki sebesar 91,43%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,948. Untuk nilai rata-rata *precision* dan recall ialah sebesar 0,987 dan 0,948. Berdasarkan data Berdasarkan Tabel 4.4 dapat ditentukan grafik dari

perbandingan pada Gambar 4.7.



Gambar 4.7 Performa Model pada *Object Classification Class 4 Fungus Damage Bean*

Gambar 4.7 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9143, nilai rata-rata *precision* sebesar 0,913, nilai rata-rata *recall* sebesar 0,987, dan nilai rata-rata *f1-score* sebesar 0,948. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

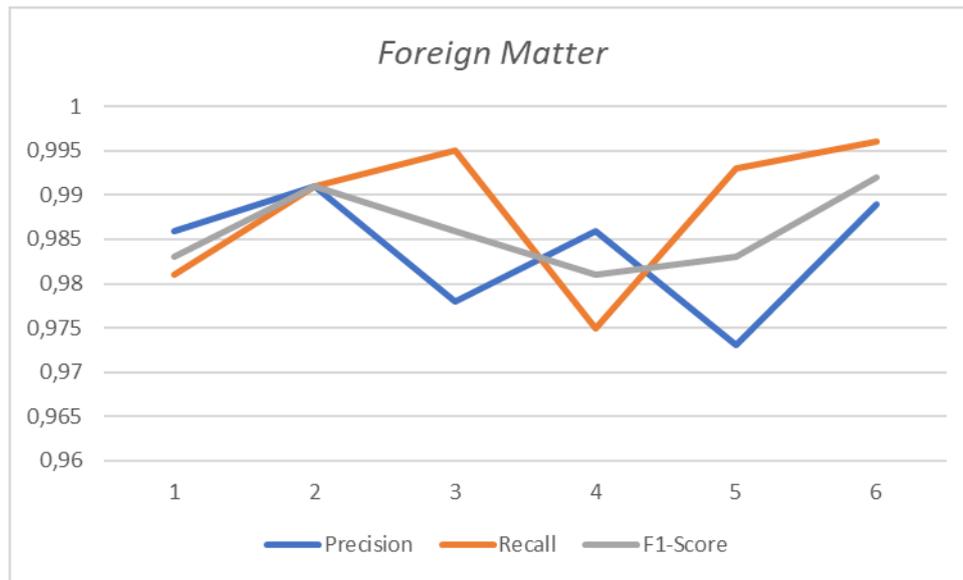
#### 4.4.5 Analisis *Object Classification Class 5 Busuk Pucuk*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 5 Busuk Pucuk*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.5 dibawah ini.

Tabel 4.5 *Object Classification Class 5 Foreign Matter*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	98,54	0,986	0,980	0,983
100	99,13	0,991	0,990	0,991
150	97,82	0,978	0,995	0,986
200	98,65	0,986	0,975	0,981
250	97,43	0,973	0,993	0,983
300	98,91	0,989	0,996	0,992
$\bar{X}$	<b>98,41</b>	<b>0,983</b>	<b>0,986</b>	<b>0,988</b>

Berdasarkan Tabel 4.5 merupakan hasil dari pengukuran dari parameter object classification pada class 4 untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 98,41%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,988. Untuk nilai rata-rata *precision* dan recall ialah sebesar 0,983 dan 0,986. Berdasarkan data Berdasarkan Tabel 4.5 dapat ditentukan grafik dari perbandingan pada Gambar 4.8.



Gambar 4.8 Performa Model pada *Object Classification Class 5 Foreign Matter*

Gambar 4.8 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9841, nilai rata-rata *precision* sebesar 0,983, nilai rata-rata *recall* sebesar 0,986, dan nilai rata-rata *f1-score* sebesar 0,988. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

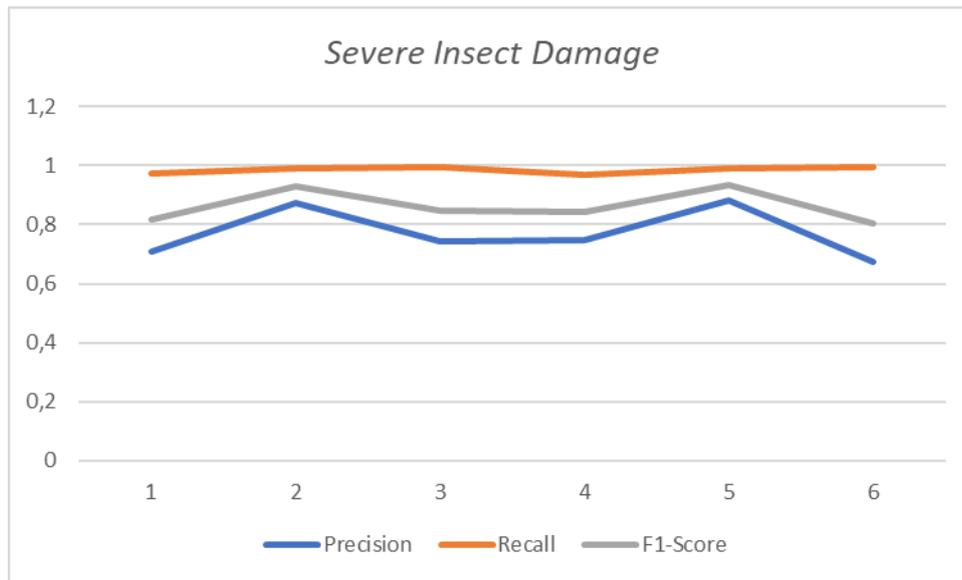
#### 4.4.6 Analisis *Object Classification Class 6 Severe Insect Damage*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 6 Severe Insect Damage*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.6 dibawah ini.

Tabel 4.6 *Object Classification Class 6 Severe Insect Damage*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	70,79	0,707	0,972	0,819
100	87,50	0,874	0,989	0,928
150	74,37	0,743	0,993	0,848
200	74,78	0,747	0,968	0,844
250	88,45	0,883	0,992	0,934
300	67,22	0,672	0,994	0,804
$\bar{X}$	<b>77,16</b>	<b>0,771</b>	<b>0,984</b>	<b>0,862</b>

Berdasarkan Tabel 4.6 merupakan hasil dari pengukuran dari parameter object classification pada class 6 untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample *Severe Insect Damage* sebesar 77,16%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,862. Untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,771 dan 0,984. Berdasarkan data Berdasarkan Tabel 4.6 dapat ditentukan grafik dari perbandingan pada Gambar 4.9.



Gambar 4.9 Performa Model pada *Object Classification Class 6 Severe Insect Damage*

Gambar 4.9 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,7716, nilai rata-rata *precision* sebesar 0,771, nilai rata-rata *recall* sebesar 0,984, dan nilai rata-rata *f1-score* sebesar 0,862. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

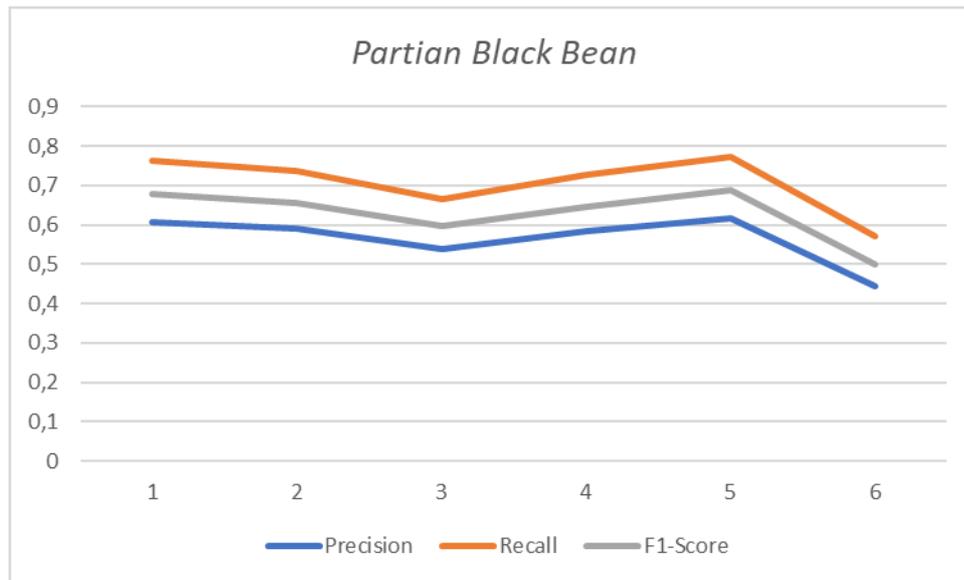
#### 4.4.7 Analisis Object Classification Class 7 Partial Black Bean

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 7 Partial Black Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.7 dibawah ini.

Tabel 4.7 *Object Classification Class 7 Partial Black Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	66,01	0,608	0,762	0,677
100	64,66	0,592	0,738	0,657
150	60,26	0,538	0,667	0,596
200	64,11	0,584	0,726	0,647
250	66,65	0,617	0,774	0,688
300	50,64	0,445	0,571	0,500
<b>X</b>	<b>62,05</b>	<b>0,564</b>	<b>0,706</b>	<b>0,627</b>

Berdasarkan Tabel 4.7 merupakan hasil dari pengukuran dari parameter object *classification* pada *class 7* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 62,05%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,627. Untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,706 dan 0,564. Berdasarkan data Berdasarkan Tabel 4.7 dapat ditentukan grafik dari perbandingan pada Gambar 4.10.



Gambar 4.10 Performa Model pada *Object Classification Class 7 Partial Black Bean*

Gambar 4.10 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,6205, nilai rata-rata *precision* sebesar 0,564, nilai rata-rata *recall* sebesar 0,706, dan nilai rata-rata *f1-score* sebesar 0,627. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

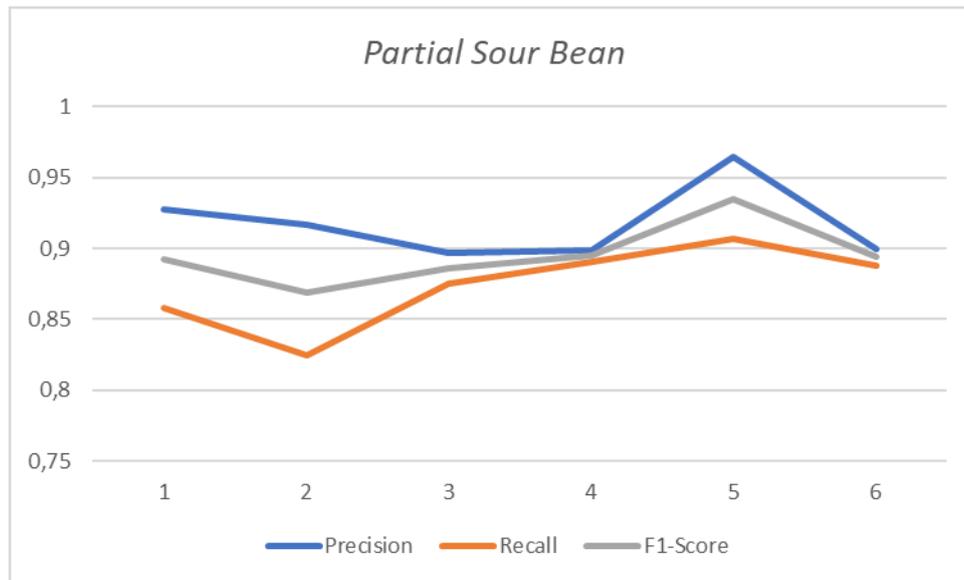
#### 4.4.8 Analisis *Object Classification Class 8 Partial Sour Bean*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 8 Partial Sour Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.8 dibawah ini.

Tabel 4.8 *Object Classification Class 8 Partial Sour Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	93,31	0,928	0,858	0,892
100	91,53	0,917	0,825	0,869
150	93,89	0,897	0,875	0,886
200	95,59	0,899	0,891	0,895
250	96,77	0,965	0,907	0,935
300	95,43	0,900	0,888	0,894
$\bar{X}$	<b>94,42</b>	<b>0,917</b>	<b>0,874</b>	<b>0,895</b>

Berdasarkan Tabel 4.8 merupakan hasil dari pengukuran dari parameter object *classification* pada *class 8* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 94,42%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,895, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,917 dan 0,874. Berdasarkan data Berdasarkan Tabel 4.8 dapat ditentukan grafik dari perbandingan pada Gambar 4.11.



Gambar 4.11 Performa Model pada *Object Classification Class 8 Partial Sour Bean*

Gambar 4.11 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9422, nilai rata-rata *precision* sebesar 0,917, nilai rata-rata *recall* sebesar 0,874, dan nilai rata-rata *f1-score* sebesar 0,895. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

#### 4.4.9 Analisis Object Classification Class 9 Parchment Pergamino Bean

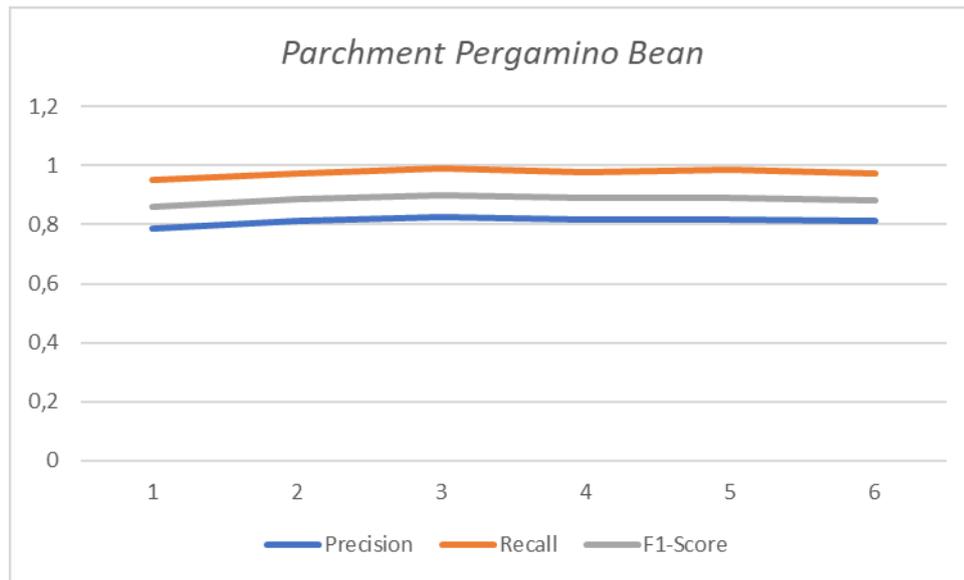
Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 9 Parchment Pergamino Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.9 dibawah ini.

Tabel 4.9 *Object Classification Class 9 Parchment Pergamino Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	89,69	0,788	0,951	0,862
100	93,85	0,814	0,974	0,887
150	96,51	0,826	0,989	0,900
200	94,62	0,817	0,978	0,891
250	94,73	0,815	0,984	0,892
300	93,49	0,810	0,970	0,883
$\bar{X}$	<b>93,98</b>	<b>0,811</b>	<b>0,974</b>	<b>0,885</b>

Berdasarkan Tabel 4.9 merupakan hasil dari pengukuran dari parameter object *classification* pada *class 9* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 93,98%, se

dangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,885, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,811 dan 0,974. Berdasarkan data Berdasarkan Tabel 4.9 dapat ditentukan grafik dari perbandingan pada Gambar 4.12.



Gambar 4.12 Performa Model pada *Object Classification Class 9 Parchment Pergamino Bean*

Gambar 4.12 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9398, nilai rata-rata *precision* sebesar 0,811, nilai rata-rata *recall* sebesar 0,974, dan nilai rata-rata *f1-score* sebesar 0,885. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

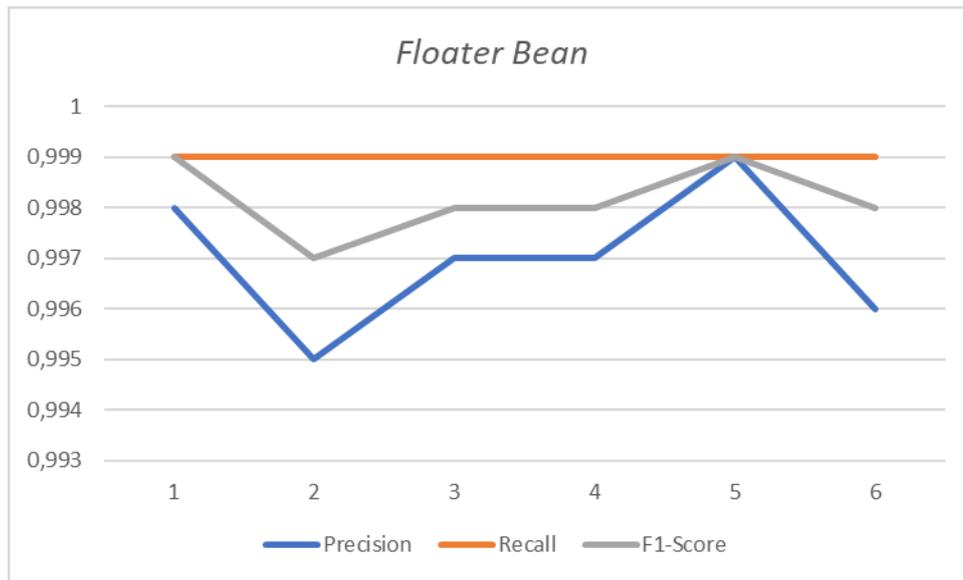
#### 4.4.10 Analisis *Object Classification Class 10 Floater Bean*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 10 Floater Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.10 dibawah ini.

Tabel 4.10 *Object Classification Class 10 Floater Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	99,71	0,998	0,999	0,999
100	99,58	0,995	0,999	0,997
150	99,74	0,997	0,999	0,998
200	99,71	0,997	0,999	0,998
250	99,79	0,999	0,999	0,999
300	99,69	0,996	0,999	0,998
<b>X</b>	<b>99,70</b>	<b>0,997</b>	<b>0,999</b>	<b>0,998</b>

Berdasarkan Tabel 4.10 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 10* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 99,70%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,998, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,997 dan 0,999. Berdasarkan data Berdasarkan Tabel 4.10 dapat ditentukan grafik dari perbandingan pada Gambar 4.13.



Gambar 4.13 Performa Model pada *Object Classification Class 10 Floater Bean*

Gambar 4.13 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9970, nilai rata-rata *precision* sebesar 0,997, nilai rata-rata *recall* sebesar 0,999, dan nilai rata-rata *f1-score* sebesar 0,998. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

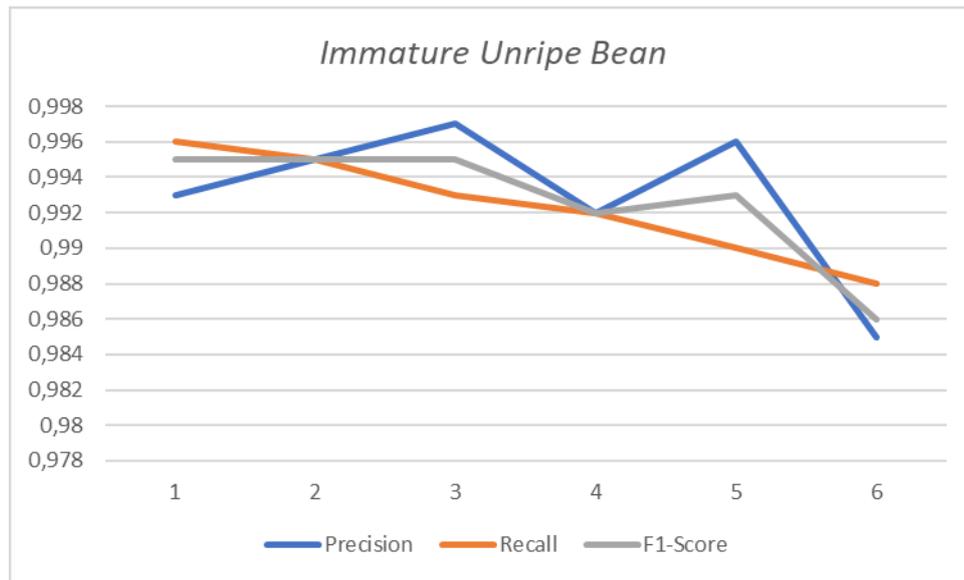
#### 4.4.11 Analisis *Object Classification Class 11 Immature Unripe Bean*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 11 Immature Unripe Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari confusion matrix didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.11 dibawah ini.

Tabel 4.11 *Object Classification Class 11 Immature Unripe Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	98,92	0,993	0,996	0,995
100	98,98	0,995	0,995	0,995
150	99,07	0,997	0,993	0,995
200	98,35	0,992	0,992	0,992
250	98,44	0,996	0,990	0,993
300	99,10	0,985	0,988	0,986
$\bar{X}$	<b>98,97</b>	<b>0,993</b>	<b>0,992</b>	<b>0,992</b>

Berdasarkan Tabel 4.11 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 11* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 98,97%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,992, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,993 dan 0,992. Berdasarkan data Berdasarkan Tabel 4.11 dapat ditentukan grafik dari perbandingan pada Gambar 4.14.



Gambar 4.14 Performa Model pada *Object Classification Class 11 Immature Unripe Bean*

Gambar 4.14 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9897, nilai rata-rata *precision* sebesar 0,993, nilai rata-rata *recall* sebesar 0,992, dan nilai rata-rata *f1-score* sebesar 0,992. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

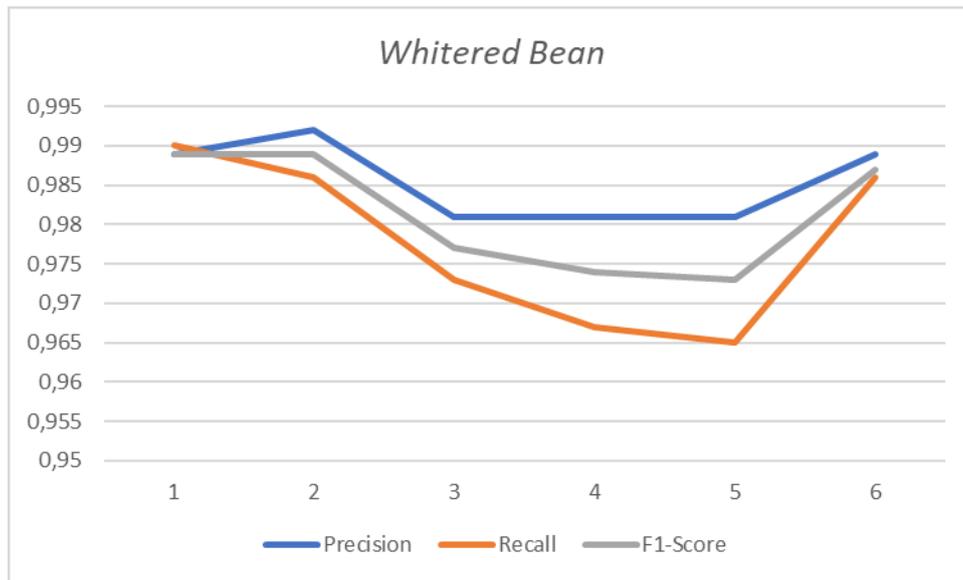
#### 4.4.12 Analisis *Object Classification Class 12 Whitered Bean*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 12 Whitered Bean*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.12 dibawah ini.

Tabel 4.12 *Object Classification Class 12 Whitered Bean*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	98,40	0,989	0,990	0,989
100	98,80	0,992	0,986	0,989
150	98,17	0,981	0,973	0,977
200	98,03	0,981	0,967	0,974
250	98,17	0,981	0,965	0,973
300	98,90	0,989	0,986	0,987
$\bar{X}$	<b>98,37</b>	<b>0,985</b>	<b>0,977</b>	<b>0,981</b>

Berdasarkan Tabel 4.12 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 12* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 98,37%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,981, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,985 dan 0,977. Berdasarkan data Berdasarkan Tabel 4.12 dapat ditentukan grafik dari perbandingan pada Gambar 4.15.



Gambar 4.15 Performa Model pada *Object Classification Class 12 Whitered Bean*

Gambar 4.15 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9837, nilai rata-rata *precision* sebesar 0,985, nilai rata-rata *recall* sebesar 0,977, dan nilai rata-rata *f1-score* sebesar 0,981. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

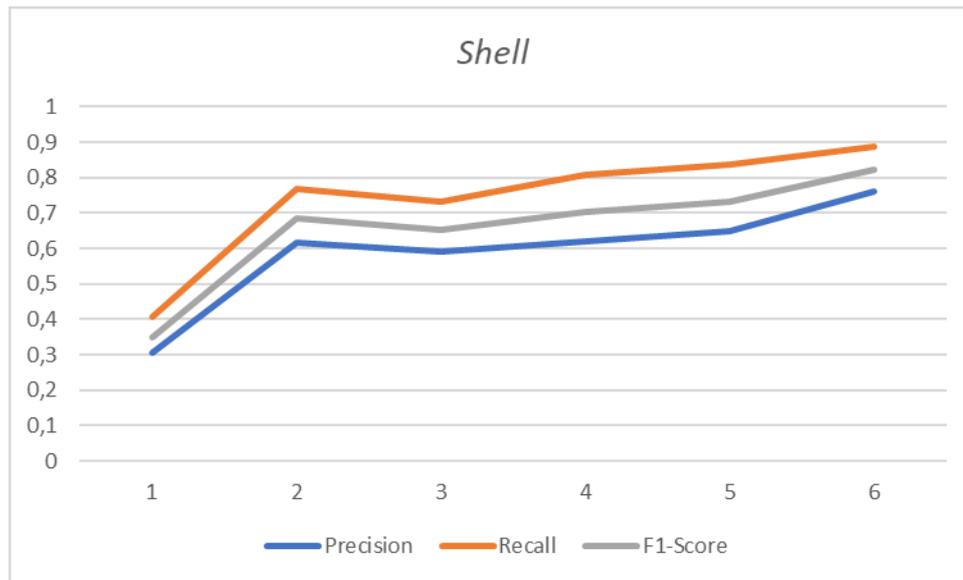
#### 4.4.13 Analisis *Object Classification Class 13 Shell*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 13 Shell*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.13 dibawah ini.

Tabel 4.13 *Object Classification Class 13 Shell*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	48.73	0,305	0,408	0,349
100	67.02	0,617	0,769	0,684
150	62.98	0,592	0,731	0,654
200	71.57	0,622	0,808	0,702
250	75.99	0,649	0,837	0,731
300	87.99	0,762	0,889	0,821
$\bar{X}$	<b>69,04</b>	<b>0,591</b>	<b>0,740</b>	<b>0,823</b>

Berdasarkan Tabel 4.13 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 13* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 69,04%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,823, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,591 dan 0,740. Berdasarkan data Berdasarkan Tabel 4.13 dapat ditentukan grafik dari perbandingan pada Gambar 4.16.



Gambar 4.16 Performa Model pada *Object Classification Class 13 Shell*

Gambar 4.16 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,6904, nilai rata-rata *precision* sebesar 0,591, nilai rata-rata *recall* sebesar 0,740, dan nilai rata-rata *f1-score* sebesar 0,823. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

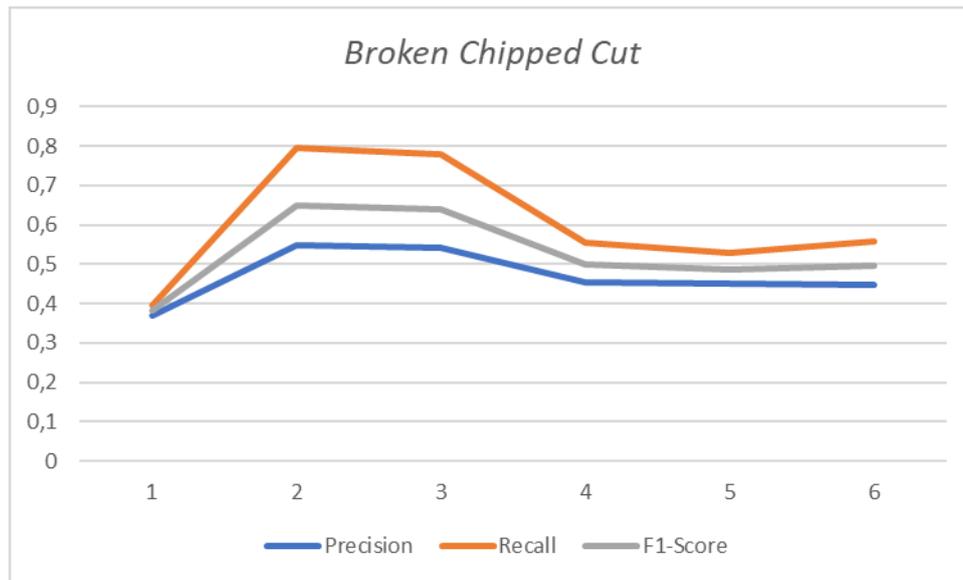
#### 4.4.14 Analisis *Object Classification Class 14 Broken Chipped Cut*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 14 Broken Chipped Cut*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.14 dibawah ini.

Tabel 4.14 *Object Classification Class 14 Broken Chipped Cut*

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	48.21	0,371	0,394	0,382
100	71.08	0,548	0,794	0,648
150	69.31	0,541	0,778	0,640
200	56.13	0,455	0,556	0,501
250	53.56	0,450	0,528	0,486
300	55.47	0,449	0,557	0,497
$\bar{X}$	<b>58,96</b>	<b>0,469</b>	<b>0,767</b>	<b>0,525</b>

Berdasarkan Tabel 4.14 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 14* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 58,96%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,525, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,469 dan 0,767. Berdasarkan data Berdasarkan Tabel 4.14 dapat ditentukan grafik dari perbandingan pada Gambar 4.17.



Gambar 4.17 Performa Model pada *Object Classification Class 14 Broken Chipped Cut*

Gambar 4.17 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,5896, nilai rata-rata *precision* sebesar 0,469, nilai rata-rata *recall* sebesar 0,767, dan nilai rata-rata *f1-score* sebesar 0,525. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang cukup baik.

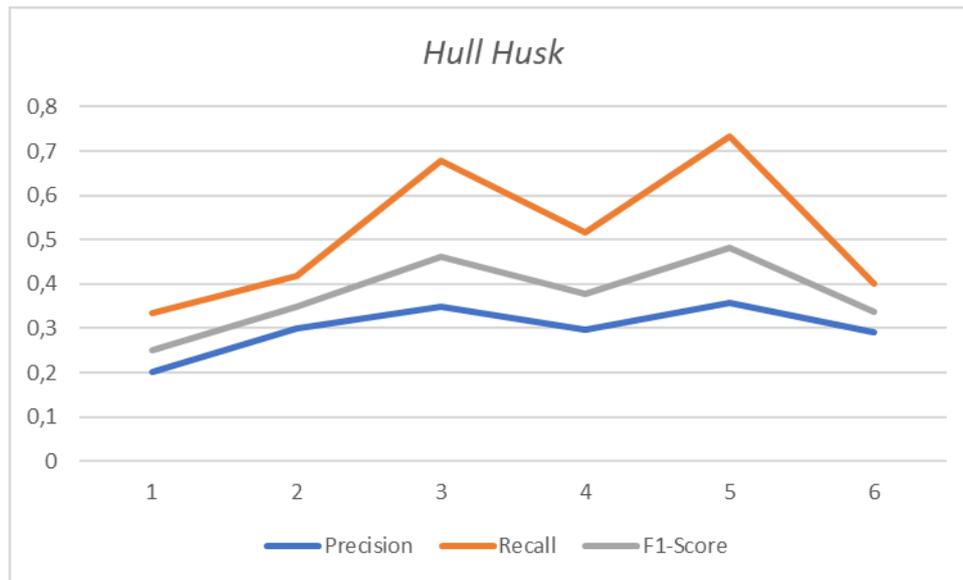
#### 4.4.15 Analisis *Object Classification Class 15 Hull Husk*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 15 Hull Husk*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.15 dibawah ini.

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	34,67	0,200	0,333	0,250
100	49,59	0,299	0,417	0,348
150	70,22	0,349	0,677	0,462
200	55,18	0,297	0,516	0,378
250	72,72	0,358	0,733	0,482
300	47,19	0,291	0,400	0,336
$\bar{X}$	<b>54,92</b>	<b>0,299</b>	<b>0,512</b>	<b>0,376</b>

Tabel 4.15 *Object Classification Class 15 Hull Husk*

Berdasarkan Tabel 4.15 merupakan hasil dari pengukuran dari parameter object *classification* pada *class 15* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 54,92%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,376, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 0,299 dan 0,512. Berdasarkan data Berdasarkan Tabel 4.15 dapat ditentukan grafik dari perbandingan pada Gambar 4.18.



Gambar 4.18 Performa Model pada *Object Classification Class 15 Hull Husk*

Gambar 4.18 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,5492, nilai rata-rata *precision* sebesar 0,229, nilai rata-rata *recall* sebesar 0,512, dan nilai rata-rata *f1-score* sebesar 0,376. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang sedikit cukup baik.

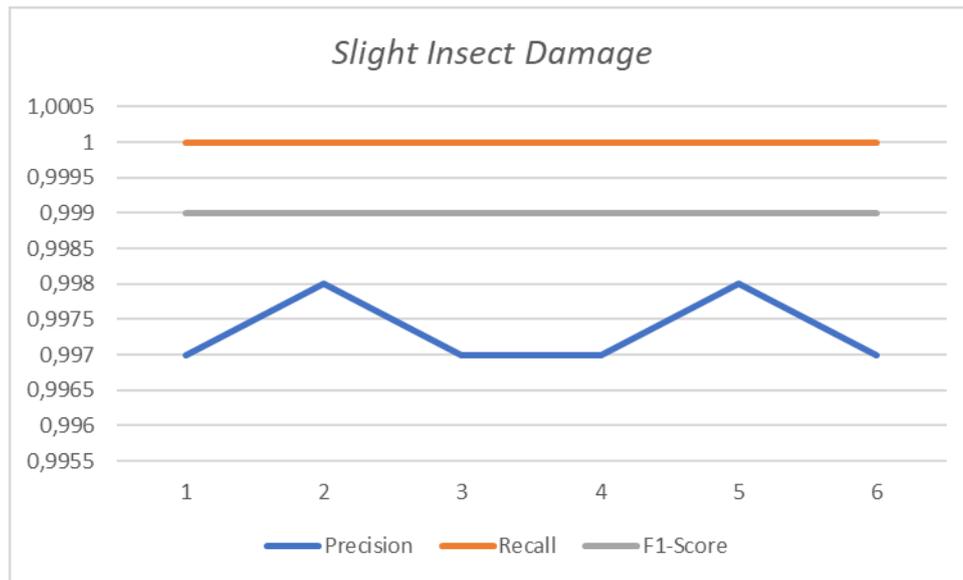
#### 4.4.16 Analisis *Object Classification Class 16 Slight Insect Damage*

Pada penelitian selanjutnya yang mengacu pada parameter dari *object classification class 16 Slight Insect Damage*, pada parameter ini hampir sama dengan analisa sebelumnya yakni membahas parameter hasil dari *confussion matrix* didapatkan dari hasil *training* pada *web tools teachable machine* Berdasarkan Tabel 4.16 dibawah ini.

<i>Epoch</i>	<i>Accuracy (%)</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
50	99.77	0,997	1,000	0,999
100	99.86	0,998	1,000	0,999
150	99.77	0,997	1,000	0,999
200	99.78	0,997	1,000	0,999
250	99.82	0,998	1,000	0,999
300	99.79	0,997	1,000	0,999
$\bar{X}$	<b>99,91</b>	<b>0,997</b>	<b>1,000</b>	<b>0,999</b>

Tabel 4.16 *Object Classification Class 16 Slight Insect Damage*

Berdasarkan Tabel 4.16 merupakan hasil dari pengukuran dari parameter *object classification* pada *class 16* untuk mengetahui *precision*, *recall* dan *f1-score*. Berdasarkan Tabel tersebut dapat diketahui bahwa nilai rata-rata dari akurasi yang didapatkan pada model bagian sample busuk pucuk sebesar 99,91%, sedangkan evaluasi untuk *f1-score* nilai rata-rata nya sebesar 0,999, untuk nilai rata-rata *precision* dan *recall* ialah sebesar 1,000 dan 0,997. Berdasarkan data Berdasarkan Tabel 4.16 dapat ditentukan grafik dari perbandingan pada Gambar 4.19.



Gambar 4.19 Performa Model pada *Object Classification Class 16 Slight Insect Damage*

Gambar 4.19 merupakan grafik performa dengan membandingkan *f1-score* dan juga *accuracy*. Berdasarkan gambar di atas ialah hasil dari performa model, dapat disimpulkan bahwa besar nilai *f1-score* akan sama dengan *accuracy*.

Berdasarkan keseluruhan model yang didapatkan pada objek ini memiliki rata-rata *accuracy* sebesar 0,9991, nilai rata-rata *precision* sebesar 0,997, nilai rata-rata *recall* sebesar 1,000, dan nilai rata-rata *f1-score* sebesar 0,999. Berdasarkan nilai rata-rata yang didapatkan pada penelitian ini ialah dapat disimpulkan bahwa model yang diperoleh memiliki hasil *learning* dan *prediction* yang hampir sempurna untuk pendeteksian.

## 4.5 Analisis Data Gambar dengan *K-Nearest Neighbor*

Teknik pemrosesan gambar dan model *K-Nearest Neighbor* (KNN) untuk menganalisis dan memvisualisasikan kesamaan piksel antara gambar utuh dengan potongan gambar (crop). Teknik ini relevan untuk berbagai aplikasi seperti deteksi objek, segmentasi gambar, dan analisis tekstur. Fokus utama dari penelitian ini adalah untuk menghasilkan peta skor yang menunjukkan tingkat kesamaan antara piksel pada gambar utuh dengan piksel dari potongan gambar, yang divisualisasikan dalam beberapa format untuk memberikan pemahaman yang lebih baik tentang distribusi kesamaan tersebut.

### 4.5.1 Pengumpulan dan Pra-Pemrosesan Data Gambar

Langkah awal dalam penelitian ini melibatkan pemuatan dan pra-pemrosesan gambar. Gambar utuh dan potongan gambar diambil dari direktori yang telah ditentukan. Kedua gambar tersebut diubah dari format BGR ke format RGB untuk kompatibilitas dengan pustaka visualisasi yang digunakan. Selanjutnya, nilai piksel dari kedua gambar dinormalisasi ke rentang  $[0, 1]$ . Proses normalisasi ini penting untuk memastikan bahwa nilai piksel berada dalam rentang yang konsisten untuk analisis lebih lanjut.

### 4.5.2 Pembangunan Model *K-Nearest Neighbor*

Untuk menganalisis kesamaan piksel, model *K-Nearest Neighbor* (KNN) dibuat dan dilatih menggunakan piksel dari potongan gambar. Dalam penelitian ini, kami menggunakan 3 tetangga terdekat (neighbors) untuk memprediksi kesamaan piksel. Setiap piksel dalam gambar utuh kemudian dievaluasi terhadap model ini untuk menentukan jarak rata-rata ke tetangga terdekat dalam ruang warna RGB. Metode KNN dipilih karena kemampuannya dalam mengukur kesamaan berdasarkan jarak dalam ruang fitur yang sederhana namun efektif. Untuk mendukung analisis statistik dari piksel,

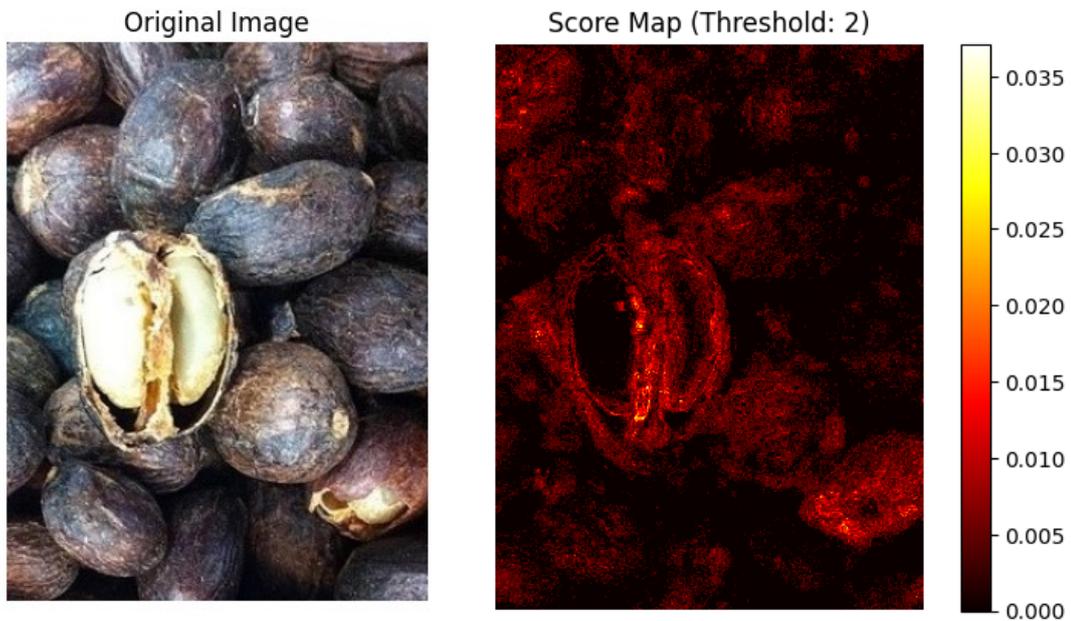
distribusi peluang diskret diperkirakan. Proses ini melibatkan pembentukan histogram dari nilai piksel dengan jumlah bin yang telah ditentukan, dan normalisasi distribusi dengan penambahan nilai prior untuk menghindari probabilitas nol. Estimasi distribusi peluang ini membantu dalam memahami distribusi kesamaan piksel dan memberikan dasar statistik untuk analisis lebih lanjut.

### **4.5.3 Visualisasi Hasil Analisis**

Setelah peta skor dihitung berdasarkan jarak rata-rata dari tetangga terdekat untuk setiap piksel dalam gambar utuh, peta ini kemudian divisualisasikan dengan beberapa cara:

1. Menampilkan gambar asli untuk referensi visual.
2. Menampilkan peta skor dengan skala warna untuk menunjukkan variasi kesamaan.
3. Menampilkan peta skor biner berdasarkan ambang batas tertentu untuk mengidentifikasi area dengan kesamaan tinggi.
4. Menampilkan gambar asli yang dimasker dengan peta skor biner untuk menyoroti area yang relevan.

Visualisasi ini memberikan gambaran yang jelas tentang distribusi kesamaan piksel dan membantu dalam mengidentifikasi area dalam gambar utuh yang memiliki kesamaan tinggi dengan potongan gambar.



Gambar 4.20 Pemrosesan data gambar menggunakan RGB

Gambar yang ditampilkan adalah peta skor dari sebuah gambar berdasarkan kesamaan piksel dengan potongan gambar (crop1) menggunakan model *K-Nearest Neighbors* (KNN). Peta skor ini menunjukkan intensitas kesamaan piksel dalam gambar asli dengan warna merah dan kuning. Area dengan warna lebih cerah menunjukkan kesamaan yang lebih tinggi. Untuk melihat perhitungan matematis yang digunakan dalam menghasilkan peta skor dengan model *K-Nearest Neighbors* (KNN) adalah sebagai berikut.

Piksel dalam gambar asli :  $P_{Original} = (0.6, 0.4, 0.2)$

Tetangga terdekat didalam potongan gambar adalah

$$P_{neighbor1} = (0.65, 0.35, 0.25)$$

$$P_{neighbor2} = (0.55, 0.45, 0.15)$$

$$P_{neighbor3} = (0.60, 0.40, 0.20)$$

Langkah-langkah perhitungannya sebagai berikut:

1. Jarak Euclidean :

$$d_1 = \sqrt{(0.6 - 0.65)^2 + (0.4 - 0.35)^2 + (0.2 - 0.25)^2}$$

$$d_1 = \sqrt{(-0.05)^2 + (0.05)^2 + (-0.05)^2}$$

$$d_1 = \sqrt{(0.025 + 0.025 + 0.025)}$$

$$d_1 = \sqrt{(0.0075)}$$

$$d_1 = 0.0866$$

$$d_2 = \sqrt{(0.6 - 0.55)^2 + (0.4 - 0.45)^2 + (0.2 - 0.15)^2}$$

$$d_2 = \sqrt{(-0.05)^2 + (0.05)^2 + (-0.05)^2}$$

$$d_2 = \sqrt{(0.025 + 0.025 + 0.025)}$$

$$d_2 = \sqrt{(0.0075)}$$

$$d_2 = 0.0866$$

$$d_3 = \sqrt{(0.6 - 0.60)^2 + (0.4 - 0.40)^2 + (0.2 - 0.20)^2}$$

$$d_3 = \sqrt{(-0.00)^2 + (0.00)^2 + (-0.00)^2}$$

$$d_3 = 0.0$$

## 2. Menghitung Rata-Rata Jarak

$$\text{Average Distance} = \frac{0.0866 + 0.0866 + 0.0}{3} = \frac{0.1732}{3} = 0.0577$$

## 3. Menghitung Peta Skor

Untuk piksel pada koordinat (i,j), nilai pada peta skor adalah:

$$\text{Score Map}(i,j) = 0.0577$$

Dengan nilai 0.0577 pada peta skor untuk piksel pada koordinat (i, j), kesimpulannya adalah bahwa piksel tersebut memiliki kontribusi atau tingkat signifikansi sebesar 0.0577 dalam konteks analisis yang sedang dilakukan. Hal ini mungkin menunjukkan bahwa piksel tersebut memiliki karakteristik atau pola yang relevan atau penting dalam analisis tersebut.

#### 4.6 Menghitung *Accuracy* dengan *K-Nearest Neighbor*

Dalam eksperimen ini, penelitian mengadopsi algoritma *K-Nearest Neighbor* (KNN) untuk melakukan klasifikasi terhadap dataset *MNIST* yang terdiri dari gambar-gambar digit tulisan tangan. Dataset ini dibagi menjadi tiga bagian penting: data pelatihan, data validasi, dan data uji. Langkah awal dalam pengolahan data adalah melakukan normalisasi, di mana setiap nilai piksel dibagi dengan nilai maksimumnya. Hal ini dilakukan untuk mempermudah proses komputasi selanjutnya.

Model KNN dilatih menggunakan data pelatihan beserta label yang sesuai. Tujuan utama pelatihan adalah untuk memungkinkan model untuk menangkap pola-pola yang mungkin terdapat dalam dataset, sehingga nantinya dapat mengklasifikasikan data uji dengan tepat. Setelah pelatihan selesai, model tersebut diuji menggunakan data uji, dan performanya diukur menggunakan metrik akurasi. Akurasi ini mencerminkan persentase jumlah sampel yang berhasil diklasifikasikan dengan benar oleh model.

Dalam eksperimen ini, nilai  $k$  untuk KNN ditetapkan sebagai 3. Ini berarti dalam proses klasifikasi, model akan mempertimbangkan tiga tetangga terdekat untuk setiap sampel yang akan diprediksi. Selain itu, dilakukan pengukuran terhadap waktu yang diperlukan baik untuk melatih model maupun melakukan prediksi. Hasil pengukuran menunjukkan bahwa waktu yang diperlukan untuk melatih model adalah 0.06 detik, sementara waktu untuk melakukan prediksi pada data uji adalah 37.71 detik.

Hasil evaluasi ini memberikan gambaran yang komprehensif tentang kinerja model KNN dalam konteks klasifikasi dataset *MNIST*. Meskipun model memiliki akurasi yang tinggi, namun waktu komputasi yang signifikan untuk prediksi dapat menjadi pertimbangan penting, terutama dalam situasi yang memerlukan respons cepat atau pemrosesan data dalam skala besar.

```

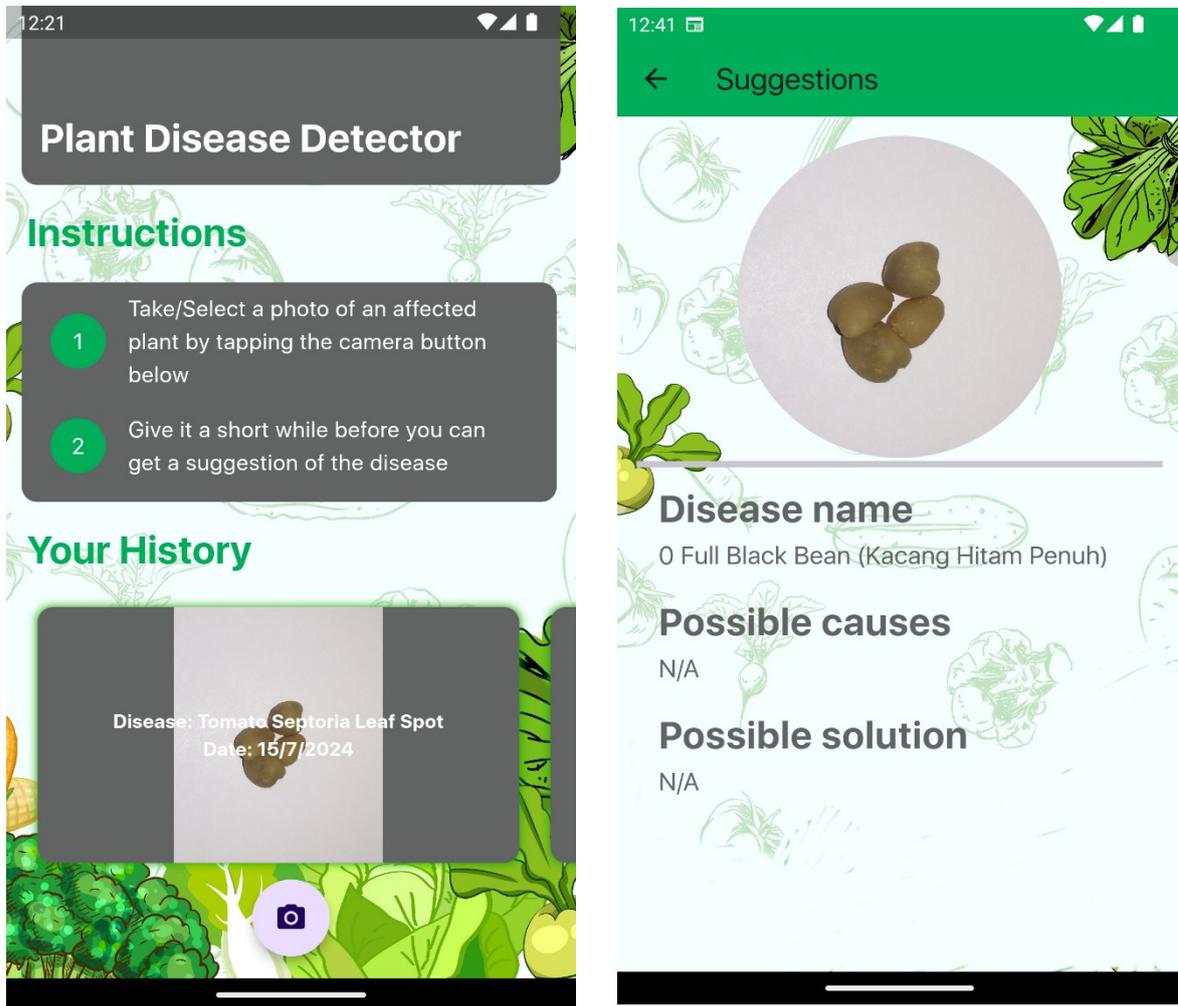
1 import numpy as np
2 from keras.datasets import mnist
3 from sklearn.neighbors import KNeighborsClassifier
4 from sklearn.metrics import accuracy_score
5 import time
6
7 # Function to load MNIST dataset and preprocess it
8 def load_mnist():
9     (x_train, y_train), (x_test, y_test) = mnist.load_data()
10    x_train = np.reshape(x_train, (len(x_train), 28*28))
11    x_test = np.reshape(x_test, (len(x_test), 28*28))
12    maxval = x_train.max()
13    x_train = x_train / maxval
14    x_test = x_test / maxval
15    x_val = x_train[:10000]
16    y_val = y_train[:10000]
17    x_train = x_train[10000:]
18    y_train = y_train[10000:]
19    train_indices = {
20        'xs': np.arange(50),
21        's': np.arange(500),
22        'm': np.arange(5000),
23        'all': np.arange(50000)
24    }
25    return (x_train, y_train), (x_val, y_val), (x_test, y_test), train_indices
26
27 # Load data
28 (x_train, y_train), (x_val, y_val), (x_test, y_test), train_indices = load_mnist()
29
30 # 1. Klasifikasi dengan KNN
31 k = 3 # You can choose any value of k
32 knn = KNeighborsClassifier(n_neighbors=k)
33
34 # Train KNN model
35 start_time = time.time()
36 knn.fit(x_train, y_train)
37 train_time = time.time() - start_time
38 print(f"Time to train KNN model: {train_time:.2f} seconds")
39
40 # Predict on test data
41 start_time = time.time()
42 y_pred_knn = knn.predict(x_test)
43 predict_time = time.time() - start_time
44 print(f"Time to predict using KNN model: {predict_time:.2f} seconds")
45
46 # Calculate accuracy
47 accuracy_knn = accuracy_score(y_test, y_pred_knn) * 100
48 print(f"KNN Test Accuracy: {accuracy_knn:.2f}%")
49

```

Gambar 4.21 *Coding-an* untuk menentukan akurasi

#### 4.7 Menampilkan Tampilan Aplikasi

Aplikasi yang ditampilkan adalah aplikasi berbasis mobile yang dirancang menggunakan *flutter* sebagai *framework* serta menggunakan bahasa pemrograman *dart* sebagai pembuatan aplikasi yang dirancang.



Gambar 4.22 Tampilan Aplikasi Pendeteksi Penyakit Kopi

Aplikasi pendeteksi penyakit tanaman pada halaman beranda menampilkan judul "Plant Disease Detector" di bagian atas, dengan instruksi bagi pengguna untuk mengambil atau memilih foto tanaman yang terinfeksi menggunakan tombol kamera di

bawah. Setelah itu, pengguna diminta menunggu sebentar hingga aplikasi memberikan saran mengenai penyakit yang terdeteksi. Di bagian bawah terdapat riwayat foto tanaman yang telah dianalisis sebelumnya beserta diagnosis penyakitnya.

Halaman kedua aplikasi menampilkan judul "*Suggestions*" di bagian atas, dengan gambar close-up dari tanaman yang terinfeksi. Bagian ini memberikan rincian mengenai nama penyakit yang terdeteksi, seperti "*Full Black Bean* (Kacang Hitam Penuh)". Aplikasi ini secara keseluruhan berfungsi untuk mendeteksi penyakit pada tanaman dengan menyediakan diagnosis berdasarkan foto yang diambil atau dipilih oleh pengguna serta menampilkan riwayat diagnosis sebelumnya.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Kesimpulan yang didapatkan pada penelitian ini berdasarkan analisis dan pembahasan antara lain adalah:

1. Penelitian ini telah berhasil menerapkan metode *K-Nearest Neighbor* (KNN) untuk klasifikasi *defect* pada biji kopi Arabika Gayo. Dari hasil pengujian yang dilakukan, metode KNN menunjukkan performa yang optimal dengan akurasi pelatihan dan pengujian yang konsisten. Hal ini menunjukkan bahwa model KNN yang digunakan mampu mengklasifikasikan *defect* pada biji kopi dengan sangat baik tanpa adanya tanda-tanda *overfitting* atau *underfitting*.
2. Kelebihan dari metode KNN yang diterapkan dalam penelitian ini termasuk kemampuannya untuk memberikan hasil klasifikasi yang akurat dan efisien. Namun, terdapat keterbatasan terkait dengan waktu komputasi yang diperlukan untuk proses prediksi, terutama dalam skala besar atau situasi yang memerlukan respons cepat. Hal ini menjadi pertimbangan penting dalam penerapan metode ini di lapangan.
3. Implementasi sistem klasifikasi *defect* pada biji kopi Arabika ini memberikan manfaat signifikan bagi petani. Dengan adanya sistem ini, petani dapat lebih mudah dan cepat dalam menentukan kualitas biji kopi mereka, yang pada akhirnya dapat meningkatkan efisiensi kerja dan kualitas produk kopi yang dihasilkan.

## 5.2 Saran

Dari hasil penelitian didapatkan saran untuk kemajuan para petani yang ada di Koperasi Usaha Tani GAYO Kecamatan Bebeseb Aceh Tengah, antara lain:

1. Melakukan pelatihan dan edukasi bagi petani mengenai penggunaan sistem klasifikasi ini sangat penting. Dengan pemahaman yang baik tentang cara kerja dan manfaat sistem ini, petani akan lebih termotivasi untuk menggunakannya secara optimal dalam kegiatan sehari-hari mereka.
2. Pengembangan aplikasi mobile berbasis *Flutter* yang terintegrasi dengan sistem klasifikasi ini dapat membantu petani untuk melakukan identifikasi *defect* biji kopi secara langsung di lapangan. Aplikasi ini harus dirancang untuk mudah digunakan dan memiliki fitur yang lengkap untuk memenuhi kebutuhan petani.
3. Selain KNN, metode lain seperti *Convolutional Neural Networks* (CNN) atau *Support Vector Machine* (SVM) dapat dicoba untuk membandingkan hasil dan mungkin mendapatkan akurasi yang lebih tinggi.

## DAFTAR PUSTAKA

- Ariska, D. M., Astuti, I. F., & Agus, F. (2024). Rancangan dan Implementasi Sistem Pendukung Keputusan Pemilihan Bibit Tanaman Karet. *IKRA-ITH Informatika : Jurnal Komputer Dan Informatika*, 8(2), 82–91. <https://doi.org/10.37817/ikraith-informatika.v8i2.2961>
- Chazar, C., & Rafsanjani, M. H. (2022). Penerapan Teachable Machine Pada Klasifikasi Machine Learning Untuk Identifikasi Bibit Tanaman. *Prosiding Seminar Nasional Inovasi Dan Adopsi Teknologi (INOTEK)*, 2(1), 32–40. <https://doi.org/10.35969/inotek.v2i1.207>
- Fichri, M., Ramadhan, A., Arsyad, F., & Jumaryadi, Y. (2022). Implementasi Simple Additive Weighting Dalam Menentukan Biji Kopi Terbaik. *Journal of Information System Research (JOSH)*, 3(3), 234–241. <https://doi.org/10.47065/josh.v3i3.1505>
- Juliaviani, N., Sahara, S., & Winandi, R. (2017). Transmission Price of Gayo Arabica Coffee In Aceh Province. *Jurnal Agribisnis Indonesia*, 5(1), 39.
- Mahendra, A. A., Suranti, D., & Fredricka, J. (2023). Sistem Pendukung Keputusan Calon Penerima BPJS-PBI Pada Dinas Sosial Kota Bengkulu Menggunakan Metode K-Nearest Neighbor (KNN). *Jurnal Media Infotama*, 19(2), 391–400. <https://doi.org/10.37676/jmi.v19i2.4290>
- Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, S., & Zaqiah, Q. Y. (2022). Deep Learning dan Penerapannya dalam Pembelajaran. *JIIP - Jurnal Ilmiah Ilmu Pendidikan*, 5(9), 3258–3267. <https://doi.org/10.54371/jiip.v5i9.805>
- Tolana, F., Muzakkir, I., & Riadi, A. (2023). Metode K-Nearest Neighbor (KNN) Untuk Sistem Pakar Diagnosa Penyakit Pneumonia Pada Balita. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, 6(3), 363–373.
- Varian, M. A. (2017). *Analisa karakteristik kopi arabika dan metode penjemuran terhadap citarasa kopi arabika menggunakan analisa varian*. 14–36.

- Yuhandri, Y., Ramadhanu, A., & Syahputra, H. (2022). Pengenalan Teknologi Pengolahan Citra Digital (Digital Image Processing) Untuk Santri Di Rahmatan Lil'Alamin International Islamic Boarding School. *Community Development Journal: Jurnal Pengabdian Masyarakat*, 3(2), 1239–1244. <https://doi.org/10.31004/cdj.v3i2.5868>
- Biessek, M. (2019). *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*. Packt Publishing.
- Giordano, A. (2019). *Flutter Projects: A practical, project-based guide to building real-world cross-platform mobile applications and games*. Packt Publishing.
- Kurosaki, K. (2020). *Mastering Flutter: A step-by-step guide to designing and building Flutter apps*. Independently published.
- Napoli, C. (2019). *Beginning Flutter: A hands on guide to app development*. Apress.
- Sari, A. (2023). *Perbandingan Metode K-Nearest Neighbour (KNN) dan Support Vector Machine (SVM) dalam Identifikasi Pola Motif Batik Menggunakan Ekstraksi Fitur Radially Averaged Power Spectrum Value (RAPSV)*. Skripsi. Universitas Muhammadiyah Gresik
- Rezky Firmansyah. 2021. IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI BUNGA. Skripsi. UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA

## LAMPIRAN

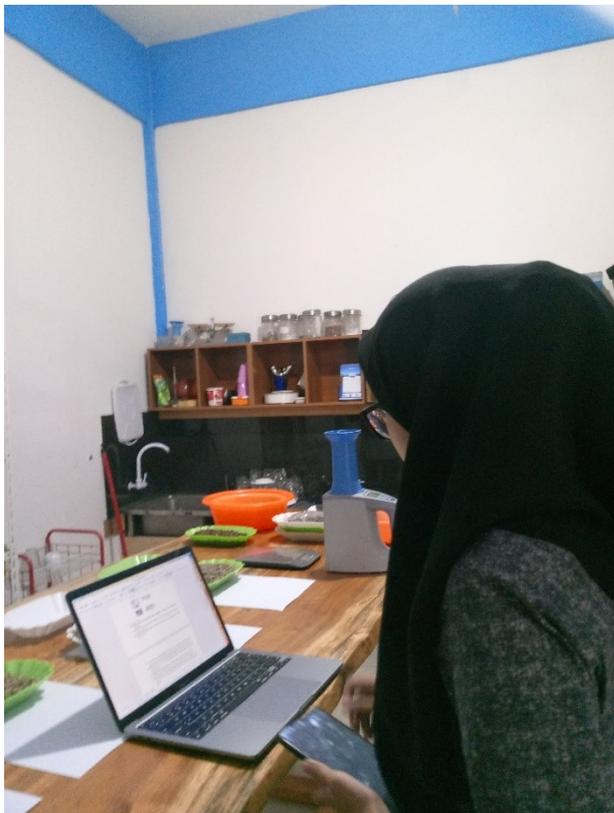
### 1. Kodingan RGB KNN

```

1 import numpy as np
2 import cv2
3 import os # Add this line
4 from sklearn.neighbors import KNeighborsClassifier
5 from matplotlib import pyplot as plt
6 from google.colab.patches import cv2_imshow
7
8 # Function to estimate discrete pdf
9 def estimate_discrete_pdf(x, K, prior=1e-5):
10     p = np.ones(K) * prior
11     for value in x:
12         p[int(value)] += 1
13     p /= np.sum(p)
14     return p
15
16 # Function to display score maps
17 def display_score_maps(im, score_map, thresh):
18     plt.imshow(im)
19     plt.title('Original Image')
20     plt.axis('off')
21     plt.show()
22
23     plt.imshow(score_map, cmap='hot', interpolation='nearest')
24     plt.title(f'Score Map (Threshold: {thresh})')
25     plt.colorbar()
26     plt.axis('off')
27     plt.show()
28
29     plt.imshow(score_map > thresh, cmap='gray')
30     plt.title(f'Thresholded Score Map (Threshold: {thresh})')
31     plt.axis('off')
32     plt.show()
33
34     masked_im = np.tile(np.reshape(score_map > thresh, (im.shape[0], im.shape[1], 1)), (1, 1, 3)) * im
35     plt.imshow(masked_im)
36     plt.title('Masked Image')
37     plt.axis('off')
38     plt.show()
39
40 # Mount Google Drive
41 from google.colab import drive
42 drive.mount('/content/drive')
43
44 # Check if the files exist in the directory
45 datadir = "/content/drive/My Drive/Colab Notebooks/kopi123/"
46 print("Files in directory:", os.listdir(datadir))
47
48 # Load images
49 im_path = datadir + '3_Dried_Cherry_Pod.jpg'
50 crop_path = datadir + '3_Dried_Cherry_Pod_crop.jpg'
51
52 # Ensure image paths are correct and images are loaded
53 im = cv2.imread(im_path)
54 crop = cv2.imread(crop_path)
55
56 if im is None:
57     print(f"Error: Unable to load image at {im_path}")
58 else:
59     print(f"Successfully loaded image at {im_path}")
60
61 if crop is None:
62     print(f"Error: Unable to load image at {crop_path}")
63 else:
64     print(f"Successfully loaded image at {crop_path}")
65
66 # Convert images to RGB and normalize if loaded correctly
67 if im is not None and crop is not None:
68     im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB) / 255
69     crop = cv2.cvtColor(crop, cv2.COLOR_BGR2RGB) / 255
70
71 # Set the number of bins K
72 K = 150
73
74 # Reshape pixels to (h*w, 3)
75 full_pixels = im.reshape(-1, 3)
76 crop_pixels = crop.reshape(-1, 3)
77
78 # Create KNN model
79 knn = KNeighborsClassifier(n_neighbors=3)
80
81 # Train KNN on crop pixels
82 knn.fit(crop_pixels, np.zeros(len(crop_pixels)))
83
84 # Predict on full image pixels
85 distances, indices = knn.kneighbors(full_pixels)
86 predictions = knn.predict(full_pixels)
87
88 # Estimate score for each pixel in full image according to nearest neighbors
89 score_map = np.zeros(im.shape[:2])
90 for i in range(im.shape[0]):
91     for j in range(im.shape[1]):
92         idx = i * im.shape[1] + j
93         score_map[i, j] = np.mean(distances[idx])
94
95 # Display score maps
96 thresh = 2 # Example threshold, can be adjusted
97 display_score_maps(im, score_map, thresh)
98 else:
99     print("Images could not be loaded, check the file paths and ensure the images exist.")
100

```

## 2. Pengambilan Dataset Kopi







MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH  
**UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**  
**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019  
 Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003  
<https://fki.umsumed.ac.id> fki@umsumed.ac.id @umsumedan umsumedan umsumedan umsumedan

Nomor : 322/IL.3-AU/UMSU-09/F/2024 Medan, 18 Dzulhijjah 1445 H  
 Lampiran : - 25 Juni 2024 M  
 Perihal : IZIN RISET PENDAHULUAN

Kepada Yth.  
 Bapak/Ibu Pimpinan  
 Koperasi Usaha Tani Gayo  
 Jl.Keramat Mulakat No.202,Desa Bebesen,Kec Bebesan Aceh Tengah

Di Tempat

*Assalamu 'alaikum Warahmatullahi Wabarakatuh*

Dengan hormat, sehubungan mahasiswa kami akan menyelesaikan studi, untuk itu kami memohon kesediaan Bapak / Ibu untuk memberikan kesempatan pada mahasiswa kami melakukan riset di Perusahaan / Instansi yang Bapak / Ibu pimpin, guna untuk penyusunan skripsi yang merupakan salah satu persyaratan dalam menyelesaikan Program Studi Strata Satu (S-1)

Adapun Mahasiswa/i di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara tersebut adalah:

Nama : Khairunnisa  
 Npm : 2009010042  
 Jurusan : Sistem informasi  
 Semester : VIII (Delapan)  
 Judul : Mengklasifikasi Defect Pada Kopi Arabika Berdasarkan Grade Menggunakan Metode KNN ( K-NEAREST NEIGHBOR) Di Koperasi Usaha Tani Gayo Kec.Bebeseb Aceh Tengah  
 Email : knisa4806@gmail.com  
 Hp/Wa : 082390581455

Demikianlah surat kami ini, atas perhatian dan kerjasama yang Bapak / Ibu berikan kami ucapkan terimakasih

*Wassalamu 'alaikum Warahmatullahi Wabarakatuh*



Cc.File

a.n.Dekan  
 Wakil Dekan I  
  
 H. H. Maulana, S.T., M.Kom  
 NIDN : 0121119102





**KOPERASI USAHA TANI GAYO**  
Kabupaten Aceh Tengah-Indonesia

Nomor. PAD : AHU-0002130.AH.01.27.TAHUN 2021

Office: jln. Keramat Mufakat, No. 202 Desa Bebesen, Kec. Bebesen, Kab. Aceh Tengah  
Telp: +6282360071321, Email : [info@utanigayo.com](mailto:info@utanigayo.com) / [Adm@utanigayo.com](mailto:Adm@utanigayo.com) , pos 24552



Nomor : 087/UTG/SKP/VI/2024

Beberes, 28 Juni 2024

Lampiran :

Kepada YTH :

Perihal : Balasan Izin Riset

Dekan

Universitas Muhammadiyah Sumatera utara  
Fakultas Ilmu Komputer Dan Teknologi  
Informasi

**Tempat**

Dengan hormat,

Yang bertanda tangan dibawah ini :

Nama : Rafiyandi

Jabatan : Ketua Koperasi

Menerangkan Bahwa :

Nama : Khairunnisa

NPM : 2009010042

Program Studi : Sistem Informasi

Semester : VII (Delapan )

Alamat : Jln.puskesmas I Gg. Ridho No.4 Medan Sunggal

Kami telah setuju nama tersebut melakukan kegiatan riset pada koperasi usaha tani gayo Kecamatan bebesen kabupaten aceh Tengah sebagai syarat penyusunan skripsi dengan judul:

**"KLASIFIKASI DEFECT PADA BIJI KOPI ARABIKA MENGGUNAKAN METODE KNN (K-NEAREST NAIGHBOR ) DI KOPERASI USAHA TANI GAYO KEC. BEBESEN ACEH TENGAH".**

Demikian surat ini kami sampaikan atas perhatian serta kerja samanya kami ucapkan terima kasih.

Hormat Kami,

Ketua Koperasi

  
**UTANIGAYO**  
RAFIYANDI



### LETTER OF ACCEPTANCE (LoA)

Kepada Yth Bpk/Ibu/Sdr

***Khairunnisa, Martiano***

Di

Tempat

Dengan ini kami sampaikan bahwa naskah dengan rincian berikut dinyatakan diterima untuk diterbitkan di IJATCoS (Indonesian Journal Of Applied Technology, Computer, and Science, ISSN: xxx terbitan Volume 1 Nomor 3 Edisi Februari 2025).

Judul	Classification of Defects in Arabica Coffee Beans Using the KNN (K-Nearest Neighbor) Method in the Gayo Farming Cooperative, Bebeseb District, Central Aceh
Penulis	Khairunnisa, Martiano
Correspondent Email	knisa4806@gmail.com

Demikianlah surat keterangan ini kami buat untuk dapat digunakan seperlunya.

Medan, 22 September 2024

Editor in Chief



**Indah Purnama Sari, S.T., M.Kom.**

