

# **KLASIFIKASI JENIS KOPRA BERDASARKAN EKSTRAKSI FITUR MENGGUNAKAN METODE K- NEAREST NEIGHBOUR**

Skripsi



Oleh

**PUSPITASARI**

NIM : 11190910000003

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH**

**JAKARTA**

**2023 M / 1444 H**

## DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang .....	4
1.2 Rumusan Masalah .....	7
1.3 Batasan Masalah.....	7
1.4 Tujuan Penelitian.....	8
1.5 Manfaat Penelitian.....	8
1.5.1 Manfaat Bagi Peneliti.....	8
1.5.2 Manfaat Bagi Universitas.....	8
1.6 Metodologi Penelitian .....	9
1.6.1 Metode Pengumpulan Data .....	9
1.6.2 Metode Perancangan Model.....	9
1.7 Sistematika Penelitian .....	10
BAB II.....	12
LANDASAN TEORI.....	12
2.1 Kelapa.....	12
2.2 Jenis-jenis Kelapa.....	12
2.3 Kopra .....	13
2.4 Pengolahan Citra .....	14
2.5 Citra Digital .....	15
2.4.1 Definisi Citra Digital.....	15
2.4.2 Pembentukan Citra Digital .....	16
2.4.3 Jenis-jenis Citra Digital.....	17
3.5 Konversi Citra .....	19
2.5.1 Warna RGB .....	19
2.5.2 Warna HSV .....	19
2.5.3 Warna 23	
2.5.4 Moment warna .....	20
3.6 Artificial Intelligence .....	21
3.7 K-Nearest Neighbors (K-NN) .....	26

3.8 Mengukur Performansi .....	29
3.9 Studi Pustaka .....	31
BAB III .....	38
METODOLOGI PENELITIAN .....	38
3.1 Metode Pengumpulan Data .....	38
3.1.1 Studi Pustaka.....	38
3.1.2 Jenis dan Sumber Data.....	38
3.2 Metode Perancangan Model .....	38
3.2.1 46	
3.2.2 Ekstraksi Ciri Warna .....	39
3.2.3 Split Dependent dan Independent value.....	43
3.2.4 Feature Scaling.....	44
3.2.5 Split Data.....	45
3.2.6 Penerapan Metode K-Nearest Neighbor .....	45
3.2.7 Pengujian Akurasi .....	48
3.3 Alur Penelitian.....	51
DAFTAR PUSTAKA .....	53

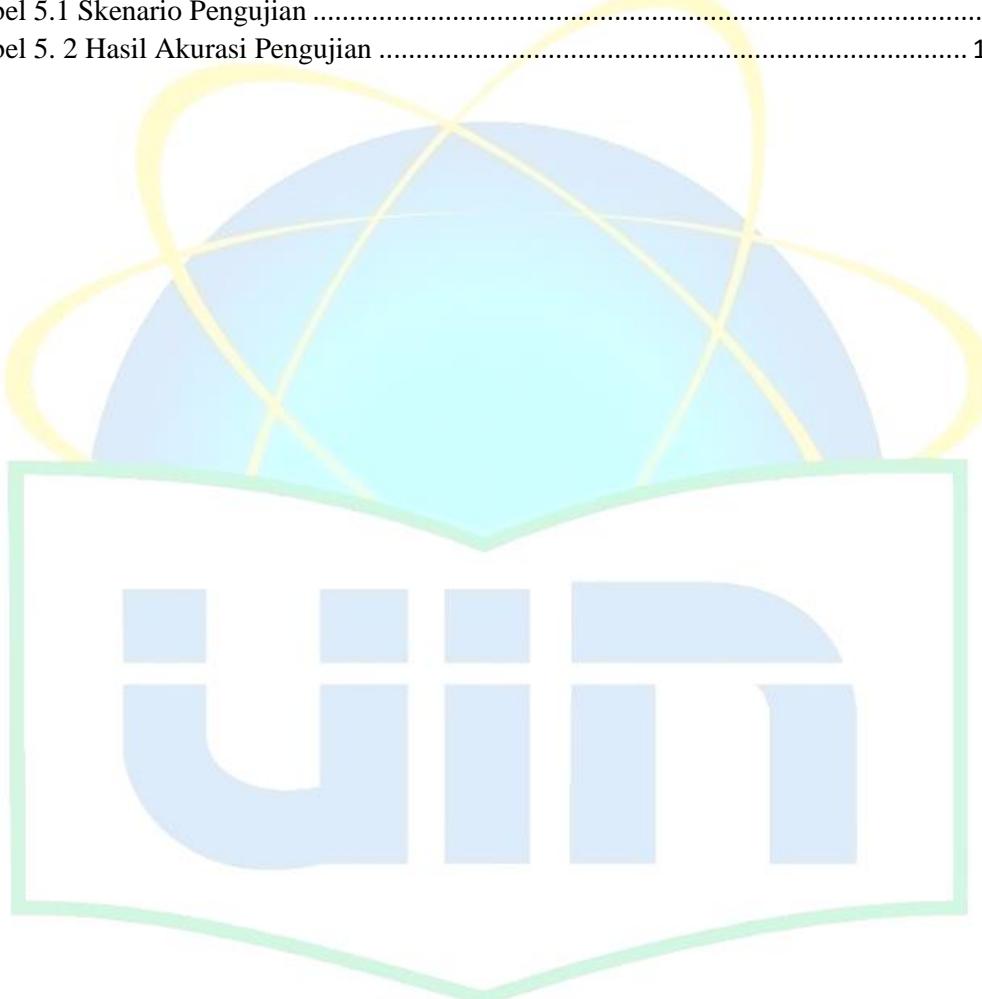


## DAFTAR GAMBAR

Gambar 2. 1 Pengolahan Citra .....	17
Gambar 2. 2 Koordinat Citra Digital .....	19
Gambar 2. 3 Perbedaan Pemograman Tradisional dan Machine Learning.....	29
Gambar 2. 4 Supervised Learning.....	30
Gambar 2. 5 Unsupervised Learning .....	31
Gambar 2. 6 Reinforcement Learning.....	32
Gambar 2. 7 Ilustrasi Algoritma K-NN .....	33
Gambar 3. 1 Alur Penelitian .....	50
Gambar 4. 1 Pra-Proses Simulasi Sistem.....	55
Gambar 4. 2 Contoh Sampel Kopra Edible.....	62
Gambar 4. 3 Contoh Sampel Kopra Reguler .....	62
Gambar 4. 4 Contoh Sampel Kopra Reject.....	63
Gambar 4. 5 Citra RGB $3 \times 3$ .....	64
Gambar 4. 6 Citra HSV $3 \times 3$ .....	65
Gambar 4. 7 Citra Grayscale $3 \times 3$ .....	66
Gambar 5. 1 Confusion Matrix skenario 1 .....	96
Gambar 5. 2 <i>Classification Report</i> skenario 1 .....	97
Gambar 5. 3 Confusion Matrix skenario 2.....	99
Gambar 5. 4 <i>Classification Report</i> skenario 2 .....	100
Gambar 5. 5 Confusion Matrix skenario 3.....	102
Gambar 5. 6 <i>Classification Report</i> skenario 3 .....	103
Gambar 5. 7 Confusion Matrix skenario 4.....	106
Gambar 5. 8 <i>Classification Report</i> skenario 4 .....	107
Gambar 5. 9 Confusion Matrix skenario 5.....	109
Gambar 5. 10 <i>Classification Report</i> skenario 5 .....	110
Gambar 5. 11 Confusion Matrix skenario 6.....	112
Gambar 5. 12 <i>Classification Report</i> skenario 6 .....	113
Gambar 5. 13 Confusion Matrix skenario 7 .....	115
Gambar 5. 14 <i>Classification Report</i> skenario 7 .....	116
Gambar 5. 15 Confusion Matrix skenario 8.....	119
Gambar 5. 16 <i>Classification Report</i> skenario 8 .....	120
Gambar 5. 17 Confusion Matrix skenario 9.....	122
Gambar 5. 18 <i>Classification Report</i> skenario 9 .....	123
Gambar 5. 19 Confusion Matrix skenario 10.....	125
Gambar 5. 20 <i>Classification Report</i> skenario 10 .....	126
Gambar 5. 21 Confusion Matrix skenario 11.....	129
Gambar 5. 22 <i>Classification Report</i> skenario 11 .....	130
Gambar 5. 23 Confusion Matrix skenario 12.....	132
Gambar 5. 24 <i>Classification Report</i> skenario 12 .....	133

## DAFTAR TABEL

Tabel 2. 1 Confusion Matrix .....	36
Tabel 2. 2 Studi Pustaka.....	41
Tabel 4. 1 Spesifikasi Perangkat Keras.....	52
Tabel 4. 2 Spesifikasi Perangkat Lunak.....	52
Tabel 4. 3 Contoh data training citra kopra .....	73
Tabel 4. 4 Contoh data testing citra kopra .....	73
Tabel 4. 5 Hasil pengurutan nilai jarak .....	77
Tabel 5.1 Skenario Pengujian .....	92
Tabel 5. 2 Hasil Akurasi Pengujian .....	135



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Indonesia merupakan negara yang memiliki iklim tropis. Terdapat banyak tanaman yang dapat tumbuh pada negara yang beriklim tropis, salah satunya adalah tanaman kelapa. Tanaman kelapa merupakan salah satu tanaman yang sering kali dibudidayakan oleh masyarakat sekitar karena memiliki nilai ekonomis yang tinggi. Hampir semua bagian tanaman kelapa dapat memberikan manfaat, mulai dari akar, batang, buah, hingga daunnya memiliki manfaat tersendiri (Rahayu Marlis et al., 2021). Buah kelapa adalah bagian dari tanaman kelapa yang jika diolah memberikan manfaat untuk kehidupan manusia. Salah satu bentuk pengolahannya adalah dengan menjadikannya kopra. Kopra merupakan kelapa yang telah dikeluarkan isinya lalu dikeringkan (Adang et al., 2020).

Berdasarkan kualitas dan penggunaanya, kopra terbagi menjadi tiga jenis yaitu kopra edible, reguler, dan reject (Lahay et al., 2023). Kopra edible adalah kopra yang memenuhi standar kebersihan dan keamanan pangan yang ditetapkan oleh badan pengawas makanan. Jenis kopra ini dapat dijadikan sebagai bahan baku dalam produksi minyak kelapa, makanan ringan, produk susu kelapa, dan sebagainya. Kopra reguler adalah kopra yang tidak memenuhi standar kebersihan dan keamanan untuk dikonsumsi langsung, tetapi masih dapat digunakan dalam berbagai industri lainnya. Kopra reguler biasanya digunakan dalam industri kosmetik, sabun, minyak sayur, dan produk-produk non-pangan lainnya. Kopra reject adalah kopra yang tidak memenuhi standar kualitas yang diperlukan untuk digunakan dalam industri makanan, minuman, atau industri lainnya (Zubair et al., 2020).

Pemisahan dan pengklasifikasian jenis kopra berdasarkan kualitas dan penggunaannya sangat penting dilakukan untuk kebutuhan hilirisasi pada

industry kelapa khusunya pada bagian kopra. Kesalahan dalam menentukan ini dapat berakibat adanya undang-undang yang dilanggar, seperti yang terjadi pada industri kelapa sawit (M. Perdagangan dan R. Indonesia, 2022).

Selama ini, dari hasil wawancara yang dilakukan kepada salah satu petani di kabupaten Indragiri Hilir Provinsi Riau, pengklasifikasian jenis kopra umumnya masih dilakukan oleh petani secara manual. Tentunya proses ini membutuhkan tenaga, waktu, dan biaya yang cukup tinggi. Oleh karena itu, dibutuhkan pengembangan metode yang dapat mengklasifikasikan jenis kopra secara otomatis dan akurat untuk mendukung proses ini. Klasifikasi yang akurat dan efisien dapat membantu mengelola pasokan kopra dengan standar kualitas yang berlaku, serta memenuhi kebutuhan beragam pengguna kopra di berbagai sektor industri dengan cara yang lebih tepat (Rachmawati & Gunawan, 2020).

Penelitian terkait telah banyak dilakukan, diantara penelitian yang dilakukan oleh (Adang et al., 2020), penelitian telah membuat metode untuk klasifikasi kematangan kopra menggunakan naïve bayes berdasarkan warna daging kopra dengan cara membandingkan nilai variabel penduga setiap model, hasil yang diperoleh dari model 1-3 rata-rata akurasinya mencapai 91,12% (Adang et al., 2020). Penelitian lainnya yang dilakukan oleh Idam et al (2023), membuat sistem untuk menentukan *grade* kopra berdasarkan kadar air, kadar minyak, dan jamur menggunakan logika fuzy. Sistem ini mampu memberikan kemudahan, menghemat waktu, dan menimilisir kesalahan yang di akibatkan oleh *human error*. Akurasi yang didapat dari sistem ini sebesar 95% (Lahay et al., 2023).

Penelitian ini akan fokus membuat model klasifikasi jenis kopra berdasarkan ekstraksi fitur menggunakan metode *k-nearest neighbour*. Untuk mendukung proses klasifikasi, perlu adanya pengolahan citra digital untuk mempersiapkan datanya. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra yang bertujuan untuk

memperbaiki kualitas citra agar mudah di interpretasi oleh manusia atau mesin komputer (Rilo Pambudi et al., 2020). Pada pengolahan citra tentunya tidak lepas dari proses pengenalan objek untuk memisahkan bagian tertentu dalam suatu citra agar dapat mengenali objek dengan akurat. Ekstraksi fitur pada citra merupakan tahap yang sangat penting dalam pengolahan citra digital (Nurul Huda, 2019).

Ekstraksi fitur adalah proses untuk mengekstrak atau mengambil nilai unik dari setiap citra yang ingin dikenali untuk dapat dibedakan dengan objek lainnya (Priadana, 2018). Pada penelitian yang dilakukan oleh (Rahayu Marlis et al., 2021) ekstraksi fitur dilakukan pada citra kopra putih yang terdiri dari kopra putih yang masuk kelas A (kualitas ekspor) dan kelas B (bukan kualitas ekspor) menggunakan model ekstraksi fitur warna dan bentuk. Ekstraksi fitur ini digunakan untuk mendapatkan nilai dari citra kopra putih dalam bentuk numerik atau angka. Begitu juga dengan (Anushya, 2020) melakukan ekstraksi fitur warna dan tekstur pada batok kelapa untuk mendapatkan ciri dari citra tersebut. Hasil fitur-fitur yang telah diekstraksi dapat digunakan sebagai inputan untuk proses klasifikasi (Sugiarto & Budiman, 2021).

Terdapat banyak metode yang dapat digunakan untuk proses klasifikasi, salah satunya adalah *K-Nearest Neighbor*. *K-Nearest Neighbor* merupakan metode yang dapat digunakan untuk klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Siti Raysyah et al, 2021). Metode ini memiliki prinsip yang sederhana dan mampu bekerja dengan maksimal terhadap data yang mempunyai kesalahan acak (Setiawan & Triayudi, 2022)(Sari, 2020). Selain itu, metode ini juga mudah untuk dimplementasikan, membuat banyak penelitian yang menggunakan *K-Nearest Neighbor* sebagai metode klasifikasi (Nabila et al., 2021). Penelitian sebelumnya telah menggunakan *K-Nearest Neighbor* sebagai metode klasifikasi terhadap dataset citra penyakit *pneumonia* Dimana hasil dari klasifikasi tersebut mendapat akurasi yang tinggi yaitu sekitar 96% (Halim & Anraeni, 2021).

Penelitian lainnya yang dilakukan oleh Sandi Prayoga dkk pada tahun 2020, telah membuat sistem klasifikasi jenis bunga mawar menggunakan metode K-*Nearest Neighbor* (K-NN). Pengujian yang dilakukan menggunakan 250 citra yang dibagi dengan perbandingan 90% data *training* dan 10% data *testing*, sehingga data latih yang digunakan sebanyak 150 citra bunga mawar dan 75 citra bukan bunga mawar, serta 25 citra sisanya digunakan sebagai data uji. Data citra tersebut diuji menggunakan *confusion matrix* dan berhasil mendapatkan hasil akurasi sebesar 92%, presisi sebesar 93%, dan *recall* sebesar 92% (Prayogo et al., 2022).

Dengan merujuk pada penjelasan sebelumnya, penulis berencana untuk melakukan penelitian dengan judul "**Klasifikasi Jenis Kopra Menggunakan Ekstraksi Fitur Warna dengan Metode K-Nearest Neighbor**". Melalui penelitian ini, diharapkan dapat mengembangkan model klasifikasi yang dapat membedakan jenis kopra dengan akurasi yang tinggi berdasarkan ciri-ciri warnanya. Penelitian ini diharapkan memberikan kontribusi dalam pengolahan dan identifikasi jenis kopra serta mendukung pengambilan keputusan yang lebih baik dalam industri kelapa.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan di atas, maka penulis merumuskan masalah yaitu bagaimana mengklasifikasi jenis kopra berdasarkan hasil ekstraksi fitur menggunakan metode *K-Nearest Neighbor*.

## 1.3 Batasan Masalah

Sebagai pembatasan pengembangan penelitian dan sesuai dengan tujuan yang ditetapkan, maka penulis memberikan ruang lingkup batasan sebagai berikut:

### 1.3.1 Proses

1. Dataset yang digunakan adalah citra berwarna.

2. Dataset citra terdiri atas tiga jenis kopra yaitu edible, regular, dan reject.
3. Citra yang digunakan yaitu kopra non telungkup.
4. Format citra yang digunakan adalah PNG (*Portable Network Graphics*).

### 1.3.2 Metode

1. Klasifikasi jenis kopra ini dilakukan berdasarkan hasil 12 percobaan dengan mempertimbangkan kombinasi fitur.
2. Ekstraksi fitur warna menggunakan RGB, HSV, *GrayScale* dengan tiga *color moment* yaitu *mean*, standar deviasi, dan variasi.
3. Ekstraksi fitur bentuk menggunakan area dan perimeter.
4. Ekstraksi fitur tekstur menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*) dengan jarak 1 dan sudut 0 serta beberapa fitur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*.
5. Proses klasifikasi jenis kopra menggunakan metode *K-Nearest Neighbor*.

### 1.3.3 Tools

1. *Tools* yang digunakan untuk mengekstraksi ciri dan mengklasifikasikan adalah *Visual studio code* dengan Bahasa pemrograman python.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengklasifikasikan jenis kopra edible, regular, dan reject dengan menggunakan hasil ekstraksi citra dan metode *k-nearest neighbour* sebagai metode klasifikasinya.

## **1.5 Manfaat Penelitian**

### **1.5.1 Manfaat Bagi Penulis**

1. Untuk memenuhi salah satu persyaratan untuk mendapatkan gelar sarjana computer (S.Kom) pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi.
2. Menambah pengalaman penulis dalam melakukan penelitian dalam bidang *artificial intelligence* khususnya machine learning.
3. Sarana implementasi ilmu-ilmu yang telah didapatkan penulis selama masa perkuliahan,

### **1.5.2 Manfaat Bagi Universitas**

1. Dapat dijadikan referensi literatur penelitian pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Memberikan gambaran sejauh mana kemampuan mahasiswa dalam menguasai ilmu yang telah diberikan selama masa perkuliahan.

### **1.5.3 Manfaat Bagi Pembaca**

1. Menjadi referensi dalam hal pengolahan citra digital dengan ekstraksi fitur pada citra.
2. Memberikan pengetahuan kepada pembaca mengenai metode klasifikasi menggunakan algoritma K-Nearest Neighbor (K-NN).

## **1.6 Metodologi Penelitian**

Pada penyusunan penelitian ini, penulis menggunakan metode-metode sebagai berikut:

### **1.6.1 Metode Pengumpulan Data**

1. Studi Pustaka

Studi pustaka ini dilakukan dengan mengumpulkan data dengan membaca buku referensi, karya ilmiah, dan jurnal yang berhubungan dengan topik yang dibahas.

## 2. Wawancara

Wawancara dilakukan dengan pihak yang bersangkutan terkait dengan objek skripsi yang diteliti oleh penulis.

### 1.6.2 Metode Pengembangan Model

Dalam penelitian ini, penulis menggunakan metode simulasi. Adapun tahap-tahap pada metode simulasi adalah sebagai berikut (Gandomi et al., 2010):

1. *Problem Formulation*
2. *Model Conceptual*
3. *Collection of Input Data*
4. *Modelling*
5. *Simulation*
6. *Verification and Validation*
7. *Design of Experiment*
8. *Execute Simulation and Analysis Output*

## 1.7 Sistematika Penelitian

Pada penyusunan penelitian ini, penulis melakukan sistematika penelitian yang tersusun ke dalam enam bab. Dimana setiap bab-nya memiliki beberapa sub bab yang secara keseluruhan saling berkaitan satu sama lain. Berikut merupakan uraian dari masing-masing bab:

### BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penelitian.

### BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini membahas mengenai berbagai literatur yang dijadikan sebagai bahan acuan dan teori-teori yang berkaitan dengan topik yang dibahas.

### BAB III : METODOLOGI PENELITIAN

Bab ini membahas mengenai metodologi yang akan digunakan dan langkah-langkah untuk melakukan penelitian.

#### **BAB IV : PERANCANGAN DAN PENGUJIAN SISTEM**

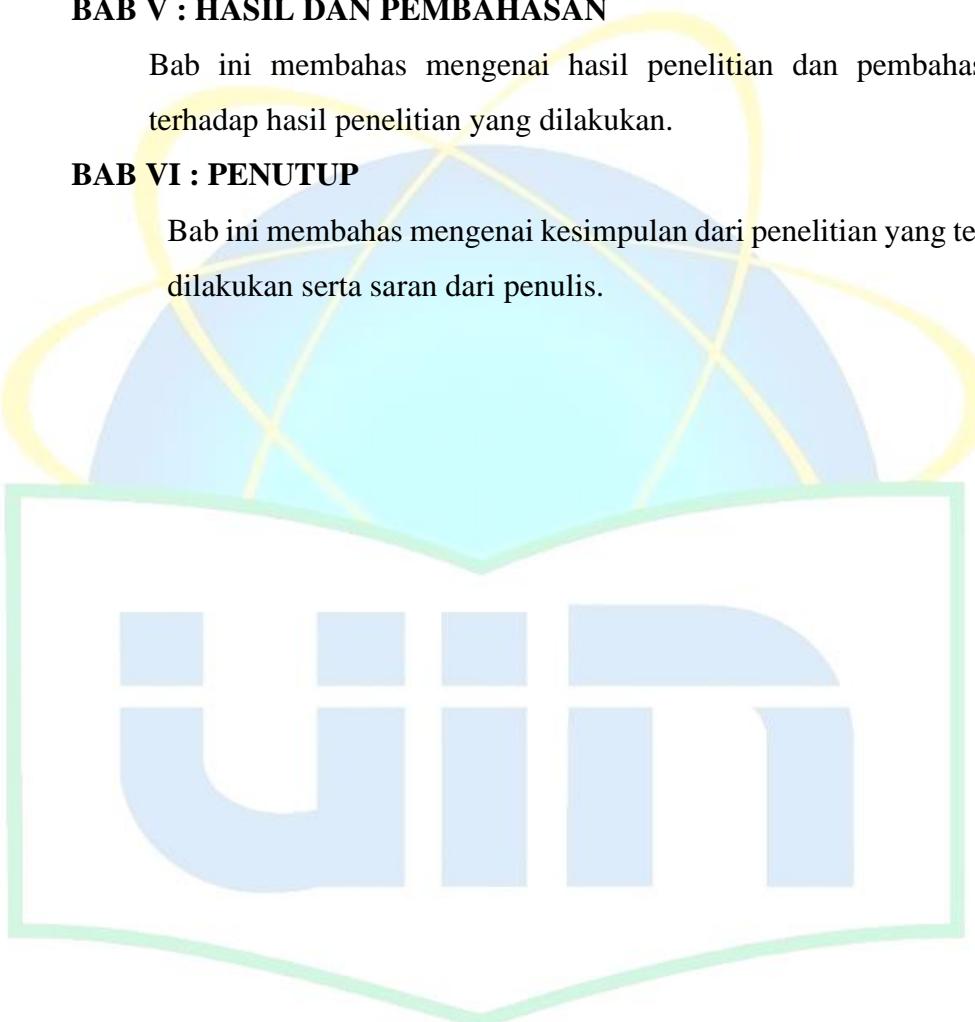
Bab ini membahas tentang simulasi sistem dari mulai perancangan sampai pengujian sistem yang sesuai dengan metode yang digunakan.

#### **BAB V : HASIL DAN PEMBAHASAN**

Bab ini membahas mengenai hasil penelitian dan pembahasan terhadap hasil penelitian yang dilakukan.

#### **BAB VI : PENUTUP**

Bab ini membahas mengenai kesimpulan dari penelitian yang telah dilakukan serta saran dari penulis.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kelapa**

Kelapa (*Cocos nucifera*) merupakan anggota tunggal dalam marga *Cocos* dari suku *Aren-arenan* atau *Arecaceae* yang tersebar di seluruh daerah tropis maupun subtropis (Abdullah & Usman, 2016). Tanaman ini berasal dari daerah pesisir Asia Tenggara (Indonesia, Malaysia, Filipina) dan Melanesia. Kemudian menyebar ke Amerika Latin, Karibia hingga ke Afrika. Saat ini tanaman kelapa telah tersebar di 200 negara di dunia (Majda, 2016).

Bagi masyarakat Indonesia, kelapa juga memiliki sejarah tersendiri yang merupakan bagian penting dari kehidupan karena kelapa memiliki banyak manfaat dan nilai ekonomi yang cukup tinggi. Mulai dari bagian batang hingga buah kelapa memiliki manfaat tersendiri. Buah kelapa adalah bagian yang paling bernilai ekonominya (Khatib et al., 2022). Arti penting tanaman kelapa bagi masyarakat juga tercermin dari luasnya area perkebunan rakyat yang mencapai 98% dari 3,74 juta Ha perkebunan kelapa (Novrianto, 2008).

#### **2.2 Jenis-jenis Kelapa**

Kelapa dibagikan menjadi tiga varietas yaitu:

1) Kelapa Varietas Dalam

Kelapa jenis ini mempunyai ciri-ciri dengan batang tinggi dan besar, dimana tingginya dapat mencapai 30 meter atau lebih. Berbuahnya tergolong agak lambat sebab, kelapa ini baru berbuah ketika sudah berumur 6-8 tahun setelah ditanam. Umur dari tanaman kelapa jenis ini bisa mencapai 100 tahun lebih. Adapun keunggulan dari jenis varietas ini adalah produksi kopranya tinggi, yaitu pertahun bisa menghasilkan 1 ton kopra/Ha, daging buahnya tebal dan keras dengan kadar minyak yang tinggi, serta lebih tahan terhadap hama dan penyakit. Beberapa jenis kelapa yang masuk ke dalam varietas jenis ini antara lain yaitu *viridis*

(kelapa hijau), *rubescens* (kelapa merah), *macrocorpu* (kelapa kelabu), *sakarina* (kelapa manis)(Sirnawati, 2023).

### 2) Kelapa Varietas Genjah

Kelapa varietas genjah ini mempunyai ciri-ciri dengan batang besar namun tidak terlalu tinggi, berbuah lebat tetapi mudah dipengaruhi oleh iklim dan lingkungan. Memiliki ukuran buah yang relatif kecil dengan kadar kopra rendah, yaitu 130 gr/buah, sementara kadar minyaknya 65% dari bobot kering daging buah. Beberapa jenis kelapa yang termasuk dalam varietas ini antara lain yaitu *eburnea* (kelapa gading), *regia* (kelapa raja), *pumila* (kelapa puyuh), dan *pretiosa* (kelapa raja malabar) (Sirnawati, 2023).

### 3) Kelapa Varietas Hibrida

Kelapa varietas hibrida merupakan jenis kelapa yang diperoleh dari hasil persilangan antara varietas Genjah dan varietas Dalam. Hasil persilangan itu merupakan kombinasi yang baik dari kedua jenis varietas aslinya. Dimana keunggulan dari jenis kelapa varietas ini adalah lebih cepat berbuah dengan waktu sekitar 3-4 tahun setelah tanam, produksi kopra tinggi sekitar 6-7 ton/ha/tahun pada umur 10 tahun dengan produktivitas sekitar 140/butir/pohon/tahun, produktivitas tandan buah sekitar 12 tandan yang berisi sekitar 10-20 butir kelapa, daging buahnya keras dan tebal dengan ketebalan sekitar 1,5cm, serta kandungan minyaknya tinggi (Sirnawati, 2023).

## 2.3 Kopra

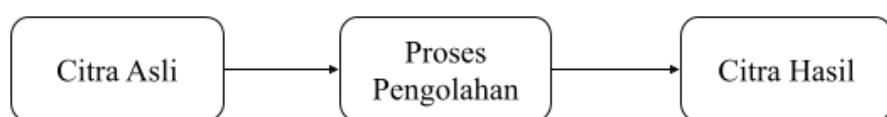
Kopra berasal dari daging buah kelapa (*Cocos nucifera*) yang diolah secara konvensional dengan cara dijemur atau di asapkan. Buah kelapa yang sudah kering harus segera dikeringkan menjadi kopra. Jika tetap berair maka permukaan daging buah kelapa akan berlendir dan berwarna kuning(Abdullah et al., 2017). Cara pengeringan buah kelapa digolongkan dengan dua cara, yaitu cara pertama dengan Cahaya matahari yaitu daging kelapa yang masih melekat pada tempurung dijemur langsung dibawah sinar matahari. Sedangkan cara kedua dengan pemanasan api yaitu dengan mengadakan kontak langsung

daging buah kelapa dengan panas api yang ada dari pembakaran. Umumnya cara ini disebut sebagai pengasapan (Abdullah et al., 2017).

Terdapat tiga jenis kopra, yaitu edible, regular, dan reject (Lahay et al., 2023). Kopra edible adalah kopra yang memenuhi standar kebersihan dan keamanan pangan yang ditetapkan oleh badan pengawas makanan. Kopra jenis ini dapat dijadikan sebagai bahan baku dalam produksi minyak kelapa, makanan ringan, produk susu kelapa, dan sebagainya. Kopra regular adalah kopra yang tidak memenuhi standar kebersihan dan keamanan untuk dikonsumsi langsung, tetapi masih dapat digunakan dalam berbagai industri lainnya. Kopra reguler biasanya digunakan dalam industri kosmetik, sabun, minyak sayur, dan produk-produk non-pangan lainnya. Kopra reject adalah kopra yang tidak memenuhi standar kualitas yang diperlukan untuk digunakan dalam industri makanan, minuman, atau industri lainnya. Kopra ini mungkin memiliki kualitas yang rendah, terkontaminasi, atau rusak secara fisik. Biasanya, kopra reject dianggap tidak dapat digunakan untuk tujuan apa pun dan harus dibuang.

#### 2.4 Pengolahan Citra

Pengolahan citra adalah proses manipulasi dan analisa citra dengan bantuan komputer dengan menggunakan deteksi tepi yang menentukan titik-titik tepi dari objek (Putri, 2016). Pada pengolahan citra terdapat proses input dan output, dimana ketika citra yang di input memiliki kualitas yang kurang baik karena ada noda atau bintik-bintik hitam, maka hasil output dari citra yang dihasilkan menjadi kurang jelas informasinya. Maka dari itu, perlu dilakukan pengolahan citra untuk memperbaiki kualitas citra dengan menghilangkan noda dan bintik-bintik hitam sehingga hasil citra yang baru dapat lebih baik dan juga dengan informasi yang lengkap (Kusumanto et al., 2011).



Gambar 2. 1 Pengolahan Citra

Tujuan dari pengolahan citra yaitu untuk meningkatkan kualitas citra dari asli menjadi lebih baik. Di dalam proses pengolahan citra terdapat Analisa citra yang bertujuan untuk mengidentifikasi ciri-ciri yang akan digunakan untuk dijadikan parameter dalam penelitian dengan citra tersebut. Tahapan-tahapan dalam menganalisa suatu objek citra terdiri dari preprocessing, segmentasi dan ekstraksi ciri. Deteksi tepi merupakan salah satu faktor yang penting untuk ekstraksi ciri suatu citra. Deteksi tepi kurang bagus jika tidak digunakan dengan tepat karena mengandung nilai dari citra tersebut. Setiap objek citra memiliki batas tepi yang memiliki kegunaan sebagai proses untuk tahapan segmentasi pada citra (Soepomo, 2013).

## 2.5 *Preprocessing* Citra

*Preprocessing* citra adalah proses yang dilakukan pada citra agar dapat diproses pada tahap selanjutnya. Tujuan dari *Preprocessing* citra yaitu untuk meningkatkan kualitas citra. Pada penulisan ini, untuk tahap *preprocessing* citra dilakukan *resize* ukuran citra untuk memperkecil ukuran agar penulisan dapat berjalan lebih cepat (Wuryani & Agustiani, 2021), penghapusan *background* yang tidak dibutuhkan dari citra dan, pemotongan bagian berdasarkan *bounding box*.

### 1) *Resize*

Proses *resize* dilakukan untuk mengubah ukuran data yang sebelumnya memiliki dimensi yang berbeda-beda. Tujuan utama dari proses ini adalah untuk mengurangi waktu pelatihan model. Misalnya, jika kita memiliki ukuran gambar yang besar, dengan mengubah ukuran gambar, waktu yang dibutuhkan untuk melatih model dapat dikurangi tanpa mengurangi kinerja model secara keseluruhan (Febriawan, 2022).

### 2) *Remove Background*

Proses ini dilakukan untuk menghapus latar belakang yang tidak relevan atau mengganggu sebuah gambar atau citra. Tujuannya untuk

memfokuskan objek dan menyeragamkan latar belakang agar terlihat perbedaan antara objek dan latar belakang (Sutanto et al., 2022).

### 3) *Cropping*

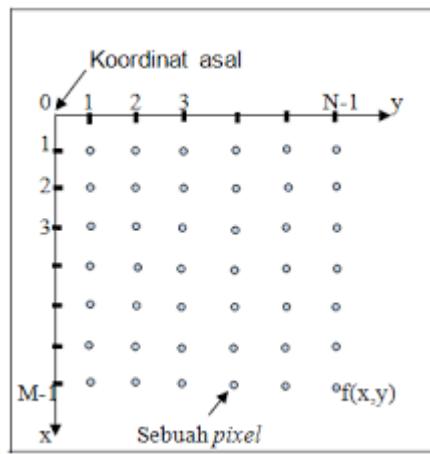
Proses *cropping* dilakukan untuk memperkecil ukuran sebuah citra dengan cara memotong citra pada koordinat yang telah ditentukan pada suatu area pada citra. Hasil proses *cropping* dapat memudahkan peneliti fokus pada obyek yang diperlukan (Yuhandri, 2019).

## 2.6 Citra Digital

### 2.6.1 Definisi Citra Digital

Berdasarkan bentuk sinyal penyusunnya, citra dapat digolongkan menjadi dua jenis yaitu citra digital dan citra analog. Citra analog adalah citra yang dibentuk dari sinyal analog bersifat kontinyu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit (Yuhandri et al., 2022).

Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel yang memiliki koordinat  $(x,y)$  dan amplitudo  $f(x,y)$ . koordinat  $(x,y)$  menunjukkan letak atau posisi piksel dalam suatu citra, sedangkan amplitudo  $f(x,y)$  menunjukkan nilai intensitas warna citra. Gambar berikut menunjukkan posisi koordinat citra digital (Riontopah, 2021).



Gambar 2. Koordinat Citra Digital

Nilai pada suatu irisan antara baris dan kolom (pada posisi  $x,y$ ) disebut dengan picture elements, image elements, pels, atau pixels. Istilah pixel paling sering digunakan pada citra digital.

### 2.6.2 Pembentukan Citra Digital

Pembentukan citra digital terdiri dari beberapa proses, yaitu akuisisi citra, *sampling*, dan kuantisasi.

#### 1) Akuisisi Citra

Proses akuisisi citra adalah pemetaan suatu pandangan menjadi citra kontinu dengan menggunakan sensor. Ada beberapa macam sensor yang dapat digunakan pada akuisisi citra, yaitu sensor tunggal, sensor garis, dan sensor larik. Sensor tersebut nantinya akan mengubah intensitas dan frekuensi cahaya menjadi gelombang analog. Selanjutnya gelombang analog akan diubah menjadi sinyal digital (Edha et al., 2020).

Pada setiap baris sinyal, diawali oleh *colorburst* sebagai informasi gelombang warna dan antar baris dibatasi oleh sinyal sinkronisasi (*sync pulse*). Sementara itu, sinyal tersebut terbentuk dalam fungsi waktu  $t$  dan amplitudo  $A$ . Untuk mendapatkan citra digital, sinyal analog harus menjalani proses digitalisasi yang terdiri dari proses *sampling* dan kuantisasi (Kartika, 2021).

## 2) Sampling

Proses sampling adalah proses konversi data analog ke dalam format digital pada koordinat x,y. Pada proses ini dilakukan penentuan jumlah sampel piksel. Dimana kecepatan proses sampling dipengaruhi oleh resolusi citra yang diinginkan. Seperti yang disebutkan di atas, hasil dari sensor pada tahap akuisisi citra masih berupa citra kontinu. Dimana hasil tersebut merupakan fungsi kontinu  $f(x,y)$ . fungsi tersebut merupakan sinyal kontinu pada nilai x,y dan juga amplitudonya. Nilai x,y yang kontinyu akan diubah menjadi bentuk diskrit (Kartika, 2021)..

## 3) Kuantisasi

Proses kuantisasi adalah proses perubahan nilai amplitudo kontinu menjadi nilai baru berupa nilai diskrit. Nilai amplitudo yang di kuantisasi adalah nilai-nilai pada koordinat diskrit hasil dari proses sampling (Kartika, 2021).

### 2.6.3 Jenis-jenis Citra Digital

Berdasarkan representasi ruang warna, citra terbagi menjadi tiga jenis yaitu, citra berwarna, citra keabuan, dan citra biner. Berikut merupakan penjelasan dari masing-masing jenis citra.

#### 1) Citra berwarna atau RGB

RGB merupakan jenis citra yang masing-masing pikselnya memiliki warna tertentu, warna tersebut terdiri dari merah (*Red*), Hijau (*Green*), dan biru (*Blue*) (Farokhah, 2020). RGB memiliki nilai interval [0,255], dimana 255 merepresentasikan warna putih, sedangkan 0 mempresentasikan warna hitam. Citra berwarna atau RGB memiliki nilai yang lebih besar daripada citra keabuan (*grayscale*) karena memiliki z layer. Penyimpanan warna asli ini sebesar 1 byte = 8 bit atau sebaliknya, yang berarti masing-masing warna memiliki tingkat atau penyebaran sampai 256 warna. Terdapat 16 jutaan atau sama dengan  $2^{24}$  kombinasi warna citra. Citra berwarna ini terdiri dari tiga matriks

yang mewakili nilai-nilai merah, hijau, dan biru untuk setiap pikselnya (Siregar et al., 2021).

## 2) Citra Keabuan (*grayscale*)

Citra keabuan merupakan jenis citra yang setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Citra keabuan memiliki interval nilai [0,255], dimana 255 mempresentasikan pada warna putih dan 0 cenderung pada warna hitam. Citra keabuan dikodekan dengan 8 bit nilai atau 1 byte ( $2^8$ ) sehingga memiliki nilai yang lebih besar daripada citra biner (Siregar et al., 2021). Citra keabuan merupakan hasil dari rata-rata color image, dengan demikian maka persamaannya dapat dituliskan sebagai berikut:

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.2)$$

Dimana  $I_R(x, y)$  adalah nilai piksel *Red* titik  $(x, y)$ ,  $I_G(x, y)$  adalah nilai piksel *Green* titik  $(x, y)$ ,  $I_B(x, y)$  adalah nilai piksel *Blue* titik  $(x, y)$ , dan  $I_{BW}(x, y)$  adalah nilai piksel *Greyscale* titik  $(x, y)$

## 3) Citra Biner

Citra biner merupakan citra yang setiap pikselnya hanya terdiri dari warna hitam atau putih, dimana hanya perlu 1 bit per piksel 1 (putih) dan 0 (hitam). Apabila dikodekan dalam 8 bit menjadi 0 dan 255, sehingga sangat efisien dalam hal penyimpanan. Citra yang direpresentasikan dengan biner sangat cocok untuk teks, sidik jari, atau gambar arsitektur (Siregar et al., 2021). Citra biner merupakan hasil pengolahan dari hitam dan putih gambar, dengan menggunakan fungsi sebagai berikut.

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (2.3)$$

Dan dalam bentuk *floating point*

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (2.4)$$

Dimana  $I_{BW}(x, y)$  merupakan nilai piksel gray titik (x,y), sedangkan  $I_{Bin}(x, y)$  merupakan nilai piksel biner titik (x,y), dan  $T$  merupakan *threshold* (Kusumanto et al., 2011).

## 2.7 Ekstraksi Fitur

Ekstraksi fitur merupakan proses pengindeksan suatu database berupa citra dengan isinya. Salah satu proses ekstraksi fitur adalah menganalisa berdasarkan isi visual seperti warna, bentuk, dan tekstur (Luthfi et al, 2019).

### 2.7.1 Ekstraksi Fitur Warna

#### 2.7.1.1 Warna RGB

Warna RGB merupakan model warna yang terdiri dari tiga komponen warna primer yaitu Red, Green, dan Blue. Jika ketiga komponen warna tersebut digabungkan, maka akan menghasilkan satu warna baru (Farokhah, 2020). Dalam model warna RGB, sebuah warna direpresentasikan dengan menentukan sejauh mana warna merah (red), hijau (green), dan biru (blue) dicampurkan. Warna ini dituliskan dalam bentuk triplet RGB, di mana setiap komponen dapat memiliki nilai dari 0 hingga nilai maksimum yang telah ditetapkan, biasanya 255. Misalnya, warna putih dalam RGB direpresentasikan dengan nilai maksimum pada ketiga komponen warna, yaitu (255, 255, 255), sementara warna hitam direpresentasikan dengan nilai minimum pada ketiga komponen warna, yaitu (0, 0, 0). Warna lainnya dapat dihasilkan dengan mengatur kombinasi nilai-nilai antara 0 hingga 255 untuk setiap komponen warna. Menggunakan model warna RGB, berbagai warna dapat direproduksi dengan presisi yang tinggi. Penggunaan triplet RGB memungkinkan pengontrolan yang fleksibel terhadap intensitas setiap komponen warna, sehingga memungkinkan penciptaan berbagai nuansa dan variasi warna (Rahmanto et al., 2022).

#### 2.7.1.2 Warna HSV

Warna HSV menunjukkan ruang warna dalam tiga kategori komponen utama yaitu *hue*, *saturation*, dan *value*. *Hue* adalah sudut 0 sampai 360 derajat, biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah *cyan*, 240 derajat adalah biru, dan 300 derajat adalah warna *magenta*. *Hue* menunjukkan jenis warna (seperti merah, biru, atau kuning) atau corak warna yang menjadi tempat asal warna ditemukan dalam spektrum warna. *Saturation* merupakan ukuran seberapa besar kemurnian warna tersebut (Areni et al., 2019). Sebagai contoh suatu warna yang semuanya merah tanpa putih adalah saturasi penuh. Jika ditambahkan putih ke merah, warna merah akan digeser ke merah muda, maka akan menghasilkan warna yang berwarna-warni. *Hue* masih tetap merah tetapi nilai saturasinya berkurang. Saturasi biasanya bernilai 0 sampai 1 dan menunjukkan nilai keabu-abuan warna di mana 0 menunjukkan abu-abu dan 1 menunjukkan warna primer murni. Komponen ketiga dari warna HSV adalah *value* atau disebut juga intensitas yaitu ukuran seberapa besar kecerahan dari suatu warna atau seberapa besar cahaya datang dari suatu warna. Value dapat bernilai 0 sampai 100% (Areni et al., 2019).

#### 2.7.1.3 Warna Grayscale

Warna *grayscale* merupakan model yang setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Proses pengubahan citra berwarna menjadi *grayscale* dapat dilakukan dengan menyederhanakan citra berwarna (RGB). Setiap proses pengubahan citra dilakukan perhitungan dengan ketiga layernya, dimana ketiga perhitungan tersebut sama pada tiap layer (Munantri et al., 2020). Untuk mendapatkan citra dalam bentuk *grayscale*, dapat menggunakan rumus berikut:

$$Grayscale = 0,3R + 0,59G + 0,11B \quad (2.5)$$

#### 2.7.1.4 Momen Warna

Moment warna merupakan representasi suatu nilai ciri warna untuk memberikan karakter pada setiap warna agar dapat dibedakan atau dipisahkan sesuai dengan ciri-ciri yang dimilikinya. Terdapat tiga momen warna yang terbukti efisien dan efektif untuk mewakili distribusi warna dalam gambar yaitu *mean*, standar deviasi, *skewness* (Rakhmawati et al., 2018). Pada penelitian ini hanya mengekstrak dua momen warna yaitu *mean*, Standar Deviasi dan variasi. *Mean* merupakan rata-rata nilai piksel pada masing-masing *channel* R, G, B, H, S, V, dan Gray. *Mean* digunakan untuk merepresentasikan nilai rata-rata pada masing-masing komponen warna. Untuk menghitung nilai *mean* dari momen warna menggunakan persamaan berikut (Kadafi, 2018).

$$\mu = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad (2.6)$$

Keterangan:

$\mu$  : Nilai *mean*

$x_i$  : nilai *sample* ke-i

n : Jumlah piksel

Selanjutnya persamaan yang digunakan untuk menghitung nilai standar deviasi adalah sebagai berikut.

$$\sigma = \sqrt{\frac{N \sum_{i=1}^n (x_i - \mu)^2}{n-1}} \quad (2.7)$$

Keterangan:

$\sigma$  : Standar deviasi

$x_i$  : nilai x ke-i

$\mu$  : nilai *mean*

n : Jumlah piksel

## 2.7.2 Ekstraksi Fitur Bentuk

Bentuk merupakan ciri fisik yang dapat dilihat dengan menggunakan indera manusia. Agar bentuk dari suatu objek dapat dibedakan dengan objek

lainnya, digunakan beberapa parameter dari ekstraksi fitur bentuk diantaranya *area* dan *perimeter* (Maulidin Yusuf et al., 2021).

- 1) *Area* atau luas, berfungsi untuk menghitung luas daerah berdasarkan jumlah *pixel* yang menempati objek gambar.
- 2) *Perimeter* atau keliling, berfungsi untuk menghitung banyaknya *pixel* yang ada pada wilayah tepi dari objek.

### 2.7.3 Ekstraksi Fitur Tekstur

Tekstur adalah sifat karakteristik yang dimiliki oleh suatu daerah (di dalam citra) yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Pengertian dari tekstur dalam hal ini kurang lebih adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital (Kadir, A et al, 2012). Untuk membedakan tekstur objek satu dengan objek yang lainnya, perlu dilakukan proses ekstraksi ciri tekstur. Salah satu ciri tekstur yang dapat digunakan adalah dengan menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM).

*Gray Level Co-Occurrence Matrix* (GLCM) adalah metode yang yang melakukan analisis terhadap piksel pada citra untuk mengetahui tingkat keabuan yang sering terjadi (Mukti et al., 2013). Metode ini juga digunakan untuk mendapatkan nilai statistic orde ke-2 dengan menghitung nilai probabilitas hubungan kedekatan antara dua buah piksel pada jarak ( $d$ ) dan sudut ( $\theta$ ) tertentu (Asery et al., 2016). Metode GLCM bekerja dengan membentuk korelasi pada data citra dan menentukan ciri fungsi matriks piksel. Kookurasi adalah kejadian ketika satu level piksel saling bertetangga dengan nilai berdasarkan jarak dan orientasi sudut. Jarak dapat berupa 1-10 piksel, sementara orientasi sudut terdiri dari  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$  (Mukherjee et al., 2016). Untuk perhitungan dengan nilai besaran yang dicari yaitu:

- 1) *Contrast*, *contrast* adalah suatu metrik yang mengukur tingkat perbedaan nilai keabuan di sekitar area pada citra. Jika terdapat variasi warna atau skala keabuan yang rendah dalam citra, dan nilai piksel yang bertetangga

memiliki nilai yang sama, maka kontras akan memiliki nilai 0. Berikut merupakan rumus untuk menghitung *contrast*:

$$Contrast = \sum_{i,j} (i - j)^2 P(i,j) \quad (2.8)$$

Dimana  $i$  adalah baris matriks,  $j$  adalah kolom matriks,  $P(i,j)$  adalah elemen matriks korelasi baris ( $i$ ) dan kolom ( $j$ )

- 2) *Dissimilarity*, *dissimilarity* merupakan ukuran yang mendefinisikan variasi tingkat intensitas pasangan piksel dalam citra, yang didefinisikan pada rumus berikut:

$$Dissimilarity = \sum_{i,j} |i - j| P(i,j) \quad (2.9)$$

Dimana  $i$  adalah baris matriks,  $j$  adalah kolom matriks,  $P(i,j)$  adalah elemen matriks korelasi baris ( $i$ ) dan kolom ( $j$ )

- 3) *Homogeneity*, *homogeneity* merupakan ukuran kemiripan citra. Nilai homogenitas akan bernilai tinggi jika semua piksel memiliki kesamaan nilai. Berikut merupakan rumus untuk menghitung *homogeneity*:

$$Homogeneity = \sum_{i_1} \sum_{i_2} \frac{P(i_1 i_2)}{1 + (i_1 - i_2)^2} \quad (2.10)$$

Dimana  $i$  adalah baris matriks,  $j$  adalah kolom matriks,  $P(i,j)$  adalah elemen matriks korelasi baris ( $i$ ) dan kolom ( $j$ )

- 4) *Energy*, *energy* menyatakan ukuran ketidakteraturan tingkat keabuan dalam citra, di mana nilai yang tinggi menunjukkan elemen-elemen GLCM memiliki nilai yang relatif serupa, sementara nilai yang rendah terjadi ketika elemen-elemen GLCM memiliki nilai 0 atau 1. Berikut merupakan rumus untuk menghitung *contrast*:

$$Energy = \sum_{i,j} P(i,j)^2 \quad (2.11)$$

Dimana  $i$  adalah baris matriks,  $j$  adalah kolom matriks,  $P(i,j)$  adalah elemen matriks korelasi baris ( $i$ ) dan kolom ( $j$ )

- 5) *Correlation*, *correlation* merupakan ukuran ketergantungan linear dari derajat tingkat keabuan dalam citra yang dapat menunjukkan struktur linear gambar pada setiap piksel yang saling bertetangga dengan objek citra. Nilai

yang terdapat pada korelasi antara -1 sampai dengan 1. Berikut merupakan rumus untuk menghitung *energy*:

$$\text{Correlation} = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j} \quad (2.12)$$

$$\mu_i = \sum_i \sum_j i p(i,j)$$

$$\mu_j = \sum_i \sum_j j p(i,j)$$

$$\sigma_i = \sqrt{\sum_i \sum_j (i - \mu_i)^2 P(i,j)}$$

$$\sigma_j = \sqrt{\sum_i \sum_j (j - \mu_j)^2 P(i,j)}$$

Dengan  $i$  adalah baris matriks,  $j$  adalah kolom matriks,  $P(i,j)$  adalah elemen matriks korelensi baris ( $i$ ) dan kolom ( $j$ ),  $\mu_i$ ,  $\mu_j$  adalah rata-rata elemen pada baris dan kolom matriks,  $\sigma_i$ ,  $\sigma_j$  adalah standar deviasi pada baris dan kolom matriks.

## 2.8 Feature Scaling

*Feature scaling* adalah suatu teknik yang digunakan dalam pengolahan data untuk mengubah skala atau rentang nilai dari fitur-fitur yang pada dataset. Rentang nilai yang tidak seimbang pada setiap atribut dapat mempengaruhi kualitas hasil dataset (Nasution et al., 2019). Oleh karena itu, dibutuhkan *feature scaling* untuk menghasilkan distribusi nilai yang seragam pada setiap fitur, sehingga fitur-fitur tersebut dapat diolah dan dibandingkan dengan baik. Terdapat beberapa metode yang dapat digunakan, pada penelitian ini menggunakan Min-Max scaling.

Min-Max scaling atau sering disebut dengan Min-max normalization, merupakan suatu teknik normalisasi data yang menggunakan transformasi linier pada data awal untuk mengimbangi nilai perbandingan sebelum dan sesudah proses. Min-max scaling mengubah ukuran dari rentang asli, sehingga semua nilai berada pada rentang 0 dan 1 (Rifatama et al., 2023).

## 2.9 Artificial Intelligence

Istilah *Artificial Intelligence* diperkenalkan pertama kali pada tahun 1956 dalam sebuah konferensi di bidang *Computer Science* di Dartmouth. Pada awalnya *Artificial Intelligence* dirumuskan sebagai sebuah pendekatan untuk memodelkan cara kerja otak manusia yang selanjutnya digunakan sebagai dasar perancangan sistem komputer yang lebih cerdas. Menurut Poole et al (1998), mereka mendefinisikan istilah *Artificial Intelligence* sebagai sebuah bidang ilmu mengenai pengembangan *intelligence agents* yang dirumuskan lebih lanjut sebagai:

*“..any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.”*

Berdasarkan definisi diatas, bidang kajian utama dari *Artificial Intelligence* adalah entitas yang mempunyai kemampuan untuk berinteraksi dengan lingkungan tempatnya berada dalam bentuk menerima data dari lingkungan tempatnya berada dan melakukan sejumlah aksi di dalam lingkungannya dengan tujuan untuk memaksimumkan probabilitas pencapaian tujuan yang ditetapkan.

Selanjutnya, pendapat lain dari Russell & Norvig (2009) mendefinisikan *Artificial Intelligence* sebagai berikut:

*“..machine (or computer) that mimic cognitive functions that humans associate with the human mind, such as learning and problem solving.”*

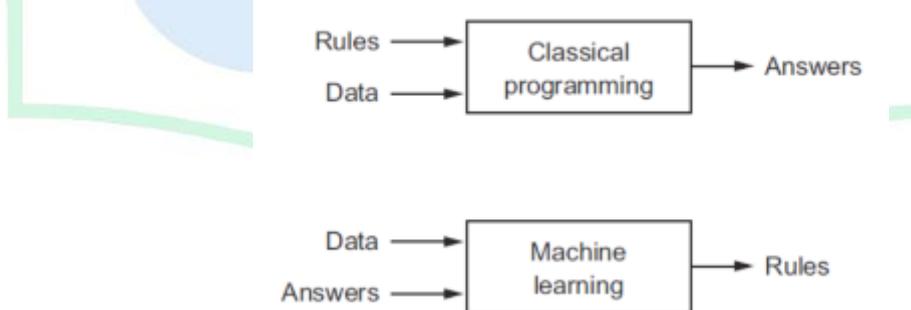
Berdasarkan definisi di atas, kajian *Artificial Intelligence* ini lebih berfokus kepada kemampuan komputer untuk meniru kemampuan berfikir manusia yaitu melakukan pembelajaran dan mengatasi masalah. Dalam dua puluh tahun terakhir, *Artificial Intelligence* telah berkembang menjadi beberapa cabang ilmu. Masing-masing cabang ilmu ini memfokuskan kepada bidang yang lebih spesifik dari kajian *Artificial Intelligence*. Salah satu cabang *Artificial Intelligence* yang berkembang dengan cepat dan banyak diimplementasikan ke

dalam sejumlah teknologi aplikatif di berbagai industri adalah *Machine Learning* dan *Deep Learning*.

### 2.1.1 Machine Learning

*Machine Learning* adalah metode analisis yang membantu menangani data besar dengan cara mengembangkan algoritma komputer. Dengan menggunakan data, pembelajaran mesin memungkinkan komputer menemukan wawasan tersembunyi tanpa diprogram secara eksplisit saat mencarinya. Contohnya, ketika ada email baru, algoritma tersebut akan menghasilkan prediksi apakah email baru itu masuk kedalam spam atau tidak (Maulana et al., 2023).

*Machine Learning* merupakan salah satu cabang dari *Artificial Intelligence* yang fokus pada pengembangan sebuah sistem yang mampu belajar “sendiri” tanpa harus berulang kali di program oleh manusia. *Machine Learning* menyediakan sistem kinerja secara otomatis serta belajar memperbaiki diri dari pengalaman tanpa diprogram secara eksplisit. Hal tersebut sesuai dengan pendapat dari Chollet Francois (2017), yang menjelaskan bahwa ciri utama dari algoritma *Machine Learning* dibandingkan dengan pemrograman konvensional lainnya adalah kemampuannya untuk membuat komputer memprogram dirinya sendiri.



Gambar 2. Perbedaan Pemograman Tradisional dan Machine Learning

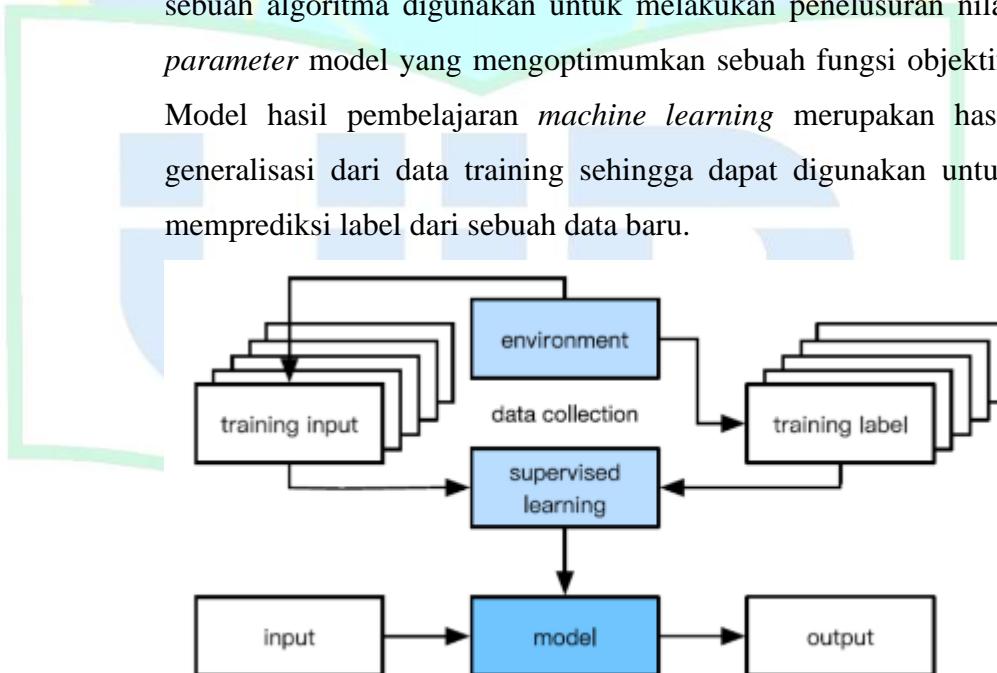
(Sumber : Francois, 2017)

Berdasarkan Gambar 2.4 Pada pemrograman konvensional, data dan *rules* (program) dijalankan pada sebuah komputer untuk menghasilkan *output*. Sedangkan pada *machine learning*, data dan *output* dijalankan didalam sebuah komputer untuk menghasilkan *rules* (program). Program hasil pembelajaran algoritma *machine learning* selanjutnya digunakan untuk memproses *input* menjadi *output* sesuai dengan tugas yang harus dilakukan.

Metode pembelajaran model *machine learning* dikelompokan kedalam tiga kategori utama sebagai berikut:

### 1) *Supervised learning*

Merupakan metode pembelajaran yang bertujuan untuk memprediksi variabel target data baru berdasarkan model yang sudah ditraining menggunakan himpunan data berlabel (Handayani, 2022). Dalam proses pembelajaran *machine learning*, sebuah algoritma digunakan untuk melakukan penelusuran nilai *parameter* model yang mengoptimalkan sebuah fungsi objektif. Model hasil pembelajaran *machine learning* merupakan hasil generalisasi dari data training sehingga dapat digunakan untuk memprediksi label dari sebuah data baru.



Gambar 2. 4 *Supervised Learning*  
(Sumber: Francois, 2017)

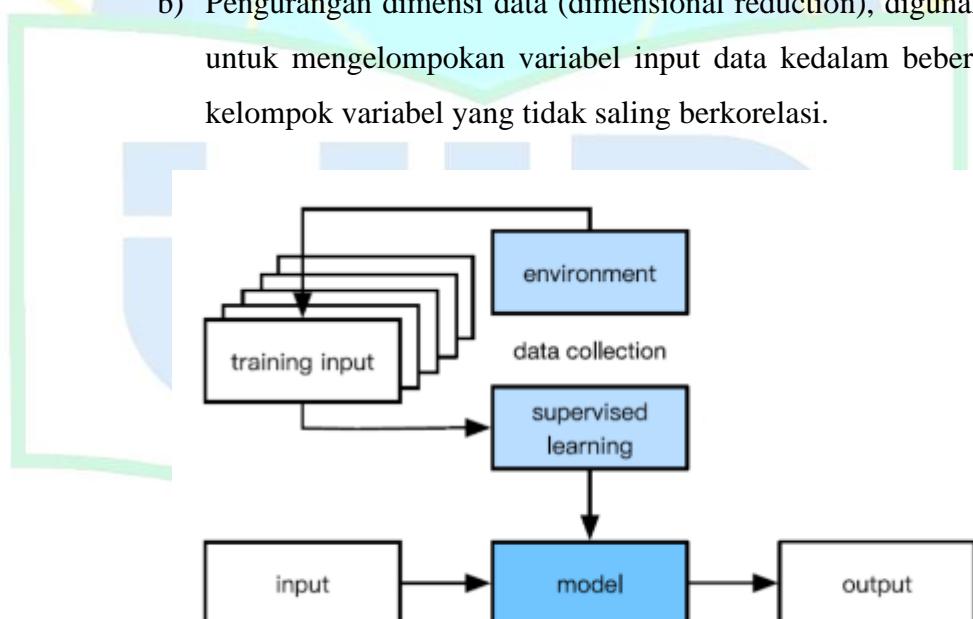
Teknik pembelajaran *supervised learning* biasanya digunakan untuk menyelesaikan dua kategori masalah komputasi yaitu:

- a) Masalah klasifikasi, digunakan untuk memprediksi variabel target berupa data kategorikal.
- b) Masalah regresi, digunakan untuk memprediksi variabel target berupa

### 2) *Unsupervised learning*

Merupakan metode pembelajaran yang menggunakan sejumlah contoh data tanpa label. Teknik unsupervised learning ini biasanya digunakan untuk menyelesaikan dua kategori masalah yaitu (Angga et al, 2023):

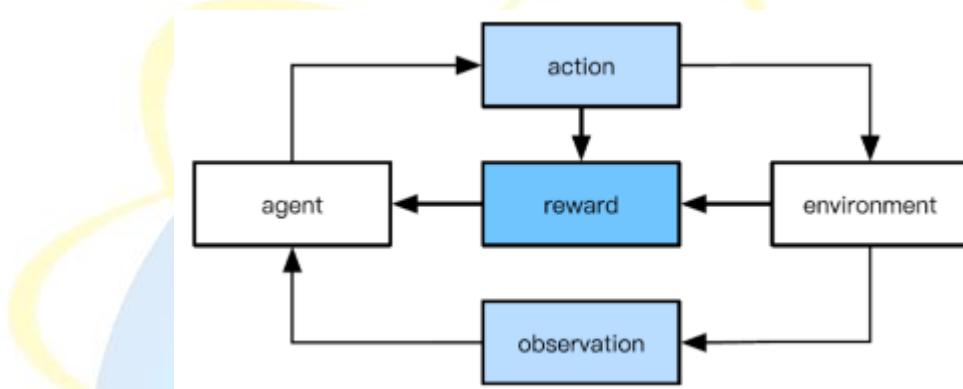
- a) Klastering, digunakan untuk mengelompokan data training kedalam beberapa klaster berdasarkan kesamaan fitur.
- b) Pengurangan dimensi data (dimensional reduction), digunakan untuk mengelompokan variabel input data kedalam beberapa kelompok variabel yang tidak saling berkorelasi.



Gambar 2. 5 *Unsupervised Learning*  
(Sumber : Francois, 2017)

### 3) *Reinforcement learning*

Merupakan metode pembelajaran yang didasarkan pada pemberian penghargaan terhadap keputusan yang diinginkan dan atau pemberian penalti terhadap keputusan yang tidak diinginkan. Metode ini mengintegrasikan metode pembelajaran dengan sebuah aplikasi yang memberi umpan balik bagi setiap keputusan yang telah di ambil. Dengan mekanisme ini, algoritma pembelajaran dapat belajar dari kesalahan dalam membuat sebuah keputusan (Angga et al, 2023).



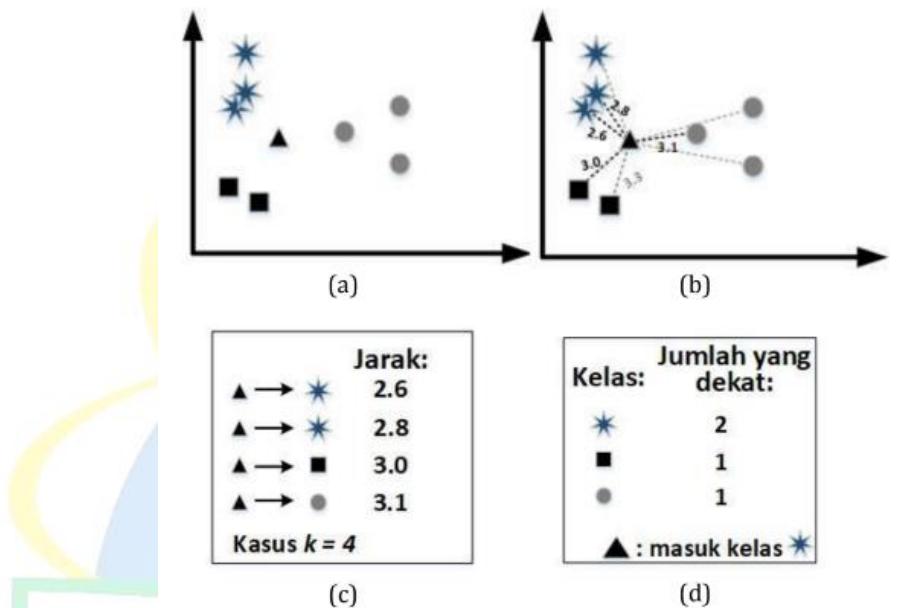
Gambar 2. 6 Reinforcement Learning  
(Sumber : Francois, 2017)

## 2.10 K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) merupakan metode klasifikasi yang sangat sederhana dalam mengklasifikasikan sebuah gambar berdasarkan jarak terdekat dengannya. Data yang memiliki jarak fitur vector terdekat akan menjadi satu kelas atau label klasifikasi. Walaupun sangat sederhana dan terdapat kelemahan jika sebaran datanya terlalu dekat namun metode K-NN ini mudah dipahami (Farokhah, 2020).

K-Nearest Neighbors (K-NN) masuk ke dalam kategori *supervised machine learning* yang dapat digunakan untuk klasifikasi dan regresi. Cara kerja metode ini dengan menggunakan data latih (*training data*) dan menghasilkan data tes berdasarkan jarak ke data latih tersebut. Tujuannya mencari sejumlah  $k$  tetangga terdekat yang ditentukan melalui label kelas terbanyak pada jangkauan  $k$  tetangga terdekat tersebut. Ilustrasi metode ini

seperti pada **Gambar 2.8**. Untuk kemudahan ilustrasi digunakan sumbu dua dimensi dengan koordinat  $x_1$  dan  $x_2$ . Hal tersebut mewakili dua nilai sensor (Romy, 2022). Dalam ilustrasi ini terdapat tiga kelas yang menjadi target klasifikasi.



Gambar 2. 7 Ilustrasi Algoritma K-NN

(Sumber : Romi, 2022)

Gambar 2.7c) Dalam contoh ilustrasi ini, akan dicari apakah data tes  $\blacktriangle$  masuk ke suatu kelas, apakah masuk ke dalam kelas  $\star$  atau  $\blacksquare$  atau  $\bullet$  ?

Metode KNN menghitung jarak antara data tes  $\blacktriangle$  dengan semua data latih. Pada Gambar 2.7b) terlihat perhitungan yang menghasilkan jarak 2.6, 2.8, 3.0, 3.1, dan 3.3. Pada Gambar 2.7c) digunakan nilai  $k = 4$  sehingga diurutkan empat jarak yang paling dekat sesuai dengan rankingnya. Terlihat bahwa jarak paling dekat adalah 2.7 Gambar 2.7d) pada langkah ini dihitung berapa jumlah label yang dekat pada kelas-kelas tersebut. Terlihat ada 2 label pada kelas  $\star$ , 1 label pada kelas  $\blacksquare$ , dan 1 label pada kelas  $\bullet$ . Dari sini dapat dilihat bahwa data tes diklasifikasikan ke dalam kelas  $\star$ .

Metode KNN secara umum terdiri atas dua langkah, yaitu pelatihan untuk menyimpan setiap pola latih dan klasifikasi. Pada saat klasifikasi sebuah pola, KNN memeriksa semua pola latih untuk menemukan sejumlah  $k$  pola terdekat. Proses pelatihan pada KNN menghasilkan  $k$  yang memberikan akurasi tertinggi dalam menggeneralisasi data-data yang akan datang. Salah satu tantangan pada proses pelatihan adalah menentukan nilai  $k$  yang optimum. Nilai  $k$  dapat ditentukan berdasarkan jarak, kesamaan, dan ketidaksamaannya sesuai jenis fitur data. Pada kasus di atas, penentuan  $k$  menggunakan jarak. Umumnya digunakan perhitungan jarak *Euclidean*, *Minkowski*, *Manhattan*, dan *Chebyshev distance* (Nooraeni & Nurfalah, 2022).

Pada Gambar 2.7b) perhitungan jarak menggunakan *Euclidean distance* untuk dua titik, dengan rumus persamaan seperti berikut.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.13)$$

Langkah-langkah klasifikasi dirangkum sebagai berikut:

- 1) Menentukan nilai  $k$ .
- 2) Menghitung jarak antara data tes yang baru dan tetangga terdekat sejumlah  $k$  pada latih.
- 3) Memeriksa dengan *voting* di kelas manakah yang terbanyak jumlah tetangga. Yang memiliki jumlah tetangga terdekat yang terbanyak akan dipilih sebagai hasil klasifikasi.

Terdapat beberapa variasi dari algoritma K-Nearest Neighbor yaitu sebagai berikut:

- 1) *Modified K-Nearest Neighbor* (MKNN)

*Modified K-Nearest Neighbor* (MKNN) variasi dari algoritma K-Nearest Neighbor (KNN). Pada algoritma ini dilakukan perhitungan data latih dengan tetangga terdekatnya. Kemudian hasil dari perhitungan data latih tersebut diklasifikasikan dengan data uji yang telah

ditentukan. Pada proses MKNN terdapat dua proses perhitungan, yaitu perhitungan validitas dan proses *weight voting* (Ravi et al., 2019).

2) *Weighted K-Nearest Neighbor* (WKNN)

*Weighted K-Nearest Neighbor* (WKNN) adalah variasi dari algoritma *K-Nearest Neighbor* (KNN) di mana tetangga terdekat akan diberikan bobot berdasarkan preferensi atau informasi tambahan yang relevan. Tujuannya untuk memberikan preferensi atau penekanan yang berbeda pada setiap tetangga terdekat sesuai dengan tingkat kepentingannya.(Nadwah et al., 2021).

3) *Distance-Weighted K-Nearest Neighbor* (DWKNN)

*Distance-Weighted K-Nearest Neighbor* (DWKNN) adalah variasi dari algoritma *K-Nearest Neighbor* (KNN) di mana tetangga terdekat akan diberikan bobot berdasarkan jarak antara mereka dan titik data yang akan diprediksi. Bobot diberikan secara terbalik proporsional terhadap jarak, artinya tetangga yang lebih dekat diberi bobot yang lebih besar. Bobot jarak digunakan untuk menggambarkan tingkat pengaruh atau relevansi tetangga terhadap titik data yang akan diprediksi (Chrismanto et al., 2020).

4) *K- Nearest Neighbor Ball Tree*

*K- Nearest Neighbor Ball Tree* adalah variasi dari algoritma *K-Nearest Neighbor* (KNN) yang menggunakan struktur data *Ball Tree* untuk menyimpan titik-titik data pada setiap node-nya. Struktur data ini dibuat dengan membagi ruang pencarian menjadi bola-bola kecil yang masing-masing berisi beberapa titik data. Dengan menggunakan struktur *Ball Tree*, algoritma *K-Nearest Neighbor* (KNN) dapat mencari jarak terpendek dengan lebih cepat karena hanya perlu mencari node yang dekat dengan data yang dicari (Dharmawan et al., 2023)

## 2.11 Mengukur Performansi

Sebuah sistem yang akan melakukan klasifikasi, diharapkan dapat melakukan klasifikasi semua data dengan benar, namun kenyataannya kinerja

sistem terkadang tidak bisa benar 100%. Oleh karena itu, suatu sistem klasifikasi harus diukur kinerjanya (Hanafi et al., 2019). Pada umumnya, pengukuran performansi kerja suatu model untuk klasifikasi menggunakan metrik akurasi, *confusion matrix*, dan *classification report* (Romy, 2022).

### 2.11.1 Akurasi

Skor akurasi adalah metrik yang paling sederhana, cara kerja pengukuran ini dengan membagi jumlah kasus prediksi yang benar dengan jumlah seluruh prediksi. Jika semua prediksi benar, akan menghasilkan skor akurasi = 1, dan jika semua prediksi salah maka skor akurasi = 0. Namun, terdapat kelamahan pada skor akurasi ini, sebab tidak memberikan informasi pengaruh ketidakseimbangan jumlah *false positif* dan *false negative*. Skor akurasi dapat dihitung dengan persamaan berikut:

$$Akurasi = \frac{\text{Jumlah Prediksi yang Benar}}{\text{Jumlah Seluruh Prediksi}} \quad (2.14)$$

### 2.11.2 Confusion Matrix

*Confusion Matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah (Normawati & Prayogi, 2021). Contoh *confusion matrix* untuk klasifikasi *multiclass* ditunjukkan pada tabel berikut.

Tabel 2. 1 Confusion Matrix

	Nilai Prediksi			
Nilai Aktual		1	2	3
	1	A	B	C
	2	D	E	F
	3	G	H	I

Pada *confusion matrix* untuk multi kelas, fokus pembahasan dilakukan per kelas. Misal saat ini fokus pada kelas 1, maka TP adalah A. Berikut nilai TN, FP, dan FN jika fokus pada kelas 1.

*True Positif:* Jika kelas 1, maka TP = A, Jika fokus pembahasan kelas 2, maka TP=E. Jika fokus pada kelas 3, maka TP=I. *True Negative:* Jumlah sel kecuali sel-sel pada kolom dan baris 1, maka TN=E+F+H+I. *False Positif:* Jumlah semua nilai aktual selain kelas 1 yang diprediksi sebagai kelas 1, maka FP=D+G. *False Negative:* Jumlah semua nilai prediksi yang bukan kelas 1 saat nilai aktualnya 1, maka FN=B+C (Romy, 2022).

#### Keterangan

- |                              |  |
|------------------------------|--|
| TP ( <i>True Positive</i> )  | = Jumlah dokumen dari kelas 1 yang benar diklasifikasikan sebagai kelas 1. |
| TN ( <i>True Negative</i> )  | = Jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0. |
| FP ( <i>False Positive</i> ) | = Jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. |
| FN ( <i>False Negative</i> ) | = Jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0. |

### 2.11.3 Classification Report

*Classification report* menghasilkan tiga metrik evaluasi yaitu *precision*, *recall*, dan *F1 skor*. *Precision* adalah rasio prediksi benar positif terhadap total hasil prediksi positif. *Recall* adalah rasio prediksi benar positif terhadap keseluruhan data yang sebenarnya positif. *F1 skor* adalah rata-rata harmonik dari presisi dan recall (Ridhovan & Suharso, 2022). Akurasi dan *confusion matrix* dapat dibentuk dengan persamaan berikut.

$$Precision = \frac{True\ Positif}{True\ Positif + False\ Positif} \quad (2.15)$$

$$Recall = \frac{True\ Positif}{True\ Positif + False\ Negative} \quad (2.16)$$

$$f1\ Score = \frac{2 \times Recall \times Presisi}{Recall + Presisi} \quad (2.17)$$

## 2.12 Metode Simulasi

Metode simulasi adalah metode yang digunakan untuk simulasi seperti mengikuti tingkah laku dari suatu sistem tertentu. Metode ini dapat digunakan untuk berbagai keperluan seperti desain sistem, pembangunan kebijakan sistem operasi, dan dalam penelitian untuk mengembangkan pemahaman sistem (Sargin, 2013). Menurut (Gandomi et al., 2010) yang dikutip dari skripsi (Artyani, 2019) metode simulasi mempunyai beberapa tahapan, tahapan tersebut diatur dalam satu siklus. Siklus tersebut dapat menyesuaikan dengan kompleksitas sistem sehingga tahapannya tidak selalu berurutan, berulang, maupun berlawanan arah. Terkadang, beberapa tahapan dapat dilangkahi oleh tahapan berikutnya tergantung dengan rumit atau tidaknya sistem tersebut. Berikut merupakan tahapan-tahapan dalam metode simulasi (Gandomi et al., 2010).

### 1. *Problem Formulation*

Proses simulasi dimulai dengan adanya masalah yang harus dipecahkan. Misalnya sebuah perusahaan kargo ingin mencoba untuk mengembangkan strategi baru untuk pengiriman truk. Pada tahap ini kita harus memahami perilaku dari sistem dan mengatur operasi sistem. Maka kita perlu menganalisa berbagai solusi dengan menyelidiki hasil sebelumnya dengan masalah yang sama.

### 2. *Model Conceptual*

Pada tahap ini dilakukan deskripsi tingkat tinggi dari struktur dan perilaku sebuah sistem serta mengidentifikasi semua benda dengan atribut dan antarmuka mereka. Pada tahap ini juga dilakukan penentuan *variabel state* dan aspek-aspek kunci dari *requirement*. Terakhir, mendokumentasikan informasi *non-fungsional*, misalnya seperti perubahan pada masa yang akan datang, perilaku *non-intuitive* atau *non-formal*, dan hubungan dengan lingkungan.

### 3. *Collection of Input Data*

Pada tahap ini dilakukan pembelajaran sistem untuk mendapatkan data *input* atau *output*. Untuk melakukannya, dapat mengumpulkan dan

mengamati atribut yang telah ditentukan pada tahap sebelumnya. Isu penting lainnya pada tahap ini adalah memilih ukuran sampel yang valid secara statistic dan format data yang dapat diproses dengan komputer.

#### 4. *Modelling*

Pada tahap ini dilakukan pemodelan dengan membangun representasi yang rinci dari sistem berdasarkan *conceptual model* dan *input/output* data yang dikumpulkan. Model ini dibangun dengan mendefinisikan objek, atribut, dan metode yang ingin digunakan. Pada tahap ini, spesifikasi model juga dibuat dengan set persamaan yang mendefinisikan perilaku dan struktur.

#### 5. *Simulation*

Pada tahap ini, kita harus memilih mekanisme untuk menerapkan model (dalam banyak kasus menggunakan computer dan Bahasa pemograman serta alat-alat yang memadai) dan model simulasi yang dibangun.

#### 6. *Verification and Validation*

Pada tahap ini, kita perlu memverifikasi dan memvalidasi ketiga model yang telah dibangun pada tahap sebelumnya yaitu model konseptual, sistem model, dan model simulasi. Validasi difokuskan pada korespondensi antara model dan realitas terkait hasil simulasi yang konsisten dengan sistem yang dianalisis dan membangun model yang tepat berdasarkan hasil yang diperoleh selama tahap ini.

#### 7. *Design of Experiment*

Pada tahap ini, dilakukan evaluasi *output* dari simulator menggunakan korelasi statistik untuk menentukan tingkat presisi untuk metrik kerja. Fase ini dimulai dengan desain eksperimen, menggunakan teknik yang berbeda. Beberapa Teknik ini meliputi analisis sensitive, optimasi, dan seleksi.

#### 8. *Execute Simulation and Analysis Output*

Pada tahap ini dilakukan untuk memahami perilaku sistem. *Output* digunakan untuk mendapatkan tanggapan tentang perilaku sistem yang

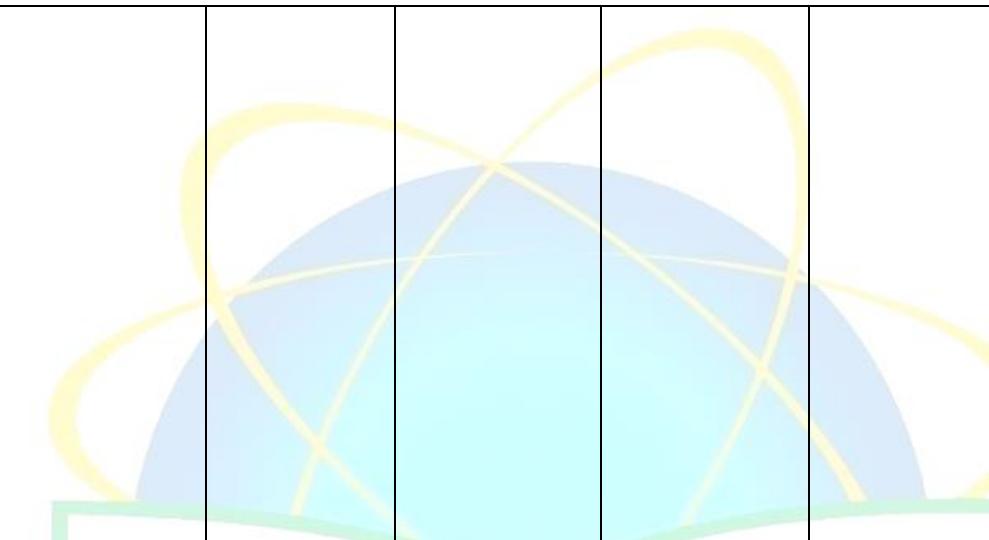
asli. Pada tahap ini, alat visualisasi dapat digunakan untuk membantu proses tersebut.



## 2.13 Studi Pustaka

Tabel 2. 2 Studi Pustaka

No	Judul	Penulis/ Tahun	Objek Penelitian	Lokasi Penelitian	Parameter Kualitas	Metode Klasifikasi	Jumlah Dataset	Evaluasi Model	Hasil
1	Klasifikasi Tingkat Kematangan Kopra Menggunakan Metode <i>naïve bayes</i>	Yunaldi M.Z. Masi Adang, Abd.Rabi,M. Kom dan Rahman Arifuddin (2020)	Tingkat kematangan kopra	Malang, Jawa Timur	Metode transformasi warna HSI dan RGB	<i>Naïve Bayes</i>	Menggunakan 30 data training dan 12 data testing	Metrik Akurasi	Berdasarkan nilai input yang digunakan yaitu r, g, b, v, *a, *b, entropi, energy, homogenitas dan metode naïve bayes dalam proses pengklasifikasian dengan membandingkan nilai probabilitas diperoleh rata-rata akurasi terbesar mencapai 91,12%.
2	Sistem Prediksi Kualitas Kopra Putih Menggunakan	Rosi Rahayu Marlis, Abdullah, dan Fitri Yunita (2021)	Prediksi kualitas kopra putih	Riau	Ekstraksi ciri warna (RGB) dan bentuk ( <i>Area and Perimeter</i> )	K-Nearest Neighbor – Euclidean Distance	Menggunakan 120 data training dan 60 data testing	Confusion Matrix	Berdasarkan pengujian dari sistem prediksi kopra putih menggunakan metode <i>Holdout</i> , maka dapat ditarik kesimpulan bahwa Metode k-Nearest

	k-Nearest Neighbor (k-NN)								Neighbor dapat melakukan prediksi terhadap jenis kopra putih berdasarkan warna dan bentuk. Akurasi yang diperoleh pada saat menggunakan k = 1 sebesar 93,33%, pada k = 3 sebesar 83,33%, dan pada k = 5 sebesar 81,67%. Akurasi tertinggi didapat pada k = 1, yakni sebesar 93,33%
3	Penentuan <i>Grade</i> Kopra dengan Penerapan Logika Fuzzy	Idham Halid Lahay, Jamal Darusalam Giu, Hasanuddin, dan Malfrin Bawole (2023)	<i>Grade</i> Kopra	Pohuwato, Gorontalo	Kadar air, kadar minyak, dan jamur.	Logika Fuzzy	Menggunakan 40 data training dan 10 data testing	Metrik Akurasi	Berdasarkan pengujian yang telah dilakukan dengan software matlab dengan logika fuzzy didapatkan tingkat akurasi sebesar 95%.
4	Sistem Prediksi Kualitas Santan Kelapa	Masparudin, Abdullah, dan Usman (2020)	Kualitas Santan Kelapa	Riau	Ekstraksi Warna RGB	<i>Nearest Mean Classifier</i> (NMC)	Menggunakan 90 data training dan 45 data testing	Metrik Akurasi	Evaluasi dilakukan dengan 3 jenis kamera smartphone yaitu kamera 1 Xiaomi Mi 8 Lite, Oppo F7, dan

	Menggunakan <i>Nearest Mean Classifier</i> (NMC)								Samsung Galaxy J3 Pro. Pada pengujian kamera pertama memiliki tingkat akurasi tertinggi yaitu 86,66% dibandingkan dengan kamera 2 dengan akurasi 60% dan kamera 3 dengan akurasi 46%.
5	Klasifikasi Jeruk Nipis Terhadap Tingkat Kematangan Buah Berdasarkan Fitur Warna Menggunakan <i>K-Nearest Neighbor</i>	Cintya Paramita, Eko Hari R, Christy Atika S, dan De Rosal M. S. (2019)	Klasifikasi Jeruk Nipis terhadap tingkat kematangannya	Semarang	Ekstraksi fitur warna RGB	K- <i>Nearest Neighbor – Euclidean Distance</i> dan <i>Cityblock Distance</i>	Menggunakan 50 data training dan 25 data testing	Metrik Akurasi	Berdasarkan hasil pengujian yang sudah dilakukan, dari 25 data citra uji, akurasi terbaik sebesar 92% saat menggunakan <i>euclidean distance</i> adalah dengan nilai $k=7$ dan $k=3$ . Sedangkan menggunakan <i>cityblock distance</i> akurasi sebesar sebesar 88% dengan nilai $k=3$ dan $k=1$ .

6	Sistem Klasifikasi Jenis Pisang Berdasarkan Ciri Warna HSV menggunakan Metode K-NN	Zenni Dwi Lestari dan Nur Nafiyah (2019)	Klasifikasi Jenis Pisang	Lamongan, Jawa Timur	Ekstraksi fitur warna HSV	<i>K-Nearest Neighbor - Euclidean Distance</i>	Menggunakan 100 data training dan 50 data testing	Metrik Akurasi	Berdasarkan hasil uji coba yang telah dilakukan, akurasi yang didapatkan dari pengujian data testing menggunakan $k=3$ sebesar 82 %.
7	Penerapan Metode <i>Grey Level Co-Occurrence Matrix</i> (GLCM) dan <i>K-Nearest Neighbor</i> (K-NN) untuk Mendeteksi Tingkat Kematangan Buah	Qurnia Shandy, Sudirman S. Panna, Yurisyanto Malago (2019)	Tingkat kematangan buah belimbing bintang	Gorontalo	<i>Grey Level Co-Occurrence Matrix</i> (GLCM)	<i>K-Nearest Neighbor - Euclidean Distance</i>	Menggunakan 50 data training dan 10 data testing	<i>Confusion Matrix</i>	Kinerja model deteksi tingkat kematangan buah belimbing bintang menggunakan metode <i>K-Nearest Neighbor</i> dengan fitur ekstraksi <i>Gray Level Co-Occurrence Matrix</i> setelah diukur menggunakan <i>Confusion Matrix</i> menghasilkan akurasi sebesar 90%

	Belimbing Bintang								
--	----------------------	--	--	--	--	--	--	--	--

Berdasarkan hasil kajian terdahulu yang relevan, penulis akan menggunakan beberapa metode dan kriteria yang telah dijabarkan sebelumnya untuk digunakan sebagai penunjang penelitian ini. Berikut merupakan kebaruan yang akan di terapkan dalam penelitian ini sebagai berikut:

1. Pada penelitian ini menggunakan dataset kopra yang terbagi menjadi tiga jenis yaitu edible, regular, dan reject dengan masing-masing jenis memiliki 613 citra sehingga total dataset yang digunakan sebanyak 1.839 citra.
2. Ekstraksi fitur yang digunakan meliputi warna, bentuk, dan tekstur. Untuk ekstraksi fitur warna, digunakan metode *color moment* yang mencakup *mean* RGB, HSV, standar deviasi, serta variasi pada *Grayscale*. Sedangkan ekstraksi fitur bentuk menggunakan *area* dan *parameter*. Dan untuk ekstraksi fitur tekstur, metode yang digunakan adalah GLCM (*Gray Level Co-occurrence Matrix*) dengan menggunakan fitur-fitur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*.
3. Klasifikasi dilakukan pada 12 skenario yang terdiri dari variasi fitur dan pembagian dataset yang berbeda.
4. Pengujian model klasifikasi menggunakan metrik akurasi, *confusion matrix*, dan *classification report*.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

Pengumpulan data merupakan salah satu prosedur sistematis yang harus dilakukan untuk memperoleh data yang dibutuhkan, dengan adanya data maka suatu penelitian dapat menjadi optimal dan efektif (Fajar, 2021).

##### **3.1.1 Studi Pustaka**

Studi pustaka dilakukan dengan membaca sumber-sumber media seperti jurnal, *e-book*, website, dan penelitian skripsi yang relevan dengan penelitian mengenai pengolahan citra digital, klasifikasi kopra, dan algoritma K-NN. Dengan tujuan untuk memperoleh informasi yang digunakan sebagai landasan teori dan referensi untuk menyelesaikan masalah yang ada.

##### **3.1.2 Wawancara**

Wawancara dilakukan dengan narasumber petani kelapa yang berada di daerah kabupaten Indragiri Hilir Provinsi Riau. Tujuan dari wawancara ini adalah untuk memperoleh wawasan yang mendalam terkait dengan berbagai aspek yang berkaitan dengan kopra. Dari hasil wawancara ini penulis akan mengumpulkan data mengenai penjelasan dari kopra edible, regular, dan reject.

#### **3.2 Metode Pengembangan Model**

Metode pengembangan model yang digunakan dalam penelitian ini adalah metode simulasi. Berikut merupakan tahapan-tahapan dari metode simulasi:

##### **3.2.1 Problem Formulation**

Tahap pertama dalam metode simulasi ini, yaitu merumuskan permasalahan yang akan dianalisa dan menentukan penggunaan simulasi

atau tidak untuk mendapatkan solusi dari permasalahan yang telah dirumuskan. Setelah mengumpulkan data, penulis merumuskan masalah bagaimana mengklasifikasikan jenis kopra berdasarkan ekstraksi fitur warna, bentuk, dan tekstur menggunakan metode *K-Nearest Neighbour*. Dari rumusan masalah tersebut, penulis memutuskan untuk menggunakan metode simulasi.

### 3.2.2 *Conceptual Model*

Setelah permasalahan dirumuskan, penulis membuat pemodelan yang akan digunakan untuk melakukan klasifikasi jenis kopra menggunakan metode *K-Nearest Neighbour*.

### 3.2.3 *Collection of Input Data*

Langkah berikutnya, menentukan *input* yang akan diproses. Input yang akan diproses pada simulasi ini yaitu tiga folder jenis kopra yang masing-masing folder tersebut terdiri dari citra dengan format .png. Dataset terdiri dari 613 data citra untuk masing-masing jenis kopra dengan total seluruhnya berjumlah 1.839 data citra. Klasifikasi jenis kopra ini dilakukan berdasarkan hasil 12 percobaan dengan mempertimbangkan kombinasi fitur, antara lain warna dan bentuk, warna dan tekstur, bentuk dan tekstur, serta warna, bentuk, dan tekstur. Setiap kombinasi fitur tersebut diuji pada tiga pembagian sampel, yaitu 50:50, 70:30, dan 90:10.

Untuk ekstraksi fitur warna, digunakan metode *color moment* yang mencakup *mean* RGB, HSV, standar deviasi, serta variasi pada *Grayscale*. Sedangkan ekstraksi fitur bentuk menggunakan *area* dan *parameter*. Dan untuk ekstraksi fitur tekstur, metode yang digunakan adalah GLCM (*Gray Level Co-occurrence Matrix*) dengan menggunakan fitur-fitur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Setelah semua nilai terkumpul dan dimasukan dalam data

latih, selanjutnya citra baru dapat diklasifikasikan menggunakan metode *K-Nearest Neighbour*.

#### **3.2.4 Modeling**

Penulis membuat sebuah model simulasi yang melibatkan metode ekstraksi fitur warna, bentuk, dan tekstur secara manual. Untuk ekstraksi fitur warna, model ini menggunakan *color moment* yang mencakup *mean* RGB, HSV, standar deviasi, dan variasi pada Grayscale. Sedangkan untuk ekstraksi fitur bentuk, model ini menggunakan area dan perimeter. Selain itu, dalam proses ekstraksi fitur tekstur, model ini menggunakan GLCM (Gray Level Co-occurrence Matrix) dengan jarak 1 dan sudut 0, serta fitur-fitur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Dalam penelitian ini, algoritma klasifikasi yang digunakan adalah *K-Nearest Neighbour*.

#### **3.2.5 Simulation**

Pada tahap ini, penulis menjalankan proses simulasi berdasarkan *conceptual model* menggunakan teknik pengolahan citra yang telah dibuat pada tahap sebelumnya.

#### **3.2.6 Verification and Validation**

Pada tahap *verification and validation* dilakukan pengulangan kembali mengenai kegiatan yang dilakukan pada tahapan-tahapan sebelumnya apakah sudah sesuai dengan yang diharapkan dan saling berhubungan.

#### **3.2.7 Experiment**

Desain eksperimen menjelaskan bagaimana data input didapatkan untuk pelaksanaan percobaan pada model. Dalam hal ini sampel jenis kopra edible, regular, dan reject. Eksperimen yang digunakan dalam penelitian ini yaitu 12 skenario dengan mempertimbangkan kombinasi fitur, antara lain warna dan bentuk, warna dan tekstur, bentuk dan tekstur, serta warna, bentuk, dan tekstur. Setiap

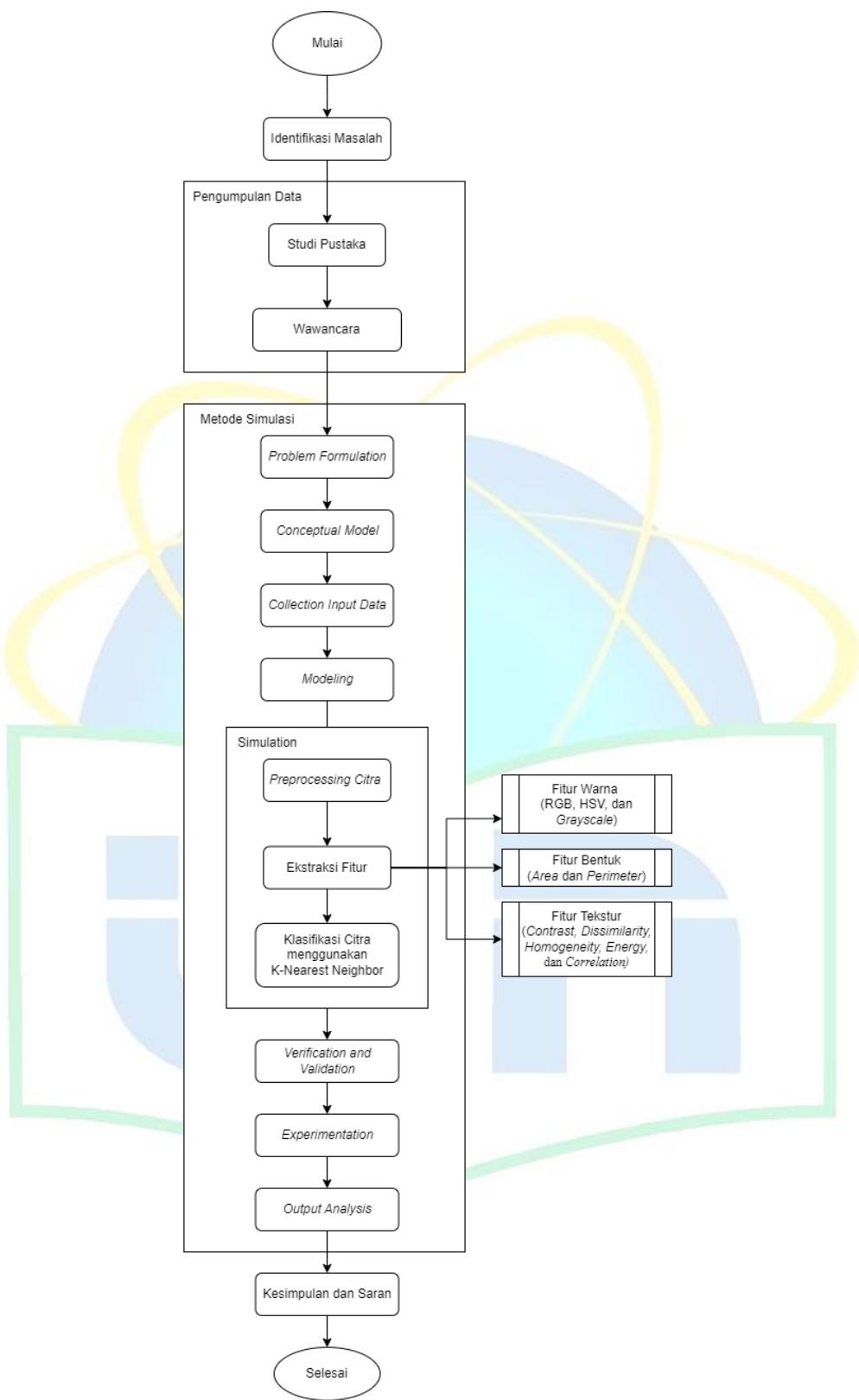
kombinasi fitur tersebut diuji pada tiga pembagian sampel, yaitu 50:50, 70:30, dan 90:10.

### **3.2.8 *Output Analysis***

Tahap ini merupakan tahap akhir dari metode simulasi. Pada tahapan ini, penulis menganalisis *output* dari hasil percobaan yang dilakukan. Hasil analisis yang diuraikan berupa hasil akurasi yang diperoleh dalam penelitian sesuai dengan permasalahan utama.

## **3.3 Alur Penelitian**





Gambar 3. 1 Alur Penelitian



## **BAB IV**

### **IMPLEMENTASI**

Pada bab ini, penulis akan menjelaskan seluruh proses implementasi dari mulai pemrosesan dataset hingga evaluasi kinerja model, serta identifikasi perangkat yang digunakan dalam eksperimen penulisan ini. Selain itu, penulis juga menyajikan penjelasan implementasi dari *code* yang digunakan, termasuk tangkapan layar dari baris kode yang dieksekusi, serta perhitungan manual yang dilakukan.

#### **4.1 Identifikasi Kebutuhan Sistem**

Kebutuhan sistem yang digunakan dalam implementasi penelitian ini dibagi menjadi dua yaitu spesifikasi perangkat keras yang dijelaskan pada Tabel 4.1 dan spesifikasi perangkat lunak yang dijelaskan pada Tabel 4.2

Tabel 4. 1 Spesifikasi Perangkat Keras

No	Nama	Spesifikasi
1	Jenis Laptop	Lenovo ideapad 330
2	Processor	Intel Core i3-6006U
3	RAM	4 GB
4	VGA	Intel UHD Graphics 600
5	System Type	64-bit Operating System
6	OS Edition	Windows 10 Home Single Language

Tabel 4. 2 Spesifikasi Perangkat Lunak

No	Nama	Spesifikasi
1	Bahasa Pemograman	Python 3.11.3

2	Library	<ul style="list-style-type: none"> <li>- Pandas</li> <li>- Numpy</li> <li>- OpenCv</li> <li>- Csv</li> <li>- Graycomatrix</li> <li>- Graycoprops</li> <li>- Image</li> <li>- Os</li> <li>- Remove</li> </ul>
3	Tools	Visual Studio Code

#### 4.2 Pengumpulan Data

Sumber data yang digunakan dalam penelitian ini berasal dari pengambilan data langsung di pabrik industri kelapa yang terletak di Kabupaten Indragili Hilir, Provinsi Riau. Dataset terdiri dari tiga jenis kopra non telungkup, yaitu edible, regular, dan reject. Terdapat 613 data citra untuk masing-masing jenis kopra, sehingga total seluruhnya berjumlah 1.839 data citra yang akan digunakan.

Data tersebut termasuk dalam kategori data primer karena dikumpulkan langsung dari sumbernya. Data primer ini diperoleh melalui kerjasama dengan para petani di Indragili Hilir, Provinsi Riau, yang bertindak sebagai pengumpul kelapa. Dengan demikian, data yang digunakan dalam penelitian ini memiliki keakuratan dan keaslian yang tinggi karena bersumber langsung dari lokasi pengambilan kelapa tersebut..

#### 4.3 Problem Formulation

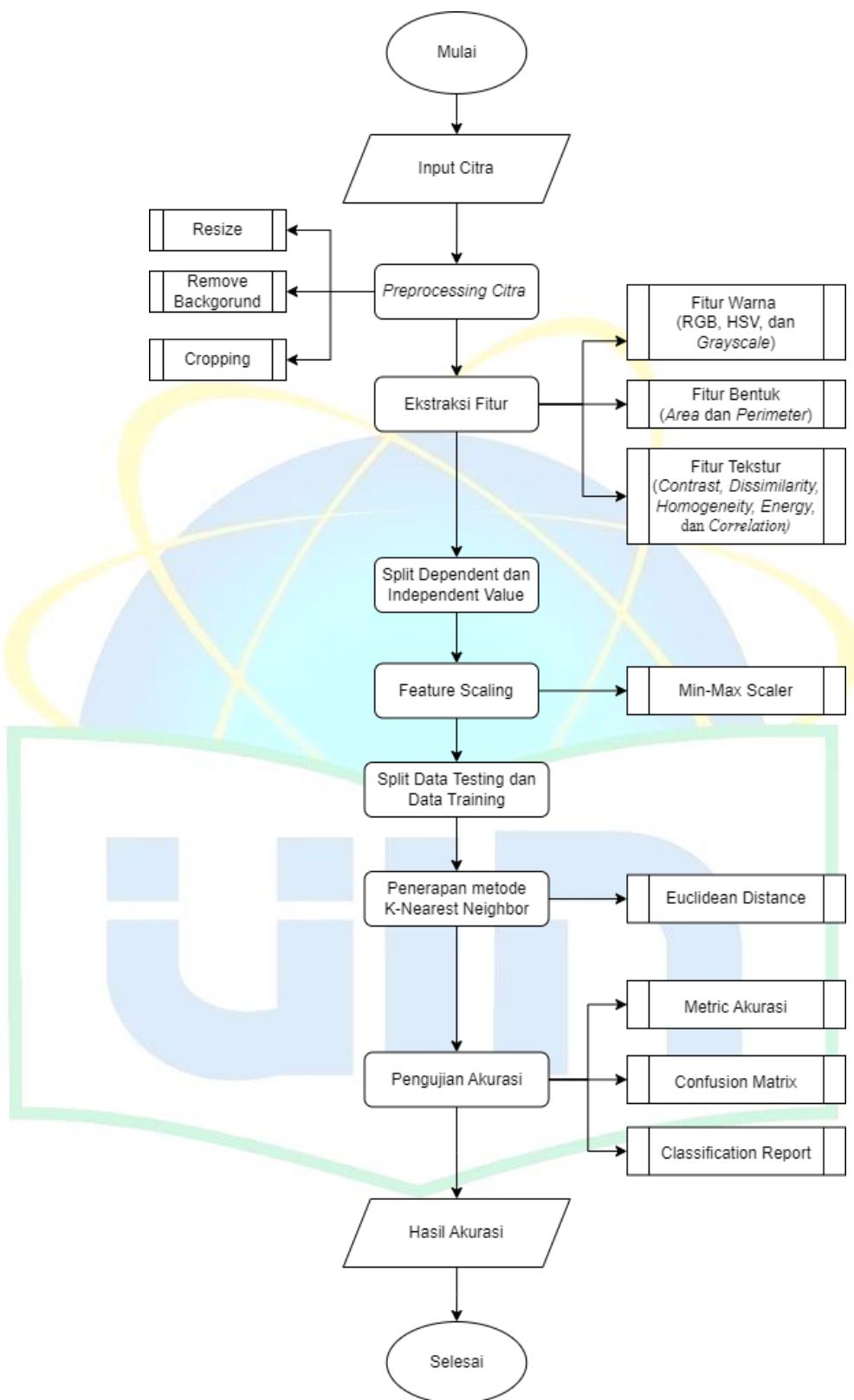
Dalam merumuskan permasalahan pada penelitian ini penulis melakukan studi literatur dan wawancara kepada pihak terkait. Berdasarkan hasil wawancara yang dilakukan, pengklasifikasian jenis kopra umumnya masih dilakukan oleh petani secara manual. Tentunya proses ini membutuhkan

tenaga, waktu, dan biaya yang cukup tinggi. Oleh karena itu, dibutuhkan pengembangan metode yang dapat mengklasifikasikan jenis kopra secara otomatis dan akurat untuk mendukung proses ini. Klasifikasi yang akurat dan efisien dapat membantu mengelola pasokan kopra dengan standar kualitas yang berlaku, serta memenuhi kebutuhan beragam pengguna kopra di berbagai sektor industri dengan cara yang lebih tepat. Penelitian terkait telah banyak dilakukan, namun pada penelitian ini akan fokus pada klasifikasi jenis kopra berdasarkan ekstraksi fitur menggunakan metode *K-Nearest Neighbour*.

#### **4.4 Conceptual Model**

Pada tahapan ini, penulis membuat konsep model yang berisi langkah-langkah yang dilakukan dalam penelitian untuk mengklasifikasikan jenis kopra berdasarkan ekstraksi fiturnya. Tujuan dari tahapan ini, yaitu untuk menjawab pertanyaan permasalahan yang telah didefinisikan sebelumnya pada penulisan ini. Berikut merupakan langkah-langkahnya:





Gambar 4. 1 Pra-Proses Simulasi Sistem

## 1) *Preprocessing* Citra

Tahap ini dilakukan untuk memproses citra sebelum digunakan untuk pemodelan. Pada penelitian ini pemrosesan citra dilakukan dengan *me-resize* ukuran citra, menghapus *background*, dan *cropping*.

### a. *Resize*

Langkah pertama yang dilakukan untuk *resize* ukuran citra adalah membuat direktori yang dibutuhkan untuk menampung hasil proses *resize* dengan menggunakan fungsi **os.makedirs()**. Menggunakan **exist\_ok=True** untuk memastikan bahwa direktori yang akan dibuat jika belum ada dan jika sudah ada, tidak akan menimbulkan masalah. Lalu melakukan iterasi pada setiap folder di dalam direktori yang dituju dengan fungsi **os.listdir()** dan **os.path.join()** untuk membentuk path lengkap ke setiap folder. Kemudian, melakukan *resize* citra menjadi  $800 \times 600$  menggunakan fungsi **img.resize(800×600)**. Setelah itu, menyimpan hasil *resize* pada folder yang sudah dibuat sebelumnya menggunakan **img.save()**.

### b. *Remove Background*

Langkah pertama yang dilakukan untuk *remove background* citra adalah membuat direktori yang dibutuhkan untuk menampung hasil proses *resize* dengan menggunakan fungsi **os.makedirs()**. Menggunakan **exist\_ok=True** untuk memastikan bahwa direktori yang akan dibuat jika belum ada dan jika sudah ada, tidak akan menimbulkan masalah. Lalu melakukan iterasi pada setiap folder di dalam direktori yang dituju dengan fungsi **os.listdir()** dan **os.path.join()** untuk membentuk path lengkap ke setiap folder. Kemudian, melakukan *remove background* dengan fungsi **remove()**. Setelah latar belakang dihapus, nama file diubah dengan mengganti ekstensi .jpg menjadi .png. Setelah itu, menyimpan hasil *remove background* pada folder yang sudah dibuat sebelumnya menggunakan **result.save()**.

### c. *Cropping*

Langkah pertama yang dilakukan untuk *remove background* citra adalah membuat direktori yang dibutuhkan untuk menampung hasil proses *resize* dengan menggunakan fungsi **os.makedirs()**. Menggunakan **exist\_ok=True** untuk memastikan bahwa direktori yang akan dibuat jika belum ada dan jika sudah ada, tidak akan menimbulkan masalah. Lalu melakukan iterasi pada setiap folder di dalam direktori yang dituju dengan fungsi **os.listdir()** dan **os.path.join()** untuk membentuk path lengkap ke setiap folder. Kemudian, melakukan *cropping* dengan fungsi **img.crop(img.getbbox())**. *Crop* dilakukan dengan memanfaatkan **img.getbbox** untuk mendapatkan kotak pembatas yang mencakup seluruh isi gambar. Gambar yang telah di *crop* disimpan menggunakan **img.save()**.

## 2) Ekstraksi Fitur

Tahap ini dilakukan untuk merubah data citra yang dikonversi menjadi data numerik, dimana menghasilkan dua belas array nilai yang diberi label *meanR*, *meanG*, *meanB*, *meanH*, *meanS*, *meanV*, *meanGrayscale*, Standar-Deviasi, Variasi, Luas, *Parimeter*, *Contrast*, *Dissimililarity*, *Homogeneity*, *Energy*, *Correlation*, serta label jenis kopra. Hasil masing-masing konversi data tersebut akan digabungkan kemudian di *export* dalam *format* file **.csv (Comma Separated Values)** dan **.xlsx (Extensible Stylesheet Language)** yang di simpan pada folder yang sudah tersedia. Dalam proses ekstraksi fitur ini dibagi menjadi tiga bagian, yaitu warna, bentuk, dan tekstur. *Library* yang digunakan untuk mengekstraksi fitur dari citra yaitu *library* pandas, NumPy, OpenCV, os, csv, graycomatrix dan graycoprops.

- a. Ekstraksi fitur warna yang digunakan pada penelitian ini yaitu RGB, HSV, dan grayscale.
  - i. Ekstraksi ciri warna RGB

Langkah pertama untuk melakukan ekstraksi ciri warna RGB pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari

OpenCV) menjadi citra RGB. Fungsi yang digunakan untuk mengubah warna citra dari BGR menjadi RGB yaitu **cvtColor** dari *library* OpenCV. Selanjutnya memisahkan komponen warna R (*Red*), G (*Green*), dan B (*Blue*) dari citra menggunakan fungsi **split** dari *library* OpenCV. Kemudian menghitung rata-rata nilai piksel untuk masing-masing komponen RGB menggunakan fungsi **mean** dari *library* numpy. Dan terakhir menambahkan nilai rata-rata pada setiap komponen ke dalam suatu daftar dengan fungsi **append**. Nilai *mean* RGB ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi.

#### ii. Ekstraksi ciri warna HSV

Langkah pertama untuk melakukan ekstraksi ciri warna HSV pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari OpenCV) menjadi citra HSV. Fungsi yang digunakan untuk mengubah warna citra dari BGR menjadi HSV yaitu **cvtColor** dari *library* OpenCV. Selanjutnya memisahkan komponen warna H (Hue), S (Saturation), dan V (Value) dengan mengambil komponennya dari citra yang sudah di konversi. Kemudian menghitung rata-rata nilai piksel untuk masing-masing komponen HSV menggunakan fungsi **mean** dari *library* numpy. Dan terakhir menambahkan nilai rata-rata pada setiap komponen ke dalam suatu daftar dengan fungsi **append**. Nilai *mean* HSV ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi.

#### iii. Ekstraksi ciri warna *Grayscale*

Langkah pertama untuk melakukan ekstraksi ciri warna *Grayscale* pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari OpenCV) menjadi citra *Grayscale*. Fungsi yang digunakan untuk mengubah warna citra dari BGR menjadi *Grayscale* yaitu **cvtColor** dari *library* OpenCV. Selanjutnya menghitung rata-rata dan standar deviasi nilai piksel dalam gambar skala abu-abu menggunakan fungsi **mean** dan **std** dari *library*

- numpy. Dan terakhir menambahkan nilai rata-rata dan standar deviasi dalam suatu daftar dengan fungsi **append**. Nilai *mean* dan standar deviasi *Grayscale* ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi.
- b. Ekstraksi fitur bentuk yang digunakan pada penelitian ini yaitu luas dan parameter.
- Langkah pertama yang dilakukan adalah melakukan *thresholding* pada gambar *grayscale* dengan nilai **threshold** 127. Hasil *thresholding* menghasilkan citra biner di mana piksel di atas *threshold* bernilai maksimum (255) dan piksel di bawah *threshold* bernilai minimum (0). Selanjutnya, dilakukan pencarian kontur dalam gambar *threshold* menggunakan fungsi **findContours** dengan Parameter **RETR\_EXTERNAL** yang digunakan untuk mengambil hanya kontur eksternal, dan **CHAIN\_APPROX\_SIMPLE** untuk menyimpan titik-titik kontur. Kontur-kontur yang ditemukan disimpan dalam variabel *contours*. Jika terdapat kontur yang ditemukan, selanjutnya mencari kontur terbesar berdasarkan luas menggunakan fungsi **max** dan **contourArea**. Luas dan panjang perimeter kontur terbesar dihitung menggunakan fungsi **contourArea** dan **arcLength**. Hasil perhitungan luas dan panjang perimeter tersebut disimpan dalam list luas dan perimeter. Jika tidak ada kontur yang ditemukan, maka list luas dan perimeter akan berisi nilai 0.
- c. Ekstraksi fitur tekstur yang digunakan pada penelitian ini yaitu GLCM (*Gray Level Co-occurrence Matrix*)

Langkah pertama yang dilakukan adalah membentuk GLCM dari gambar *grayscale* menggunakan fungsi **graycomatrix**. Selanjutnya menggunakan Parameter [1] dan [0] untuk menunjukkan jarak piksel horizontal dan vertikal dalam membangun matriks GLCM serta Parameter 256 untuk menunjukkan jumlah tingkat keabuan yang digunakan dalam gambar *grayscale*. Kemudian melakukan ekstraksi fitur tekstur menggunakan fungsi **graycoprops**. Fungsi ini menerima

matriks GLCM dan jenis fitur yang ingin diekstraksi. Terdapat beberapa jenis fitur yang diekstraksi, yaitu *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Nilai fitur tersebut diambil dari hasil ekstraksi dengan indeks [0, 0].

### 3) *Split Dependent* dan *Independent value*

Pada tahap ini dilakukan pemisahan data yaitu *dependent value* dan *independent value*. Dalam penelitian ini, *dependent value* berisi informasi tentang kategori atau kelas dari jenis kopra yaitu, kopra edible, regular, dan reject. Sedangkan *independent value* digunakan sebagai inputan atau atribut yang digunakan untuk melatih model. *Independent value* dalam penelitian ini meliputi ekstraksi fitur yaitu *meanR*, *meanG*, *meanB*, *meanH*, *means*, *meanV*, *meanGrayscale*, Standar-Deviasi, Variasi, Luas, Parimeter, *Contrast*, *Dissimililarity*, *Homogenity*, *Energy*, *Correlation*. Untuk mendapatkan nilai *dependent* dan *independent value* pada dataset, penelitian ini menggunakan fungsi **iloc** untuk mengindeks baris dan kolom berdasarkan posisi numerik. Selanjutnya menggunakan metode **.values** untuk mengkonversi data menjadi array Numpy.

### 4) *Feature Scaling*

Setelah mendapatkan nilai ekstraksi ciri pada masing-masing citra pada tahap sebelumnya, pada tahap ini akan dilakukan normalisasi data untuk mengubah skala atau rentang nilai dari fitur-fitur yang ada pada dataset. Pada penelitian ini menggunakan kelas **MinMaxScaler** dari modul **preprocessing** dalam *library Scikit-learn* untuk melakukan normalisasi.

### 5) *Train-Test Split*

Pada tahap ini dilakukan pembagian dataset menjadi dua bagian yaitu data *training* dan data *testing*. Adapun pembagian antara data *training* dan data *testing* menjadi 3 skenario dengan perbandingan 50:50, 70:30, dan 90:10. *Random state* digunakan agar dataset tidak berubah-ubah, sehingga diberikan nilai ketetapannya. Untuk membagi dataset menjadi dua bagian,

penelitian ini menggunakan fungsi `train_test_split` dari model **model\_selection** dalam *library Scikit-learn*.

#### 6) Penerapan Metode K-Nearest Neighbour

Pada tahap ini dilakukan klasifikasi citra berdasarkan hasil ekstraksi yang didapat pada tahap sebelumnya. Algoritma *K-Nearest Neighbor* digunakan untuk melakukan klasifikasi citra dengan membandingkan data *testing* dan data *training* menggunakan rumus *Euclidean distance* dengan nilai  $k = 3$  dari metode *K-Nearest Neighbor*. Pada penerapan metode *K-Nearest Neighbor* menggunakan *library Scikit-Learn*.

#### 7) Pengujian

Pada tahap ini dilakukan pengujian perfomansi untuk mengevaluasi hasil klasifikasi dari metode yang diusulkan. Untuk mengetahui performa dari algoritma *K-Nearest Neighbor*, penelitian ini menggunakan metrik akurasi, *confusion matrix*, dan *classification report*. Beberapa fungsi yang digunakan dalam pengujian ini adalah `accuracy_score`, `classification_report`, dan `confusion_matrix` dari *library Scikit-learn*.

### 4.5 Collection of Input Data

Dalam penelitian ini data sampel citra kopra yang penulis gunakan sebanyak 1.839 buah yang terdiri dari 613 buah citra kopra edible, 613 buah citra kopra regular, dan 613 buah citra kopra reject. Berikut merupakan contoh sampel citra yang penulis gunakan yang dijadikan sebagai data input antara lain:



Gambar 4. 2 Contoh Sampel Kopra Edible



Gambar 4. 3 Contoh Sampel Kopra Reguler



Gambar 4. 4 Contoh Sampel Kopra Reject

## 4.6 Modeling

### 4.6.1 Perhitungan Manual Ekstraksi Fitur Warna

Berikut ini adalah tahapan perhitungan ekstraksi fitur warna RGB, HSV, dan Grayscale:

#### 1) Ekstraksi Fitur Warna RGB

Langkah pertama untuk melakukan ekstraksi ciri warna RGB pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari OpenCV) menjadi citra RGB. Selanjutnya, melakukan pengambilan nilai *mean* pada setiap citra. Nilai *mean* RGB ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi. Misalkan terdapat citra RGB  $3 \times 3$  yang dapat digunakan untuk mencari nilai ciri warna dari suatu citra.

R : 80 G : 81 B : 76	R : 85 G : 86 B : 80	R : 92 G : 92 B : 85
R : 97 G : 97 B : 89	R : 103 G : 102 B : 92	R : 106 G : 104 B : 94
R : 107 G : 107	R : 110 G : 108	R : 111 G : 109

B : 94	B : 97	B : 96
--------	--------	--------

Gambar 4. 5 Citra RGB 3x3

Berikut merupakan proses perhitungan untuk mencari nilai *mean* berdasarkan komponen nilai citra warna RGB dengan persamaan (2.6):

$$\begin{aligned}
 MeanR &= \frac{1}{9}(80 + 85 + 92 + 97 + 103 + 106 + 107 + 110 \\
 &\quad + 111) \\
 &= 99 \\
 MeanG &= \frac{1}{9}(81 + 86 + 92 + 97 + 102 + 104 + 107 + 108 \\
 &\quad + 109) \\
 &= 98,4444 \\
 MeanB &= \frac{1}{9}(76 + 80 + 85 + 89 + 92 + 94 + 94 + 97 + 96) \\
 &= 89,2222
 \end{aligned}$$

Berdasarkan hasil perhitungan untuk mencari nilai *mean* dari momen warna RGB diperoleh tiga nilai ciri warna yaitu: *MeanR* = 99, *MeanG* = 98,4444, dan *MeanB* = 89,222.

## 2) Ekstraksi Fitur Warna RGB

Langkah pertama untuk melakukan ekstraksi ciri warna HSV pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari OpenCV) menjadi citra HSV. Selanjutnya, melakukan pengambilan nilai *mean* pada setiap citra. Nilai *mean* HSV ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi. Misalkan terdapat citra HSV 3x3 yang dapat digunakan untuk mencari nilai ciri warna dari suatu citra.

H : 36 S : 16 V : 81	H : 35 S : 18 V : 86	H : 30 S : 19 V : 92
H : 30 S : 21 V : 97	H : 27 S : 27 V : 103	H : 25 S : 29 V : 106

H : 30	H : 25	H : 26
S : 31	S : 30	S : 34
V : 107	V : 110	V : 111

Gambar 4. 6 Citra HSV 3x3

Berikut merupakan proses perhitungan untuk mencari nilai *mean* berdasarkan komponen nilai citra warna HSV dengan persamaan (2.6):

$$\text{MeanH} = \frac{1}{9}(36 + 35 + 30 + 30 + 27 + 25 + 30 + 25 + 26) \\ = 29,3333$$

$$\text{MeanS} = \frac{1}{9}(16 + 18 + 19 + 21 + 27 + 29 + 31 + 30 + 34) \\ = 25$$

$$\text{MeanV} = \frac{1}{9}(81 + 86 + 92 + 97 + 103 + 106 + 107 + 110 \\ + 111) \\ = 99,2222$$

Berdasarkan hasil perhitungan untuk mencari nilai *mean* dari momen warna HSV diperoleh tiga nilai ciri warna yaitu:  $\text{MeanH} = 29,3333$ ,  $\text{MeanS} = 25$ , dan  $\text{MeanV} = 99,222$ .

### 3) Ekstraksi ciri warna *Grayscale*

Langkah pertama untuk melakukan ekstraksi ciri warna *Grayscale* pada suatu citra yaitu dengan mengkonversinya dari BGR (default dari OpenCV) menjadi citra *Grayscale*. Berikut merupakan contoh perhitungan untuk mendapatkan citra berwarna *grayscale* menggunakan persamaan (2.5):

$$\text{GrayScale}_{11} = (0,3 \times 80) + (0,59 \times 81) + (0,11 \times 76)$$

$$\text{GrayScale}_{11} = 80,15$$

Selanjutnya, melakukan pengambilan nilai *mean* pada setiap citra. Nilai *mean* *Grayscale* ini akan digunakan sebagai salah satu ciri warna dari jenis kopra yang di ekstraksi. Misalkan terdapat citra

Greyscale  $3 \times 3$  yang dapat digunakan untuk mencari nilai ciri warna dari suatu citra.

80	85	91
96	101	103
106	107	108

Gambar 4. 7 Citra *Grayscale*  $3 \times 3$

Berikut merupakan proses perhitungan untuk mencari nilai *mean* berdasarkan komponen nilai citra warna *Grayscale* dengan persamaan (2.6):

$$\begin{aligned} MeanGray &= \frac{1}{9}(80 + 85 + 91 + 96 + 101 + 103 + 106 + 107 \\ &\quad + 108) \\ &= 97,4444 \end{aligned}$$

Kemudian, melakukan perhitungan standar deviasi dari nilai *Grayscale* dengan persamaan (2.7):

*Standar – Deviasi*

$$\begin{aligned} & (80 - 97,4444)^2 + (85 - 97,4444)^2 \\ & + (91 - 97,4444)^2 + (96 - 97,4444)^2 \\ & + (101 - 97,4444)^2 + (103 - 97,4444)^2 \\ & + (106 - 97,4444)^2 + (107 - 97,4444)^2 \\ & + (108 - 97,4444)^2 \\ & = \sqrt{\frac{(80 - 97,4444)^2 + (85 - 97,4444)^2 + (91 - 97,4444)^2 + (96 - 97,4444)^2 + (101 - 97,4444)^2 + (103 - 97,4444)^2 + (106 - 97,4444)^2 + (107 - 97,4444)^2 + (108 - 97,4444)^2}{9 - 1}} \\ & = 58,94838 \end{aligned}$$

#### 4.6.2 Perhitungan Manual Ekstraksi Fitur Bentuk

Ekstraksi bentuk yang digunakan pada penilitian ini yaitu *area* dan *perimeter*. Hal pertama yang dilakukan adalah mencari kontur dengan menghubungkan piksel yang memiliki nilai yang berbeda dari piksel sekitarnya. Setiap segmen garis yang membentuk kontur akan memberikan kontribusi terhadap *area* dan *perimeter* kontur. Penulis akan menggunakan metode shoelace formula untuk menghitung *area* dan rumus jarak *Euclidean* untuk menghitung perimeter antara titik-titik yang membentuk garis segmen. Berikut merupakan tahap perhitungan manualnya:

### 1) *Area* (luas)

Dalam hal ini menggunakan contoh citra grayscale dengan matriks piksel  $3 \times 3$ .

80	85	91
96	101	103
106	107	108

Area akan dihitung menggunakan rumus berikut:

$$\begin{aligned} \text{Area} = & (x_1 \times (y_2 - y_1)) + (x_2 \times (y_3 - y_2)) \\ & + (x_3 \times (y_1 - y_3)) \end{aligned}$$

- a. Mulai dari titik pertama pada kontur, yaitu  $(0, 0)$ ,  $x_1 = 0$ ,  $y_1 = 0$

- b. Titik berikutnya pada kontur adalah  $(0, 2)$ ,  $x_2 = 0$ ,  $y_2 = 2$

$$\text{Area} = (0 \times (2 - 0)) = 0$$

- c. Titik selanjutnya pada kontur adalah  $(1, 2)$ ,  $x_3 = 1$ ,  $y_3 = 2$

$$\text{Area} = (0 \times (2 - 2)) = 0$$

- d. Titik berikutnya pada kontur adalah  $(2, 1)$ ,  $x_4 = 2$ ,  $y_4 = 1$

$$\text{Area} = (2 \times (1 - 2)) = -2$$

- e. Terakhir, tutup kontur dengan menghubungkan titik terakhir dengan titik awal, yaitu  $(2, 1)$  ke  $(0, 0)$ .

$$\text{Area} = (0 \times (0 - 1)) = 0$$

- f. Setelah semua kontribusi luas dijumlahkan, ambil nilai absolut dan bagi dengan 2 untuk mendapatkan luas kontur akhir:

$$\text{Area} = \left| \frac{0 + 0 + (-2)}{2} \right| = \left| -\frac{2}{2} \right| = 1$$

Jadi, luas kontur dari gambar tersebut adalah 1

### 2) *Perimeter* (keliling)

Dalam hal ini menggunakan contoh citra grayscale dengan matriks piksel  $3 \times 3$ .

80	85	91
96	101	103

- $$\begin{array}{c|c|c} & 106 & 107 & 108 \end{array}$$
- a. Mulai dari titik pertama pada kontur (0,0) dengan nilai piksel 80 dan titik berikutnya pada kontur (0,2) dengan nilai piksel 91.

$$Perimeter = \sqrt{(0 - 0)^2 + (2 - 0)^2} = 2$$

- b. Titik berikutnya pada kontur (2,2) dengan nilai piksel 108.

$$Perimeter = \sqrt{(2 - 2)^2 + (2 - 2)^2} = 0$$

- c. Titik berikutnya pada kontur (2,0) dengan nilai piksel 106.

$$Perimeter = \sqrt{(2 - 2)^2 + (0 - 0)^2} = 0$$

- d. Terakhir, tutup kontur dengan menghubungkan titik terakhir dengan titik awal, yaitu (2,0) ke (0,0).

$$Perimeter = \sqrt{(2 - 0)^2 + (0 - 0)^2} = 2$$

- e. Setelah mengetahui semua jumlah kontribusi jarak, langkah selanjutnya menambahkan semua kontribusi jaraknya:

$$Perimeter = 2 + 0 + 0 + 2 = 4$$

Jadi, *perimeter* dari kontur gambbar dengan matriks tersebut adalah 4.

#### 4.6.3 Perhitungan Manual Ekstraksi Fitur Tekstur

Ekstraksi fitur yang digunakan pada penilitian ini yaitu metode GLCM (*Gray Level Co-occurrence Matrix*) dengan beberapa fitur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Berikut merupakan tahapan untuk perhitungan manualnya:

- 1) Dalam hal ini, menggunakan contoh matriks  $3 \times 3$ .

4	3	1
7	6	4
6	4	3

- 2) Menentukan hubungan spasial antar piksel dengan inisiasi nilai jarak dan sudut. Dalam hal ini menggunakan jarak 1 dan sudut  $0^\circ$ .
- 3) Membuat matriks GLCM untuk setiap pikselnya dengan menghitung seberapa sering pasangan piksel muncul pada jarak dan arah yang ditentukan.

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	2	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	1	0

- 4) Normalisasi matriks untuk mendapatkan distribusi probabilitasnya. Hal ini dilakukan dengan membagi setiap elemen matriks dengan jumlah total pasangan piksel yang dihitung.

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0,167	0	0	0	0	0	0

4	0	0	0	0,333	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0,333	0	0	0
7	0	0	0	0	0	0	0,167	0

- 5) Menghitung nilai ekstraksi fitur ekstraksi GLCM dari hasil normalisasi.

- a. *Contrast*, berikut merupakan perhitungan *contrast* menggunakan persamaan (2.8):

$$Contrast_{(3,1)} = (3 - 1)^2 \times 0,167 = 0,668$$

$$Contrast_{(4,3)} = (4 - 3)^2 \times 0,333 = 0,333$$

$$Contrast_{(6,4)} = (6 - 4)^2 \times 0,333 = 1,332$$

$$Contrast_{(7,6)} = (7 - 6)^2 \times 0,167 = 0,167$$

$$Contrast = 0,668 + 0,333 + 1,332 + 0,167 = 2,333$$

- b. *Dissimilarity*, berikut merupakan perhitungan *dissimilarity* menggunakan persamaan (2.9):

$$Dissimilarity_{(3,1)} = |3 - 1| \times 0,167 = 0,333$$

$$Dissimilarity_{(4,3)} = |4 - 3| \times 0,333 = 0,333$$

$$Dissimilarity_{(6,4)} = |6 - 4| \times 0,333 = 0,667$$

$$Dissimilarity_{(7,6)} = |7 - 6| \times 0,167 = 0,167$$

$$Dissimilarity = 0,333 + 0,333 + 0,667 + 0,167 = 1,5$$

- c. *Homogeneity*, berikut merupakan perhitungan *homogeneity* menggunakan persamaan (2.10):

$$Homogeneity_{(3,1)} = \frac{0,167}{1 + (3 - 1)^2} = 0,033$$

$$Homogeneity_{(4,3)} = \frac{0,333}{1 + (4 - 3)^2} = 0,167$$

$$Homogeneity_{(6,4)} = \frac{0,333}{1 + (6 - 4)^2} = 0,067$$

$$Homogeneity_{(7,6)} = \frac{0,167}{1 + (7 - 6)^2} = 0,084$$

$$Homogeneity = 0,033 + 0,167 + 0,067 + 0,084 = 0,267$$

- d. *Energy*, berikut merupakan perhitungan *energy* menggunakan persamaan (2.11):

$$\begin{aligned} Energy &= 0,167^2 + 0,333^2 + 0,333^2 + 0,167^2 \\ &= 0,278 \end{aligned}$$

- e. *Correlation*, berikut merupakan perhitungan *homogeneity* menggunakan persamaan (2.12):

$$\begin{aligned} \mu_i &= (3 \times 0,167) + (4 \times 0,333) + (6 \times 0,333) \\ &\quad + (7 \times 0,167) \end{aligned}$$

$$\mu_i = 5$$

$$\begin{aligned} \mu_j &= (1 \times 0,167) + (3 \times 0,333) + (4 \times 0,333) \\ &\quad + (6 \times 0,167) \end{aligned}$$

$$\mu_j = 3,5$$

$$\sigma i_{(3,1)} = (3 - 5)^2 \times 0,167 = 0,668$$

$$\sigma i_{(4,3)} = (4 - 5)^2 \times 0,333 = 0,333$$

$$\sigma i_{(6,4)} = (6 - 5)^2 \times 0,333 = 0,333$$

$$\sigma i_{(7,6)} = (7 - 5)^2 \times 0,167 = 0,668$$

$$\sigma i = 0,668 + 0,333 + 0,333 + 0,668 = 2,002$$

$$\sigma j_{(3,1)} = (1 - 3,5)^2 \times 0,167 = 1,044$$

$$\sigma j_{(4,3)} = (3 - 5)^2 \times 0,333 = 0,083$$

$$\sigma j_{(6,4)} = (4 - 5)^2 \times 0,333 = 0,083$$

$$\sigma j_{(7,6)} = (6 - 5)^2 \times 0,167 = 1,044$$

$$\sigma_j = 0,668 + 0,333 + 0,333 + 0,668 = 2,254$$

$$Correlation_{(3,1)} = \frac{(3 - 5) \times (1 - 3,5) \times 0,167}{0,668 \times 1,044} \\ = 1,198$$

$$Correlation_{(4,3)} = \frac{(4 - 5) \times (3 - 3,5) \times 0,333}{0,333 \times 0,083} \\ = 6,006$$

$$Correlation_{(6,4)} = \frac{(6 - 5) \times (4 - 3,5) \times 0,333}{0,333 \times 0,083} \\ = 6,006$$

$$Correlation_{(7,6)} = \frac{(3 - 5) \times (1 - 3,5) \times 0,167}{0,668 \times 1,044} \\ = 1,198$$

$$Correlation = 1,198 + 6,006 + 6,006 + 1,198 \\ = 14,407$$

#### 4.6.4 Pengklasifikasian dengan K-Nearest Neighbour

Untuk melakukan klasifikasi citra kopra, penulis menggunakan metode *K-Nearest Neighbour*. Klasifikasi ini menentukan jenis kopra dari data *testing* yang sudah tersedia. Berikut merupakan contoh langkah-langkah untuk mengklasifikasikan jenis kopra berdasarkan ekstraksi fitur yang telah diperoleh pada tahap sebelumnya.

Diketahui data *training* dan data *testing* ekstraksi fitur pada citra kopra yang terdiri dari ekstraksi warna, bentuk, dan tekstur.

Tabel 4. 3 Contoh data *training* citra kopra

<b>Mean -R</b>	<b>Mean -G</b>	<b>Mean -B</b>	<b>Mean -H</b>	<b>Mean -V</b>	<b>Mean -S</b>	<b>Mean -Gray</b>	<b>Standar -Deviasi</b>	<b>Varia si</b>	<b>Luas</b>	<b>Peri meter</b>	<b>Contr ast</b>	<b>Dissim ilarity</b>	<b>Homog eneity</b>	<b>Ener gy</b>	<b>Corre lation</b>	<b>Jenis Kopra</b>
0,664	0,641	0,677	0,084	0,664	0,291	0,652	0,485	0,485	0,046	0,339	0,070	0,137	0,508	0,148	0,951	Edible
0,650	0,752	0,834	0,172	0,653	0,043	0,728	0,609	0,609	0,258	0,181	0,048	0,108	0,492	0,207	0,987	Edible
0,517	0,610	0,576	0,169	0,525	0,257	0,578	0,566	0,566	0,098	0,191	0,118	0,196	0,455	0,227	0,942	Reguler
0,652	0,765	0,759	0,205	0,525	0,208	0,729	0,646	0,646	0,167	0,359	0,155	0,229	0,370	0,154	0,942	Reguler
0,837	0,797	0,509	0,139	0,838	0,690	0,782	0,694	0,694	0,425	0,887	0,530	0,661	0,145	0,192	0,794	Reject
0,490	0,399	0,177	0,102	0,490	0,853	0,406	0,341	0,341	0,008	0,084	0,147	0,288	0,324	0,224	0,824	Reject

Tabel 4. 4 Contoh data *testing* citra kopra

<b>Mean -R</b>	<b>Mean -G</b>	<b>Mean -B</b>	<b>Mean -H</b>	<b>Mean -V</b>	<b>Mean -S</b>	<b>Mean -Gray</b>	<b>Standar -Deviasi</b>	<b>Varia si</b>	<b>Luas</b>	<b>Peri meter</b>	<b>Contr ast</b>	<b>Dissim ilarity</b>	<b>Homog eneity</b>	<b>Ener gy</b>	<b>Corre lation</b>	<b>Jenis Kopra</b>
0,696	0,769	0,864	0,143	0,697	0,079	0,755	0,555	0,555	0,274	0,186	0,065	0,102	0,530	0,110	0,969	?

- 1) Langkah pertama yang harus dilakukan yaitu menentukan nilai k. Nilai k yang digunakan pada penelitian ini adalah 3.
- 2) Selanjutnya menghitung rentang atau jarak data *training* dengan data *testing* menggunakan rumus (2.13):

$$X1 = \sqrt{(0,6636 - 0,6959)^2 + (0,6414 - 0,7686)^2 + (0,6765 - 0,8640)^2}$$

$$+ (0,0835 - 0,1432)^2 + (0,6637 - 0,6971)^2$$

$$+ (0,2913 - 0,0787)^2 + (0,6519 - 0,7552)^2$$

$$+ (0,4846 - 0,5548)^2 + (0,4846 - 0,5548)^2$$

$$+ (0,0461 - 0,2743)^2 + (0,3391 - 0,1861)^2$$

$$+ (0,0698 - 0,0646)^2 + (0,1371 - 0,1020)^2$$

$$+ (0,5084 - 0,5296)^2 + (0,1479 - 0,1098)^2$$

$$+ (0,9514 - 0,9695)^2}$$

$$X1 = \sqrt{0,2017}$$

$$X1 = 0,4491$$

$$X2 = \sqrt{(0,6497 - 0,6959)^2 + (0,7520 - 0,7686)^2 + (0,8343 - 0,8640)^2}$$

$$+ (0,1725 - 0,1432)^2 + (0,6532 - 0,0435)^2$$

$$+ (0,0435 - 0,0787)^2 + (0,7281 - 0,7552)^2$$

$$+ (0,6086 - 0,5548)^2 + (0,6086 - 0,5548)^2$$

$$+ (0,2575 - 0,2743)^2 + (0,1806 - 0,1861)^2$$

$$+ (0,0475 - 0,646)^2 + (0,1080 - 0,1020)^2$$

$$+ (0,4915 - 0,5296)^2 + (0,2068 - 0,1098)^2$$

$$+ (0,9514 - 0,9695)^2}$$

$$X2 = \sqrt{0,0256}$$

$$X2 = 0,1601$$

$$X3 = \sqrt{(0,5168 - 0,6959)^2 + (0,6100 - 0,7686)^2 + (0,5764 - 0,8640)^2 + (0,1685 - 0,1432)^2 + (0,5246 - 0,0435)^2 + (0,2569 - 0,0787)^2 + (0,5780 - 0,7552)^2 + (0,5663 - 0,5548)^2 + (0,5663 - 0,5548)^2 + (0,0978 - 0,2743)^2 + (0,1912 - 0,1861)^2 + (0,1180 - 0,646)^2 + (0,1964 - 0,1020)^2 + (0,4554 - 0,5296)^2 + (0,2269 - 0,1098)^2 + (0,9421 - 0,9695)^2}$$
$$X3 = \sqrt{0,2967}$$
$$X3 = 0,5447$$
$$X4 = \sqrt{(0,6552 - 0,6959)^2 + (0,7648 - 0,7686)^2 + (0,7586 - 0,8640)^2 + (0,2050 - 0,1432)^2 + (0,5246 - 0,0435)^2 + (0,2075 - 0,0787)^2 + (0,7295 - 0,7552)^2 + (0,6460 - 0,5548)^2 + (0,6460 - 0,5548)^2 + (0,1671 - 0,2743)^2 + (0,3585 - 0,1861)^2 + (0,1547 - 0,646)^2 + (0,2292 - 0,1020)^2 + (0,3701 - 0,5296)^2 + (0,1542 - 0,1098)^2 + (0,9421 - 0,9695)^2}$$
$$X4 = \sqrt{0,1742}$$
$$X4 = 0,4174$$

$$X5 = \sqrt{(0,8366 - 0,6959)^2 + (0,7970 - 0,7686)^2 + (0,5094 - 0,8640)^2}$$

$$+ (0,1393 - 0,1432)^2 + (0,8379 - 0,0435)^2$$

$$+ (0,6902 - 0,0787)^2 + (0,7818 - 0,7552)^2$$

$$+ (0,6936 - 0,5548)^2 + (0,6936 - 0,5548)^2$$

$$+ (0,4246 - 0,2743)^2 + (0,8871 - 0,1861)^2$$

$$+ (0,5299 - 0,646)^2 + (0,6612 - 0,1020)^2$$

$$+ (0,1453 - 0,5296)^2 + (0,1919 - 0,1098)^2$$

$$+ (0,7937 - 0,9695)^2}$$

$$X5 = \sqrt{1,8080}$$

$$X5 = 1,3446$$

$$X6 = \sqrt{(0,4903 - 0,6959)^2 + (0,3987 - 0,7686)^2 + (0,1768 - 0,8640)^2}$$

$$+ (0,1020 - 0,1432)^2 + (0,4904 - 0,0435)^2$$

$$+ (0,8527 - 0,0787)^2 + (0,4062 - 0,7552)^2$$

$$+ (0,3414 - 0,5548)^2 + (0,3414 - 0,5548)^2$$

$$+ (0,0084 - 0,2743)^2 + (0,0836 - 0,1861)^2$$

$$+ (0,1473 - 0,646)^2 + (0,2883 - 0,1020)^2$$

$$+ (0,3241 - 0,5296)^2 + (0,2245 - 0,1098)^2$$

$$+ (0,8240 - 0,9695)^2}$$

$$X6 = \sqrt{1,7069}$$

$$X6 = 1,3065$$

Tabel 4.5 Nilai hasil perhitungan jarak data *training* dengan data *testing*

<i>Mea n-R</i>	<i>Mea n-G</i>	<i>Mea n-B</i>	<i>Mea n-H</i>	<i>Mea n-V</i>	<i>Mea n-S</i>	<i>Mea n- Gray</i>	<i>Standa r- Deviasi</i>	<i>Varia si</i>	<i>Luas</i>	<i>Peri mete r</i>	<i>Contr ast</i>	<i>Dissi milari ty</i>	<i>Homo geneit y</i>	<i>Energ y</i>	<i>Corre lation</i>	<i>Jenis Kopra</i>	<i>Nilai Jarak (Eucli dean)</i>
0,664	0,641	0,677	0,084	0,664	0,291	0,652	0,485	0,485	0,046	0,339	0,070	0,137	0,508	0,148	0,951	Edible	0,449
0,650	0,752	0,834	0,172	0,653	0,043	0,728	0,609	0,609	0,258	0,181	0,048	0,108	0,492	0,207	0,987	Edible	0,160
0,517	0,610	0,576	0,169	0,525	0,257	0,578	0,566	0,566	0,098	0,191	0,118	0,196	0,455	0,227	0,942	Reguler	0,545
0,652	0,765	0,759	0,205	0,525	0,208	0,729	0,646	0,646	0,167	0,359	0,155	0,229	0,370	0,154	0,942	Reguler	0,417
0,837	0,797	0,509	0,139	0,838	0,690	0,782	0,694	0,694	0,425	0,887	0,530	0,661	0,145	0,192	0,794	Reject	1,345
0,490	0,399	0,177	0,102	0,490	0,853	0,406	0,341	0,341	0,008	0,084	0,147	0,288	0,324	0,224	0,824	Reject	1,307

- 3) Setelah mendapatkan nilai jarak antara data *training* dengan data *testing*, selanjutnya jarak tersebut diurutkan dari nilai terkecil sampai yang terbesar.

Tabel 4.5 Hasil pengurutan nilai jarak

<i>Mea n-R</i>	<i>Mea n-G</i>	<i>Mea n-B</i>	<i>Mea n-H</i>	<i>Mea n-V</i>	<i>Mea n-S</i>	<i>Mea n- Gray</i>	<i>Standa r- Deviasi</i>	<i>Varia si</i>	<i>Luas</i>	<i>Peri mete r</i>	<i>Contr ast</i>	<i>Dissi milari ty</i>	<i>Homo geneit y</i>	<i>Energ y</i>	<i>Corre lation</i>	<i>Jenis Kopra</i>	<i>Nilai Jarak (Eucli dean)</i>
0,650	0,752	0,834	0,172	0,653	0,043	0,728	0,609	0,609	0,258	0,181	0,048	0,108	0,492	0,207	0,987	Edible	0,160
0,652	0,765	0,759	0,205	0,525	0,208	0,729	0,646	0,646	0,167	0,359	0,155	0,229	0,370	0,154	0,942	Reguler	0,417
0,664	0,641	0,677	0,084	0,664	0,291	0,652	0,485	0,485	0,046	0,339	0,070	0,137	0,508	0,148	0,951	Edible	0,449
0,517	0,610	0,576	0,169	0,525	0,257	0,578	0,566	0,566	0,098	0,191	0,118	0,196	0,455	0,227	0,942	Reguler	0,545
0,490	0,399	0,177	0,102	0,490	0,853	0,406	0,341	0,341	0,008	0,084	0,147	0,288	0,324	0,224	0,824	Reject	1,307

0,837		0,797		0,509		0,139		0,838		0,690		0,782		0,694		0,694		0,425		0,887		0,530		0,661		0,145		0,192		0,794		Reject		1,345
-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	--------	--	-------

Dari hasil pengurutan tersebut, kita akan melihat tiga kelas tetangga terdekatnya (pada langkah pertama kita menentukan nilai  $k=3$ ).

Dalam kasus ini, kita menemukan bahwa terdapat 1 kelas regular dan 2 kelas edible. Berdasarkan mayoritas kelas dari tetangga terdekat, kita dapat menyimpulkan bahwa data baru tersebut masuk ke dalam kelas edible.



## 4.7 Simulation

Proses simulasi dilakukan berdasarkan model konseptual yang telah dibuat pada tahap sebelumnya. Adapun penjelasan simulasi dengan langkah-langkah beserta potongan baris kode sebagai berikut:

### 1) Preprocessing Citra

Langkah pertama yang dilakukan adalah meng-import *Library* yang dibutuhkan seperti Image, imageOps, os, dan remove.

```
from PIL import Image, ImageOps
import os
from rembg import remove
```

Pemrosesan citra yang pertama yaitu *resize* ukuran citra. Dimulai dengan membuat direktori yang dibutuhkan dan melakukan iterasi melalui setiap folder dan file di dalam direktori yang dituju. Lalu setiap gambar akan dibuka dan diubah ukurannya menjadi 800×600. Gambar yang sudah diubah ukurannya akan disimpan dalam folder Kopra\_Resize.

```
os.makedirs("Kopra_Resize", exist_ok=True)
os.makedirs("Kopra_Resize/Kopra Edible", exist_ok=True)
os.makedirs("Kopra_Resize/Kopra Reguler", exist_ok=True)
os.makedirs("Kopra_Resize/Kopra Reject", exist_ok=True)

for folder in os.listdir("KopraV2Cleansed"):
    for filename in os.listdir(os.path.join("KopraV2Cleansed",
    folder)):
        img = Image.open(os.path.join("KopraV2Cleansed",
        folder, filename))
        img = img.resize((800, 600))
        img.save(os.path.join("Kopra_Resize", folder,
        filename))
        print("Image resized to:",
os.path.join("Kopra_Resize", folder, filename))
```

Pemrosesan citra yang kedua yaitu *remove background* citra. Sama seperti proses sebelumnya, proses ini dimulai dengan membuat direktori yang dibutuhkan dan melakukan iterasi melalui setiap folder dan file di dalam direktori yang dituju. Lalu setiap gambar akan dibuka dan hapus latar

belakangnya. Setelah latar belakang gambar di hapus, nama file diubah dengan mengganti ekstensi .jpg menjadi .png. hal ini dilakukan untuk menyimpan gambar hasil tanpa latar belakang dalam format png. Gambar hasil tanpa latar belakang akan disimpan dalam folder Kopra\_Rembg.

```
os.makedirs("Kopra_Rembg", exist_ok=True)
os.makedirs("Kopra_Rembg/Kopra Edible", exist_ok=True)
os.makedirs("Kopra_Rembg/Kopra Reguler", exist_ok=True)
os.makedirs("Kopra_Rembg/Kopra Reject", exist_ok=True)

for folder in os.listdir("Kopra_Resize"):
    for filename in os.listdir(os.path.join("Kopra_Resize",
    folder)):
        img = Image.open(os.path.join("Kopra_Resize", folder,
        filename))
        result = remove(img)
        filename = filename.replace(".jpg", ".png")
        result.save(os.path.join("Kopra_Rembg", folder,
        filename))
        print("Background image removed on:",
        os.path.join("Kopra_Rembg", folder, filename))
```

Pemrosesan citra yang terakhir adalah *cropping*. Proses ini pun sama seperti sebelum-sebelumnya, dimulai dengan membuat direktori yang dibutuhkan dan melakukan iterasi melalui setiap folder dan file di dalam direktori yang dituju. Lalu setiap gambar akan dibuka dan di *crop* dengan memanfaatkan kotak pembatas yang mencakup seluruh isi gambar. Gambar yang telah di *crop* akan di simpan dalam folder Kopra\_Crop.

```
os.makedirs("Kopra_Crop", exist_ok=True)
os.makedirs("Kopra_Crop/Kopra Edible", exist_ok=True)
os.makedirs("Kopra_Crop/Kopra Reguler", exist_ok=True)
os.makedirs("Kopra_Crop/Kopra Reject", exist_ok=True)

for folder in os.listdir("Kopra_Rembg"):
    for filename in os.listdir(os.path.join("Kopra_Rembg",
    folder)):
        img = Image.open(os.path.join("Kopra_Rembg", folder,
        filename))
        img = img.crop(img.getbbox())
        img.save(os.path.join("Kopra_Crop", folder, filename))
```

```
        print("Image resized to:", os.path.join("Kopra_Crop",
folder, filename))
```

## 2) Ekstraksi fitur

Langkah pertama yang dilakukan adalah meng-*import Library* yang dibutuhkan seperti pandas, numpy, openCv, os, dan csv.

```
import pandas as pd
import numpy as np
import cv2
import os
import csv
from skimage.feature import graycomatrix, graycoprops
```

Selanjutnya, menginisialisasi variabel ‘path’ yang berisi lokasi tempat citra-citra yang akan di ekstraksi fiturnya dan mengambil daftar file dalam direktori. Kemudian membuat beberapa list kosong untuk menyimpan hasil ekstraksi fitur dari setiap citra.

```
path = 'E:/A A Skripsi/SKRIPSI/AA SKRIPSI FIX
BISMILLAH/dataset/Kopra_Crop/Kopra Edible'
data = os.listdir(path)

rataR = []
rataG = []
rataB = []
rataH = []
rataS = []
rataV = []
ratagray = []
stand = []
vari = []
luas = []
perimeter = []
contrast = []
dissimilarity = []
homogeneity = []
energy = []
correlation = []
label = []
```

Kemudian melakukan iterasi ekstraksi fitur pada setiap citra. Ekstraksi fitur pertama yang dilakukan adalah ekstraksi fitur warna. Pada

setiap iterasi, gambar dibaca terlebih dahulu, lalu diubah ke format RGB. Langkah ini bertujuan untuk memastikan bahwa urutan karnal warna yang digunakan adalah RGB. Setelah gambar diubah menjadi format RGB, kanal warna R, G, dan B diekstraksi dengan fungsi split. Dari hasil tersebut dihitung nilai rata-ratanya dan disimpan dalam variabel ‘meanR’, ‘meanG’, ‘meanB’. Nilai rata-rata tersebut kemudian ditambahkan ke dalam list ‘rataR’, ‘rataG’, dan ‘rataB’ menggunakan fungsi append. Hal ini dilakukan untuk mengumpulkan nilai rata-rata kanal warna dari setiap citra ke dalam list terpisah.

Setelah melakukan ekstraksi fitur warna RGB sebelumnya, berikut merupakan proses ekstraksi fitur warna HSV. Citra yang telah dibaca sebelumnya diubah ke format HSV. Hal ini dilakukan untuk mendapatkan komponen warna H (*Hue*), S (*Saturation*), dan V (*Value*). Setelah citra diubah menjadi format HSV, komponen warna H, S, dan V diekstraksi secara terpisah. Hal ini dilakukan dengan mengakses matriks HSV menggunakan indexing [:,:,0] untuk H, [:,:,1] untuk S, [:,:,2] untuk V. Selanjutnya menghitung nilai rata-rata dari masing-masing komponen warna H, S, dan V. Nilai rata-rata tersebut kemudian ditambahkan ke dalam list ‘rataH’, ‘rataS’, dan ‘rataV’ menggunakan fungsi append. Hal ini dilakukan untuk mengumpulkan nilai rata-rata dari setiap komponen warna H, S, dan V dari citra yang sedang diproses ke dalam list terpisah

Ekstraksi fitur warna yang terakhir adalah warna *grayscale*. Citra yang sudah dibaca sebelumnya diubah menjadi citra *grayscale*. Langkah ini bertujuan untuk mengkonversi citra menjadi skala keabuan dengan hanya satu saluran warna. Setelah citra merubah menjadi *grayscale*, nilai rata-rata dari citra tersebut dihitung dan disimpan dalam variabel ‘rata’. Hal ini memberikan nilai rata-rata intensitas piksel dari citra *grayscale* yang sedang diproses. Setelah itu, dihitung juga nilai standar deviasi dan variasinya.

```
for gbr in data:  
    gbr_read = cv2.imread(os.path.join(path, gbr))
```

```

gbr_rgb = cv2.cvtColor(gbr_read, cv2.COLOR_BGR2RGB)
(R, G, B) = cv2.split(gbr_rgb)
meanR = np.mean(R)
rataR.append(meanR)
meanG = np.mean(G)
rataG.append(meanG)
meanB = np.mean(B)
rataB.append(meanB)

gbr_hsv = cv2.cvtColor(gbr_read, cv2.COLOR_BGR2HSV)
H = gbr_hsv[:, :, 0]
S = gbr_hsv[:, :, 1]
V = gbr_hsv[:, :, 2]
meanH = np.mean(H)
rataH.append(meanH)
meanS = np.mean(S)
rataS.append(meanS)
meanV = np.mean(V)
rataV.append(meanV)

gbr_gray = cv2.cvtColor(gbr_read, cv2.COLOR_BGR2GRAY)
rata = np.mean(gbr_gray)
ratagray.append(rata)
standar = np.std(gbr_gray)
stand.append(standar)
variasi = np.var(gbr_gray)
vari.append(standar)

```

Setelah melakukan ekstraksi fitur wana, selanjutnya melakukan ekstraksi fitur bentuk dari citra dalam format *grayscale*. Pertama, sebuah proses *thresholding* dilakukan pada gambar grayscale. Selanjutnya, dilakukan pencarian kontur pada gambar. Jika terdapat kontur yang ditemukan (panjang kontur lebih dari 0), maka dilakukan langkah-langkah berikut:

Kontur dengan luas terbesar diidentifikasi menggunakan max. Kontur ini memiliki luas yang paling besar dan dianggap sebagai objek utama dalam gambar. Luas kontur tersebut dihitung dan disimpan dalam variabel ‘area’. Hal ini memberikan nilai luas objek dalam satuan piksel. Selanjutnya menghitung panjang keliling kontur dengan parameter True

yang menunjukkan bahwa kontur tersebut merupakan kontur tertutup. Panjang keliling ini merupakan ukuran perimeter dari objek.

Jika tidak terdapat kontur yang ditemukan (panjang kontur sama dengan 0), maka area dan panjang keliling objek diatur sebagai 0. Terakhir, nilai area dan panjang keliling objek tersebut ditambahkan ke dalam list luas dan perimeter. Hal ini dilakukan untuk mengumpulkan nilai area dan panjang keliling objek dari setiap gambar ke dalam list terpisah.

```
_ , thresh = cv2.threshold(gbr_gray, 127, 255, 0)
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

if len(contours) > 0:
    contour = max(contours, key=cv2.contourArea)
    area = cv2.contourArea(contour)
    perimeter_val = cv2.arcLength(contour, True)
else:
    area = 0
    perimeter_val = 0

luas.append(area)
perimeter.append(perimeter_val)
```

Ekstraksi fitur yang terakhir adalah tekstur menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*). Matriks GLCM dibuat dengan parameter [1] dan [0] yang menunjukkan jarak dan sudut, tingkat keabuan pada citra sebesar 256, parameter ‘symmetric=True’ untuk menghasilkan matriks GLCM simetris, dan ‘normed=True’ untuk menghasilkan matriks GLCM yang dinormalisasi. Setelah matriks GLCM terbentuk, fitur-fitur tekstur seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* di ekstraksi. Kemudian masing-masing nilainya ditambahkan ke dalam list untuk mengumpulkan nilai-nilai tekstur dari setiap citra ke dalam list terpisah.

```
glcm = graycomatrix(gbr_gray, [1], [0], 256,
symmetric=True, normed=True)
contrast_val = graycoprops(glcm, 'contrast')[0, 0]
```

```
dissimilarity_val = graycoprops(glcm, 'dissimilarity')[0, 0]
homogeneity_val = graycoprops(glcm, 'homogeneity')[0, 0]
energy_val = graycoprops(glcm, 'energy')[0, 0]
correlation_val = graycoprops(glcm, 'correlation')[0, 0]

contrast.append(contrast_val)
dissimilarity.append(dissimilarity_val)
homogeneity.append(homogeneity_val)
energy.append(energy_val)
correlation.append(correlation_val)
```

Selanjutnya mengatur jenis kopra sebagai ‘1’ dan menambahkannya ke dalam list label. Langkah ini bertujuan untuk memberikan label pada data yang sedang diproses ‘1’ disini merepresentasikan jenis kopra edible. ‘2’ merupakan jenis kopra regular, dan ‘3’ merupakan jenis kopra reject.

```
Jenis_Kopra = '1'
label.append(Jenis_Kopra)
```

Membuat Dataframe yang merupakan wadah untuk menyimpan data fitur yang telah diekstraksi sebelumnya. Setiap fitur memiliki Dataframe tersendiri yang dibentuk menggunakan library pandas. Dataframe data1 hingga data17 dibentuk untuk masing-masing fitur. Setiap Dataframe mewakili satu fitur dan kolomnya dinamai sesuai dengan fitur yang bersangkutan. Misalnya, Dataframe data1 memiliki kolom '*Mean-R*' yang berisi nilai rata-rata komponen merah (R) dari gambar-gambar yang diproses.

Selanjutnya, semua Dataframe fitur digabungkan menjadi satu Dataframe gabungan. List listdata1 berisi semua Dataframe fitur yang telah dibentuk sebelumnya. Penggunaan axis=1 dalam fungsi concat mengindikasikan bahwa penggabungan dilakukan berdasarkan kolom. Dengan demikian, hasil dari penggabungan Dataframe-fitur ini adalah Dataframe gabung1, yang berisi semua fitur yang telah diekstraksi sebagai kolom-kolomnya.

```
data1 = pd.DataFrame(rataR, columns=['Mean-R'])
```

```

data2 = pd.DataFrame(rataG, columns=[ 'Mean-G' ])
data3 = pd.DataFrame(rataB, columns=[ 'Mean-B' ])
data4 = pd.DataFrame(rataH, columns=[ 'Mean-H' ])
data5 = pd.DataFrame(rataV, columns=[ 'Mean-V' ])
data6 = pd.DataFrame(rataS, columns=[ 'Mean-s' ])
data7 = pd.DataFrame(ratagray, columns=[ 'Mean-Gray' ])
data8 = pd.DataFrame(stand, columns=[ 'Standar-Deviasi' ])
data9 = pd.DataFrame(vari, columns=[ 'Variasi' ])
data10 = pd.DataFrame(luas, columns=[ 'Luas' ])
data11 = pd.DataFrame(perimeter, columns=[ 'Perimeter' ])
data12 = pd.DataFrame(contrast, columns=[ 'Contrast' ])
data13 = pd.DataFrame(dissimilarity,
columns=[ 'Dissimilarity' ])
data14 = pd.DataFrame(homogeneity, columns=[ 'Homogeneity' ])
data15 = pd.DataFrame(energy, columns=[ 'Energy' ])
data16 = pd.DataFrame(correlation, columns=[ 'Correlation' ])
data17 = pd.DataFrame(label, columns=[ 'Jenis Kopra' ])

listdata1 = [data1, data2, data3, data4, data5, data6, data7,
data8, data9, data10, data11, data12, data13, data14, data15,
data16, data17]
gabung1 = pd.concat(listdata1, axis=1)

```

Setelah semua fitur diekstraksi dan disimpan dalam Dataframe-fitur terpisah, langkah selanjutnya adalah menggabungkan semua Dataframe-fitur tersebut menjadi satu Dataframe tunggal. Pertama, list ‘listdata’ dibentuk untuk menyimpan Dataframe-fitur yang telah dibuat sebelumnya. Dataframe-fitur dalam ‘listdata’ digabungkan menjadi satu Dataframe tunggal yang akan disimpan dalam variabel total. Penggunaan ignore\_index=True memastikan indeks pada Dataframe gabungan diatur ulang secara otomatis. Selanjutnya, Dataframe ‘total’ disimpan dalam format CSV dan Excel.

```

listdata = [gabung1, gabung2, gabung3]
total = pd.concat(listdata, ignore_index=True)
total.to_csv("ekstrasi-fitur-kopra-nonTelungkup.csv",
index=False)
total.to_excel("ekstrasi-fitur-kopra-nonTelungkup.xlsx",
index=False)

print("tahap ekstraksi fitur telah slesai dibuat")

```

### 3) Split Dependent dan Independent Value

Langkah pertama yang dilakukan untuk split dependent dan independent value adalah membaca dataset dengan mengimpor *library* yang dibutuhkan. Selanjutnya, menggunakan fungsi `read_csv` untuk membaca file CSV dan memuatnya ke dalam variabel `def_kopra`.

```
import pandas as pd
def_Kopra = pd.read_csv('ekstrasi-fitur-kopra-
nonTelungkup.csv')
```

Setelah data dibaca, data perlu dipisahkan ke dalam dua jenis yaitu *dependent* dan *independent value*. *Independent value* merupakan atribut atau fitur yang digunakan untuk melatih model, pemilihan *Independent value* ini disesuaikan dengan skenario yang ingin dilakukan. Berikut merupakan contoh untuk skenario yang menggunakan fitur warna, bentuk, dan tekstur. Semua fitur dimasukan dalam variabel `x` dengan memilih semua baris dan kolom dengan indeks 0 hingga 15. Sedangkan *dependent value* berisi informasi tentang kategori atau kelas dari jenis kopra yaitu, kopra edible, regular, dan reject, dimana semua kelas ini dimasukan dalam variabel `y` dengan memilih semua baris dan kolom dengan indeks terakhir yaitu -1. Hal ini dilakukan untuk keperluan proses pemodelan nanti,

```
x = def_Kopra.iloc[:,0:16].values
y = def_Kopra.iloc[:,[-1]].values
```

### 4) Feature Scaling

Setelah data dibagi ke dalam dua jenis yaitu *dependent* dan *independent value*, maka langkah selanjutnya melakukan normalisasi data untuk mengubah skala atau rentang nilai dari fitur-fitur yang ada pada dataset. Penelitian ini menggunakan fungsi `MinMaxScaler` untuk menormalisasi nilai dalam attribut menjadi antara 0 dan 1.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x = scaler.fit_transform(x)
```

## 5) Train-Test Split

Selanjutnya data dibagi menjadi dua bagian yaitu data *training* dan *testing*. Pembagian ini dilakukan menggunakan fungsi yang telah tersedia dari *library* scikit-learn. Adapun pembagian antara data *training* dan data *testing* dibagi kedalam 3 scenario yaitu dengan perbandingan 50:50, 70:30, dan 90:10. Berikut pembagian dengan perbandingan dataset 50:50

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test =  
train_test_split(x,y,test_size=0.5,random_state=0)
```

Pembagian dengan perbandingan dataset 70:30

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test =  
train_test_split(x,y,test_size=0.3,random_state=0)
```

Pembagian dengan perbandingan dataset 90:10

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test =  
train_test_split(x,y,test_size=0.1,random_state=0)
```

## 6) Model Klasifikasi

Pada tahap ini dilakukan klasifikasi citra berdasarkan hasil ekstraksi fitur yang didapat pada tahap sebelumnya. Algoritma K-Nearest Neighbor digunakan untuk melakukan klasifikasi citra dengan membandingkan data *testing* dan data *training*. Pertama, mengimport *library* yang diperlukan. Kemudian, membuat objek dengan mengatur parameter *n\_neighbors* menjadi 3, yang menentukan jumlah tetangga terdekat. Selain itu, juga mengatur parameter *metric* menjadi 'euclidean', yang menandakan penggunaan metrik jarak *Euclidean* untuk mengukur kedekatan antara data. Setelah itu, objek knn tersebut dilatih menggunakan data pelatihan *x\_train* dan label kelas *y\_train*.

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn =  
KNeighborsClassifier(n_neighbors=3,metric='euclidean').fit(x_train,y_train)  
knn
```

## 7) Mengukur Performansi

Pada tahap ini dilakukan pengujian performansi untuk mengevaluasi hasil klasifikasi dari metode yang diusulkan. Untuk mengetahui performa dari algoritma *K-Nearest Neighbor*, penelitian ini menggunakan metrik akurasi, *confusion matrix*, dan *classification report*.

### a. Metrik akurasi

Hal pertama yang dilakukan adalah meng-import fungsi dan *library* yang dibutuhkan. Kemudian, menghitung jumlah prediksi yang benar berdasarkan perbandingan antara *y\_test* dan *y\_prediksi*. *Normalize=False* digunakan untuk mendapatkan jumlah prediksi yang benar dalam bentuk bilangan bulat, bukan dalam bentuk persentase. Selanjutnya menghitung panjang dari *y\_test* untuk mendapatkan jumlah seluruh prediksi yang dilakukan.

Setelah itu, menghitung nilai akurasi menggunakan fungsi *accuracy\_score* dan dua variabel yang digunakan sebagai inputan yaitu *y\_test* yang merujuk pada data target yang sebenarnya dan *y\_prediksi* yang merupakan hasil prediksi. Dan terakhir, menampilkan pesan yang berisi jumlah prediksi yang benar, jumlah seluruh prediksi dan nilai akurasi yang dihasilkan.

```
jumlah_benar = accuracy_score(y_test, y_prediksi,  
normalize=False)  
jumlah_seluruh = len(y_test)  
nilai_akurasi = accuracy_score(y_test, y_prediksi)  
  
print("Jumlah prediksi yang benar:", jumlah_benar)  
print("Jumlah seluruh prediksi:", jumlah_seluruh)  
print("Nilai akurasi", nilai_akurasi)
```

### b. *Confusion Matrix* dan *Classification Report*

Hal pertama yang dilakukan adalah meng-*import* fungsi dan *library* yang dibutuhkan. kemudian, menggunakan fungsi heatmap untuk membuat heatmap berdasarkan confusion matrix yang dihasilkan dari pemanggilan fungsi `confusion_matrix(y_test, y_prediksi)`. Heatmap digunakan untuk memvisualisasikan confusion matrix dengan warna yang berbeda-beda untuk setiap selnya. Parameter `annot=True` digunakan untuk menampilkan angka pada setiap sel heatmap. Selanjutnya, mencetak laporan klasifikasi untuk presisi, recall, f1-score, dan support untuk setiap kelas yang ada dalam data target yang sebenarnya (`y_test`) dan hasil prediksi (`y_prediksi`)

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import seaborn as sb
import numpy as np
sb.heatmap(confusion_matrix(y_test, y_prediksi), annot=True)
print(classification_report(y_test, y_prediksi))
```

## **BAB V**

### **HASIL DAN PEMBAHASAN**

#### **5.1 Verify and Validation**

Pada tahap ini, penulis melakukan verifikasi dan validasi dari tahapan-tahapan yang dilakukan pada bab 4. Verifikasi dilakukan dengan mengulas kembali setiap tahapan sebelumnya dengan tujuan memastikan setiap tahapan yang dikerjakan sesuai dan memiliki keterkaitan. Dimulai dari tahap *problem formulation*, penulis melakukan identifikasi masalah untuk dirumuskan sehingga tercipta konsep pemodelan untuk mengatasi masalah tersebut. Selanjutnya, pada tahap *conceptual model*, penulis membuat alur proses pengerjaan beserta penjelasannya. Dalam penjelasan tersebut, penulis menjelaskan fungsi dan *library* yang digunakan pada tahap *simulation*. Kemudian, pada tahap *collection input*, penulis menjelaskan jenis citra yang akan digunakan sebagai *input-an*. Selanjutnya, pada tahap *modelling*, penulis melakukan pemodelan dengan menjelaskan perhitungan manual untuk mendapatkan fitur dari suatu citra dengan mengekstrak fitur warna, bentuk, dan tekstur dari citra tersebut. Dari hasil ekstraksi fitur ini, penulis juga menjelaskan proses klasifikasinya, dengan memanfaatkan fitur-fitur tersebut sebagai nilai untuk melakukan klasifikasi. Selanjutnya, proses *simulation*, penulis melakukan simulasi berdasarkan pemodelan manual yang telah dibuat sebelumnya ke dalam *Visual Studio Code* dengan Bahasa pemrograman python.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa setiap tahapan yang telah dilakukan oleh penulis telah sesuai dan terdapat keterkaitan antara tahapan-tahapan tersebut. Hal ini dikarenakan setiap tahapan memiliki pengaruh terhadap tahapan-tahapan berikutnya. Oleh karena itu, dapat dipastikan bahwa tahapan-tahapan yang telah dibahas oleh penulis telah melalui proses verifikasi yang baik. Sedangkan untuk *Validation*, penulis melakukan implementasi dari algoritma k-nearest

*neighbour* berdasarkan hasil ekstraksi fitur warna, bentuk, dan tekstur dan pengujian dari tingkat akurasi dari algoritma tersebut.

## 5.2 Design of Experiment

Pada tahap ini, penulis melakukan pengukuran kinerja dari hasil klasifikasi dengan 12 skenario. Berikut merupakan skenario yang dilakukan pada penelitian ini.

Tabel 5.1 Skenario Pengujian

No	Parameter	Jumlah Sampel	Tujuan
1	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, dan perimeter</i>	50:50	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan bentuk dengan perbandingan dataset 50:50
2	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, dan perimeter</i>	70:30	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan bentuk dengan perbandingan dataset 70:30
3	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, dan perimeter</i>	90:10	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan bentuk dengan perbandingan dataset 90:10

4	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	50:50	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan tekstur dengan perbandingan dataset 50:50
5	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	70:30	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan tekstur dengan perbandingan dataset 70:30
6	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	90:10	Mengetahui akurasi klasifikasi jika menggunakan fitur warna dan tekstur dengan perbandingan dataset 90:10
7	<i>luas, perimeter, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	50:50	Mengetahui akurasi klasifikasi jika menggunakan fitur bentuk dan tekstur dengan perbandingan dataset 50:50
8	<i>luas, perimeter, contrast, dissimilarity,</i>	70:30	Mengetahui akurasi klasifikasi jika menggunakan fitur

	<i>homogeneity, energy, dan correlation</i>		bentuk dan tekstur dengan perbandingan dataset 70:30
9	luas, <i>perimeter, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	90:10	Mengetahui akurasi klasifikasi jika menggunakan fitur bentuk dan tekstur dengan perbandingan dataset 90:10
10	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, perimeter, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	50:50	Mengetahui akurasi klasifikasi jika menggunakan fitur warna, bentuk, dan tekstur dengan perbandingan dataset 50:50
11	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, perimeter, contrast, dissimilarity, homogeneity, energy, dan correlation</i>	70:30	Mengetahui akurasi klasifikasi jika menggunakan fitur warna, bentuk, dan tekstur dengan perbandingan dataset 70:30
12	<i>mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, perimeter,</i>	90:10	Mengetahui akurasi klasifikasi jika menggunakan fitur warna, bentuk, dan tekstur dengan

	<i>contrast, dissimilarity, homogeneity, energy, dan correlation</i>		perbandingan dataset 90:10
--	--	--	----------------------------

Melalui pengukuran ini, penulis dapat mengetahui skenario mana yang memiliki tingkat akurasi tertinggi. Analisis akurasi tersebut akan dibahas lebih lanjut pada tahap *analysis output*.

### 5.3 Output Analysis

Dalam tahap *output analysis*, penulis melakukan analisa pengujian tingkat akurasi dari hasil klasifikasi dengan menggunakan algoritma *k-nearest neighbour* dalam 12 skenario yang berbeda serta pembahasan mengenai perbandingan hasil dengan penelitian terdahulu. Untuk mendapatkan tingkat akurasi, penulis menggunakan rumus metrik akurasi, *confusion matrix*, dan *classification report*.

#### 5.3.1 Pengujian Tingkat Akurasi

Untuk mendapatkan tingkat akurasi dari hasil klasifikasi, penulis menggunakan rumus metrik akurasi, *confusion matrix*, dan *classification report*.

##### 1. Skenario 1

Pada skenario 1 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan bentuk yang terdiri *mean R, mean G, mean B, mean H, mean S, mean V, mean Gray, standar deviasi, variasi, luas, dan perimeter* dengan menggunakan perbandingan dataset 50:50 yaitu 50% untuk data *testing* dan 50% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

###### a. Metrik Akurasi

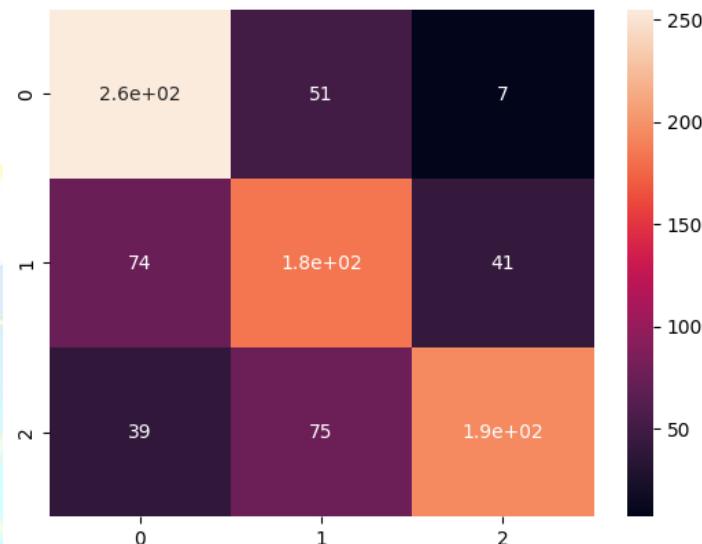
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{633}{920} \times 100\% = 69\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 69%.

b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 1 *Confusion Matrix* skenario 1

Berdasarkan Gambar 5.1 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 260. Nilai *True Negative* (TN) sebesar  $180 + 75 + 41 + 190 = 486$ . Nilai *False Positive* (FP) sebesar  $74 + 39 = 113$ . Nilai *False Negative* (FN) sebesar  $51 + 7 = 58$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 180. Nilai *True Negative* (TN) sebesar  $260 + 39 + 7 + 190 = 496$ . Nilai *False Positive* (FP) sebesar  $51 + 75 = 126$ . Nilai *False Negative*

(FN) sebesar  $74+41 = 115$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 190. Nilai *True Negative* (TN) sebesar  $260+74+51+180 = 565$ . Nilai *False Positive* (FP) sebesar  $7+41 = 48$ . Nilai *False Negative* (FN) sebesar  $39+75 = 114$ .

### c. Classification Report

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.69	0.81	0.75	313
2	0.59	0.62	0.60	299
3	0.80	0.63	0.71	308
accuracy			0.69	920
macro avg	0.70	0.69	0.69	920
weighted avg	0.70	0.69	0.69	920

Gambar 5. 2 *Classification Report* skenario 1

Berdasarkan Gambar 5.2 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{260}{260 + (74 + 39)} = 0,69$$

$$P(\text{Reguler}) = \frac{180}{180 + (51 + 75)} = 0,59$$

$$P(\text{Reject}) = \frac{190}{190 + (7 + 41)} = 0,80$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 69%, kopra regular sebesar 59%, dan kopra

reject sebesar 80%. Nilai *precision* tertinggi diperoleh oleh kopra reject.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R (\text{Edible}) = \frac{260}{260 + (51 + 7)} = 0,81$$

$$R (\text{Reguler}) = \frac{180}{180 + (74 + 41)} = 0,62$$

$$R (\text{Reject}) = \frac{190}{190 + (39 + 75)} = 0,63$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 81%, kopra regular sebesar 62%, dan kopra reject sebesar 63%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,69 \times 0,81}{0,69 + 0,81} = 0,75$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,59 \times 0,62}{0,59 + 0,62} = 0,60$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,80 \times 0,63}{0,80 + 0,63} = 0,71$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 75%, kopra regular sebesar 60%, dan kopra reject sebesar 71%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 2. Skenario 2

Pada skenario2 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan bentuk yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, luas, dan *perimeter* dengan

menggunakan perbandingan dataset 70:30 yaitu 70% untuk data *testing* dan 30% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

a. Metrik Akurasi

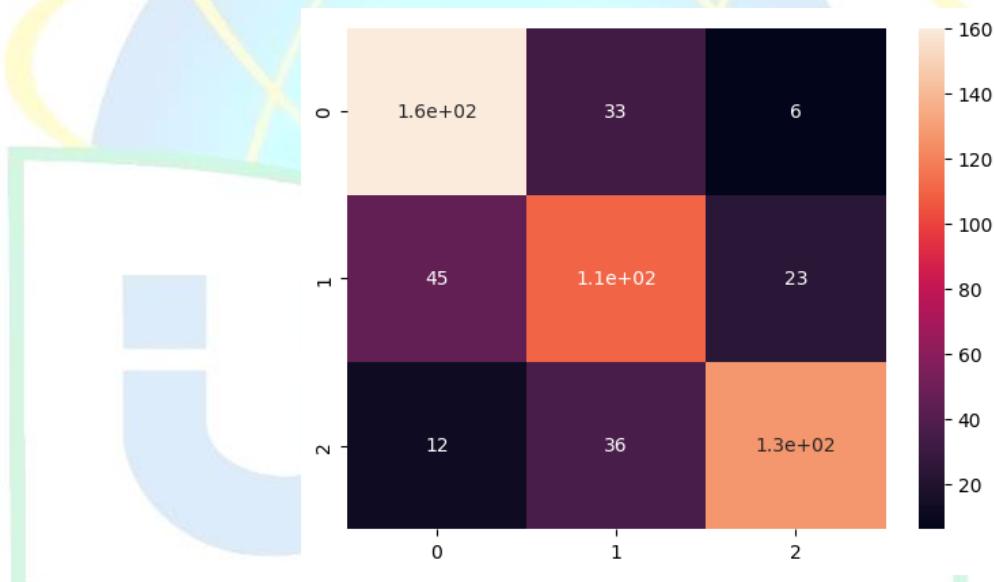
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{397}{552} \times 100\% = 71\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 71%.

b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 3 *Confusion Matrix* skenario 2

Berdasarkan Gambar 5.3 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 160. Nilai *True Negative* (TN) sebesar  $110+36+23+130 = 299$ . Nilai *False Positive* (FP) sebesar  $45+12 = 57$ . Nilai *False Negative* (FN) sebesar  $33+6 = 39$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 110. Nilai *True Negative* (TN) sebesar  $160+12+6+130 = 308$ . Nilai *False Positive* (FP) sebesar  $33+36 = 69$ . Nilai *False Negative* (FN) sebesar  $45+23 = 68$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 130. Nilai *True Negative* (TN) sebesar  $160+45+33+110 = 348$ . Nilai *False Positive* (FP) sebesar  $6+23 = 29$ . Nilai *False Negative* (FN) sebesar  $12+36 = 48$ .

#### c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.74	0.80	0.77	199
2	0.61	0.62	0.62	178
3	0.81	0.73	0.77	175
accuracy			0.72	552
macro avg	0.72	0.72	0.72	552
weighted avg	0.72	0.72	0.72	552

Gambar 5. 4 *Classification Report* skenario 2

Berdasarkan Gambar 5.4 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{160}{160 + (45 + 12)} = 0,74$$

$$P(\text{Reguler}) = \frac{110}{110 + (33 + 36)} = 0,61$$

$$P(\text{Reject}) = \frac{130}{130 + (6 + 23)} = 0,81$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 74%, kopra regular sebesar 61%, dan kopra reject sebesar 81%. Nilai *precision* tertinggi diperoleh oleh kopra reject.

2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{160}{160 + (33 + 6)} = 0,80$$

$$R(\text{Reguler}) = \frac{110}{110 + (45 + 23)} = 0,62$$

$$R(\text{Reject}) = \frac{130}{130 + (12 + 36)} = 0,73$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 80%, kopra regular sebesar 62%, dan kopra reject sebesar 73%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1\text{-score}(\text{Edible}) = \frac{2 \times 0,74 \times 0,80}{0,74 + 0,80} = 0,77$$

$$F1\text{-score}(\text{Reguler}) = \frac{2 \times 0,61 \times 0,62}{0,61 + 0,62} = 0,62$$

$$F1\text{-score}(\text{Reject}) = \frac{2 \times 0,81 \times 0,73}{0,81 + 0,73} = 0,77$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 75%, kopra regular sebesar

60%, dan kopra reject sebesar 71%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible dan reject.

### 3. Skenario 3

Pada skenario 3 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan bentuk yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, luas, dan *perimeter* dengan menggunakan perbandingan dataset 90:10 yaitu 90% untuk data *testing* dan 10% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

#### a. Metrik Akurasi

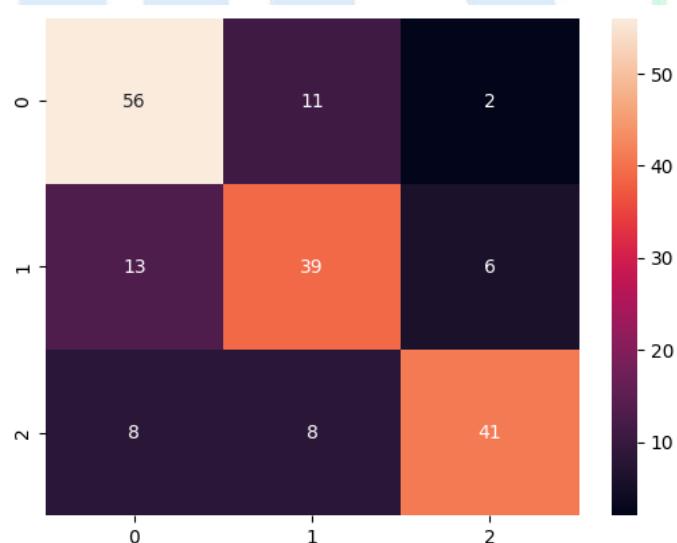
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$\text{Akurasi} = \frac{136}{184} \times 100\% = 74\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 74%.

#### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 5 *Confusion Matrix* skenario 3

Berdasarkan Gambar 5.4 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 56. Nilai *True Negative* (TN) sebesar  $39+8+6+41 = 94$ . Nilai *False Positive* (FP) sebesar  $13+8 = 21$ . Nilai *False Negative* (FN) sebesar  $11+2 = 13$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 39. Nilai *True Negative* (TN) sebesar  $59+8+2+41 = 110$ . Nilai *False Positive* (FP) sebesar  $11+8 = 19$ . Nilai *False Negative* (FN) sebesar  $13+6 = 19$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 41. Nilai *True Negative* (TN) sebesar  $56+13+11+39 = 119$ . Nilai *False Positive* (FP) sebesar  $2+6 = 8$ . Nilai *False Negative* (FN) sebesar  $8+8 = 16$ .

#### c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
1	0.73	0.81	0.77	69
2	0.67	0.67	0.67	58
3	0.84	0.72	0.77	57
<b>accuracy</b>			0.74	184
<b>macro avg</b>	0.75	0.73	0.74	184
<b>weighted avg</b>	0.74	0.74	0.74	184

Gambar 5. 6 *Classification Report* skenario 3

Berdasarkan Gambar 5.6 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{56}{56 + (13 + 8)} = 0,73$$

$$P(\text{Reguler}) = \frac{39}{39 + (11 + 8)} = 0,67$$

$$P(\text{Reject}) = \frac{41}{41 + (2 + 6)} = 0,84$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 73%, kopra regular sebesar 67%, dan kopra reject sebesar 84%. Nilai *precision* tertinggi diperoleh oleh kopra reject.

2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{56}{56 + (11 + 2)} = 0,81$$

$$R(\text{Reguler}) = \frac{39}{39 + (13 + 6)} = 0,67$$

$$R(\text{Reject}) = \frac{41}{41 + (8 + 8)} = 0,72$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 81%, kopra regular sebesar 67%, dan kopra reject sebesar 72%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score(\text{Edible}) = \frac{2 \times 0,73 \times 0,82}{0,73 + 0,82} = 0,77$$

$$F1 - score(\text{Reguler}) = \frac{2 \times 0,67 \times 0,67}{0,67 + 0,67} = 0,67$$

$$F1 - score(\text{Reject}) = \frac{2 \times 0,84 \times 0,72}{0,84 + 0,72} = 0,77$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 77%, kopra regular sebesar 67%, dan kopra reject sebesar 77%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible dan reject.

#### 4. Skenario 4

Pada skenario 4 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 50:50 yaitu 50% untuk data *testing* dan 50% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

##### a. Metrik Akurasi

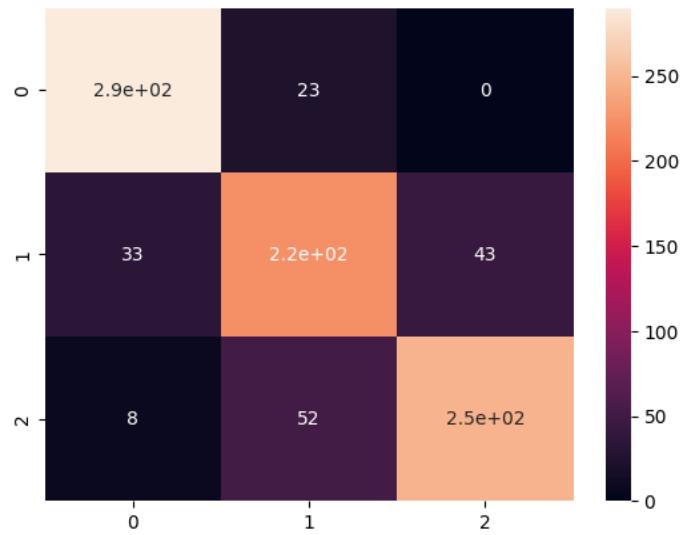
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$\text{Akurasi} = \frac{761}{920} \times 100\% = 83\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 83%.

##### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 7 *Confusion Matrix* skenario 4

Berdasarkan Gambar 5.7 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 290. Nilai *True Negative* (TN) sebesar  $220+52+43+250 = 565$ . Nilai *False Positive* (FP) sebesar  $33+8 = 41$ . Nilai *False Negative* (FN) sebesar  $23+0 = 23$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 220. Nilai *True Negative* (TN) sebesar  $290+8+0+250 = 548$ . Nilai *False Positive* (FP) sebesar  $23+52 = 75$ . Nilai *False Negative* (FN) sebesar  $33+43 = 76$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 250. Nilai *True Negative* (TN) sebesar  $290+33+23+220 = 566$ . Nilai *False Positive* (FP) sebesar  $0+43 = 43$ . Nilai *False Negative* (FN) sebesar  $8+52 = 60$ .

c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.88	0.93	0.90	313
2	0.75	0.75	0.75	299
3	0.85	0.81	0.83	308
accuracy			0.83	920
macro avg	0.83	0.83	0.83	920
weighted avg	0.83	0.83	0.83	920

Gambar 5. 8 *Classification Report* skenario 4

Berdasarkan Gambar 5.8 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{290}{290 + (33 + 8)} = 0,88$$

$$P(\text{Reguler}) = \frac{220}{220 + (23 + 52)} = 0,75$$

$$P(\text{Reject}) = \frac{250}{250 + (0 + 43)} = 0,85$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 88%, kopra regular sebesar 75%, dan kopra reject sebesar 85%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{290}{290 + (23 + 0)} = 0,93$$

$$R(\text{Reguler}) = \frac{220}{220 + (33 + 43)} = 0,75$$

$$R (\text{Reject}) = \frac{250}{250 + (8 + 52)} = 0,81$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 93%, kopra regular sebesar 75%, dan kopra reject sebesar 81%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - \text{score} (\text{Edible}) = \frac{2 \times 0,88 \times 0,93}{0,88 + 0,93} = 0,90$$

$$F1 - \text{score} (\text{Reguler}) = \frac{2 \times 0,75 \times 0,75}{0,75 + 0,75} = 0,75$$

$$F1 - \text{score} (\text{Reject}) = \frac{2 \times 0,85 \times 0,81}{0,85 + 0,81} = 0,83$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 75%, dan kopra reject sebesar 83%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 5. Skenario 5

Pada skenario 5 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 70:30 yaitu 70% untuk data *testing* dan 30% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

### a. Metrik Akurasi

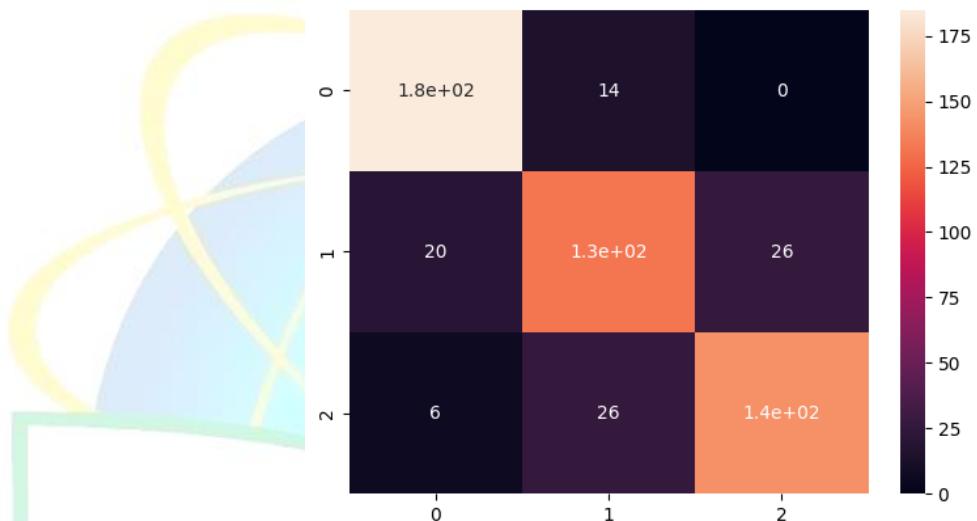
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{460}{552} \times 100\% = 83\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 83%.

#### b. Confusion Matrix

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 9 *Confusion Matrix* skenario 5

Berdasarkan Gambar 5.9 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 180. Nilai *True Negative* (TN) sebesar  $130+26+26+140 = 322$ . Nilai *False Positive* (FP) sebesar  $20+6 = 26$ . Nilai *False Negative* (FN) sebesar  $14+0 = 14$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 130. Nilai *True*

*Negative* (TN) sebesar  $180+6+0+140 = 326$ . Nilai *False Positive* (FP) sebesar  $14+26 = 40$ . Nilai *False Negative* (FN) sebesar  $20+26 = 46$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 140. Nilai *True Negative* (TN) sebesar  $180+20+14+130 = 344$ . Nilai *False Positive* (FP) sebesar  $0+26 = 26$ . Nilai *False Negative* (FN) sebesar  $6+26 = 32$ .

### c. Classification Report

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.88	0.93	0.90	199
2	0.77	0.74	0.75	178
3	0.85	0.82	0.83	175
accuracy			0.83	552
macro avg	0.83	0.83	0.83	552
weighted avg	0.83	0.83	0.83	552

Gambar 5. 10 *Classification Report* skenario 5

Berdasarkan Gambar 5.10 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{180}{180 + (20 + 6)} = 0,88$$

$$P(\text{Reguler}) = \frac{130}{130 + (14 + 26)} = 0,77$$

$$P(\text{Reject}) = \frac{140}{140 + (0 + 26)} = 0,85$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 88%, kopra

regular sebesar 77%, dan kopra reject sebesar 85%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R (\text{Edible}) = \frac{180}{180 + (14 + 0)} = 0,93$$

$$R (\text{Reguler}) = \frac{130}{130 + (20 + 26)} = 0,74$$

$$R (\text{Reject}) = \frac{140}{140 + (6 + 26)} = 0,83$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 93%, kopra regular sebesar 74%, dan kopra reject sebesar 82%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,88 \times 0,93}{0,88 + 0,93} = 0,90$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,77 \times 0,74}{0,77 + 0,74} = 0,75$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,85 \times 0,82}{0,85 + 0,82} = 0,83$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 75%, dan kopra reject sebesar 83%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 6. Skenario 6

Pada skenario 6 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, *contrast*, *dissimilarity*, *homogeneity*,

*energy*, dan *correlation* dengan menggunakan perbandingan dataset 90:10 yaitu 90% untuk data *testing* dan 10% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

a. Metrik Akurasi

Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$\text{Akurasi} = \frac{151}{184} \times 100\% = 82\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 82%.

b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 11 *Confusion Matrix* skenario 6

Berdasarkan Gambar 5.11 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 63. Nilai *True Negative* (TN) sebesar  $45+12+7+43 = 107$ . Nilai *False Positive* (FP) sebesar  $6+2 = 8$ . Nilai *False Negative* (FN) sebesar  $6+0 = 6$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 45. Nilai *True Negative* (TN) sebesar  $63+2+0+43 = 108$ . Nilai *False Positive* (FP) sebesar  $6+12 = 18$ . Nilai *False Negative* (FN) sebesar  $6+7 = 13$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 43. Nilai *True Negative* (TN) sebesar  $63+6+6+45 = 120$ . Nilai *False Positive* (FP) sebesar  $0+7 = 7$ . Nilai *False Negative* (FN) sebesar  $2+12 = 14$ .

#### c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.89	0.91	0.90	69
2	0.71	0.78	0.74	58
3	0.86	0.75	0.80	57
accuracy			0.82	184
macro avg	0.82	0.81	0.82	184
weighted avg	0.82	0.82	0.82	184

Gambar 5. 12 *Classification Report* skenario 6

Berdasarkan Gambar 5.12 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{63}{63 + (6 + 2)} = 0,89$$

$$P(\text{Reguler}) = \frac{45}{45 + (6 + 12)} = 0,71$$

$$P(\text{Reject}) = \frac{43}{43 + (0 + 7)} = 0,86$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 89%, kopra regular sebesar 71%, dan kopra reject sebesar 86%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{63}{63 + (6 + 0)} = 0,91$$

$$R(\text{Reguler}) = \frac{45}{45 + (6 + 7)} = 0,78$$

$$R(\text{Reject}) = \frac{43}{43 + (2 + 12)} = 0,75$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 91%, kopra regular sebesar 78%, dan kopra reject sebesar 75%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1\text{-score}(\text{Edible}) = \frac{2 \times 0,89 \times 0,91}{0,89 + 0,93} = 0,90$$

$$F1\text{-score}(\text{Reguler}) = \frac{2 \times 0,71 \times 0,78}{0,86 + 0,75} = 0,74$$

$$F1\text{-score}(\text{Reject}) = \frac{2 \times 0,86 \times 0,75}{0,86 + 0,75} = 0,80$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 74%, dan kopra reject sebesar 80%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 7. Skenario 7

Pada skenario 7 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur bentuk dan tekstur yang luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 50:50 yaitu 50% untuk data *testing* dan 50% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

#### a. Metrik Akurasi

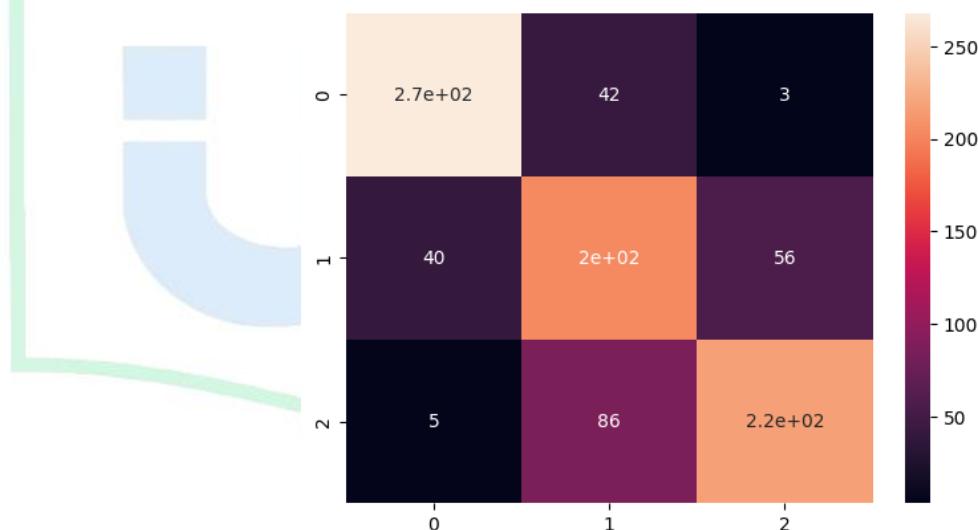
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{688}{920} \times 100\% = 75\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 75%.

#### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 13 *Confusion Matrix* skenario 7

Berdasarkan Gambar 5.13 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0,

1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 270. Nilai *True Negative* (TN) sebesar  $200+86+56+220 = 562$ . Nilai *False Positive* (FP) sebesar  $40+5 = 45$ . Nilai *False Negative* (FN) sebesar  $42+3 = 45$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 200. Nilai *True Negative* (TN) sebesar  $270+5+3+220 = 498$ . Nilai *False Positive* (FP) sebesar  $42+86 = 128$ . Nilai *False Negative* (FN) sebesar  $40+56 = 96$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 220. Nilai *True Negative* (TN) sebesar  $270+40+42+220 = 552$ . Nilai *False Positive* (FP) sebesar  $3+56 = 59$ . Nilai *False Negative* (FN) sebesar  $5+86 = 91$ .

c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.86	0.86	0.86	313
2	0.61	0.68	0.64	299
3	0.79	0.70	0.74	308
accuracy			0.75	920
macro avg	0.75	0.75	0.75	920
weighted avg	0.75	0.75	0.75	920

Gambar 5. 14 *Classification Report* skenario 7

Berdasarkan Gambar 5.14 kita dapat melihat nilai presisi, *recall*, dan *f1-score* pada masing-masing kelas.

1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{270}{270 + (40 + 5)} = 0,86$$

$$P(\text{Reguler}) = \frac{200}{200 + (42 + 86)} = 0,61$$

$$P(\text{Reject}) = \frac{220}{220 + (3 + 56)} = 0,79$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 86%, kopra regular sebesar 61%, dan kopra reject sebesar 79%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{270}{270 + (42 + 3)} = 0,86$$

$$R(\text{Reguler}) = \frac{200}{200 + (40 + 56)} = 0,68$$

$$R(\text{Reject}) = \frac{220}{220 + (5 + 86)} = 0,70$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 86%, kopra regular sebesar 68%, dan kopra reject sebesar 70%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score(\text{Edible}) = \frac{2 \times 0,86 \times 0,86}{0,86 + 0,86} = 0,86$$

$$F1 - score (Reguler) = \frac{2 \times 0,61 \times 0,68}{0,61 + 0,68} = 0,64$$

$$F1 - score (Reject) = \frac{2 \times 0,79 \times 0,70}{0,79 + 0,70} = 0,74$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 86%, kopra regular sebesar 64%, dan kopra reject sebesar 74%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 8. Skenario 8

Pada skenario 8 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur bentuk dan tekstur yang terdiri luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 70:30 yaitu 70% untuk data *testing* dan 30% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

### a. Metrik Akurasi

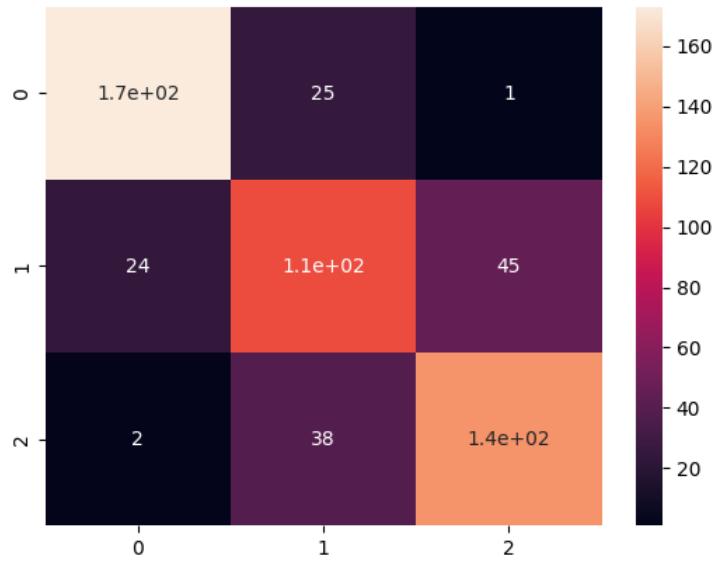
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{417}{552} \times 100\% = 76\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 76%.

### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 15 *Confusion Matrix* skenario 8

Berdasarkan Gambar 5.15 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edible, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 170. Nilai *True Negative* (TN) sebesar  $110 + 38 + 45 + 140 = 233$ . Nilai *False Positive* (FP) sebesar  $24 + 2 = 26$ . Nilai *False Negative* (FN) sebesar  $25 + 1 = 26$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 110. Nilai *True Negative* (TN) sebesar  $170 + 2 + 1 + 140 = 313$ . Nilai *False Positive* (FP) sebesar  $25 + 38 = 63$ . Nilai *False Negative* (FN) sebesar  $24 + 45 = 69$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 140. Nilai *True Negative* (TN) sebesar  $170 + 24 + 25 + 110 = 329$ . Nilai *False Positive* (FP) sebesar  $1 + 45 = 46$ . Nilai *False Negative* (FN) sebesar  $2 + 38 = 40$ .

c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
1	0.87	0.87	0.87	199
2	0.63	0.61	0.62	178
3	0.75	0.77	0.76	175
<i>accuracy</i>			0.76	552
<i>macro avg</i>	0.75	0.75	0.75	552
<i>weighted avg</i>	0.75	0.76	0.75	552

Gambar 5. 16 *Classification Report* skenario 8

Berdasarkan Gambar 5.16 kita dapat melihat nilai presisi, *recall*, dan *f1-score* pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{170}{170 + (24 + 2)} = 0,87$$

$$P(\text{Reguler}) = \frac{110}{110 + (25 + 38)} = 0,63$$

$$P(\text{Reject}) = \frac{140}{140 + (1 + 45)} = 0,75$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 87%, kopra regular sebesar 63%, dan kopra reject sebesar 75%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{170}{170 + (25 + 1)} = 0,87$$

$$R(\text{Reguler}) = \frac{110}{110 + (24 + 45)} = 0,61$$

$$R (\text{Reject}) = \frac{140}{140 + (2 + 38)} = 0,77$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 87%, kopra regular sebesar 61%, dan kopra reject sebesar 77%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,87 \times 0,87}{0,87 + 0,87} = 0,87$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,63 \times 0,61}{0,63 + 0,61} = 0,62$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,75 \times 0,77}{0,75 + 0,77} = 0,76$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 87%, kopra regular sebesar 62%, dan kopra reject sebesar 76%.

Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 9. Skenario 9

Pada skenario 9 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur bentuk dan tekstur yang terdiri luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 90:10 yaitu 90% untuk data *testing* dan 10% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

### a. Metrik Akurasi

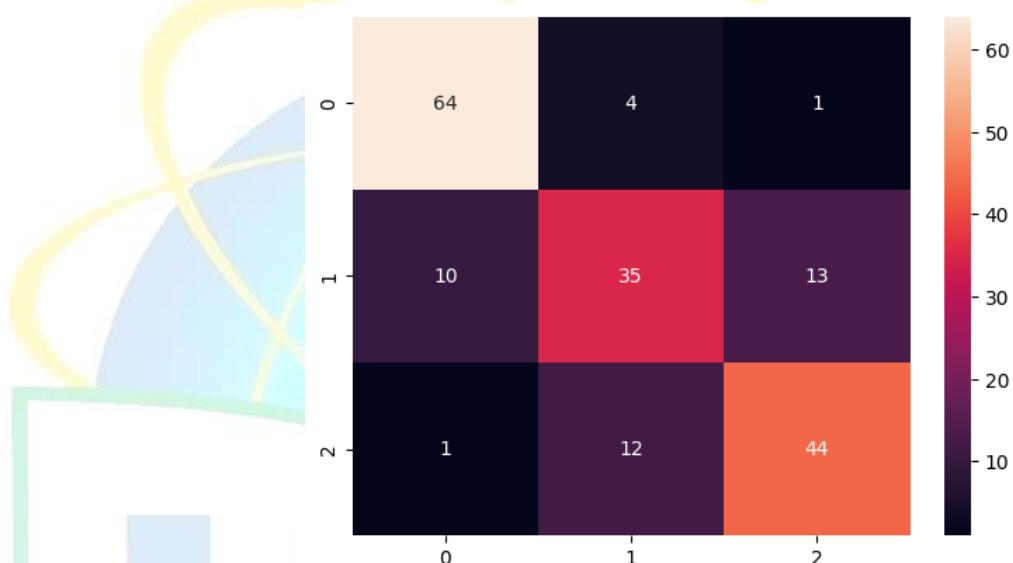
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{143}{184} \times 100\% = 77\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 77%.

#### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 17 *Confusion Matrix* skenario 9

Berdasarkan Gambar 5.17 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edible, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 64. Nilai *True Negative* (TN) sebesar  $35+12+13+44 = 104$ . Nilai *False Positive* (FP) sebesar  $10+1 = 11$ . Nilai *False Negative*

(FN) sebesar  $4+1 = 5$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 35. Nilai *True Negative* (TN) sebesar  $64+1+44 = 110$ . Nilai *False Positive* (FP) sebesar  $4+12 = 16$ . Nilai *False Negative* (FN) sebesar  $10+13 = 23$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 44. Nilai *True Negative* (TN) sebesar  $64+10+4+35 = 113$ . Nilai *False Positive* (FP) sebesar  $1+13 = 14$ . Nilai *False Negative* (FN) sebesar  $1+12 = 13$ .

### c. Classification Report

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.85	0.93	0.89	69
2	0.69	0.60	0.64	58
3	0.76	0.77	0.77	57
accuracy			0.78	184
macro avg	0.77	0.77	0.77	184
weighted avg	0.77	0.78	0.77	184

Gambar 5. 18 *Classification Report* skenario 9

Berdasarkan Gambar 5.18 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{64}{64 + (10 + 1)} = 0,85$$

$$P(\text{Reguler}) = \frac{35}{35 + (4 + 12)} = 0,69$$

$$P(\text{Reject}) = \frac{44}{44 + (1 + 13)} = 0,76$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision*

yang diperoleh untuk kopra edible sebesar 85%, kopra regular sebesar 69%, dan kopra reject sebesar 76%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R (\text{Edible}) = \frac{64}{64 + (4 + 1)} = 0,93$$

$$R (\text{Reguler}) = \frac{35}{35 + (10 + 13)} = 0,60$$

$$R (\text{Reject}) = \frac{44}{44 + (1 + 12)} = 0,77$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 93%, kopra regular sebesar 60%, dan kopra reject sebesar 77%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,85 \times 0,93}{0,85 + 0,93} = 0,89$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,69 \times 0,60}{0,69 + 0,60} = 0,64$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,76 \times 0,77}{0,76 + 0,77} = 0,77$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 89%, kopra regular sebesar 64%, dan kopra reject sebesar 77%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 10. Skenario 10

Pada skenario 10 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna, bentuk, dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*,

*mean Gray*, standar deviasi, variasi, luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 50:50 yaitu 50% untuk data *testing* dan 50% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

a. Metrik Akurasi

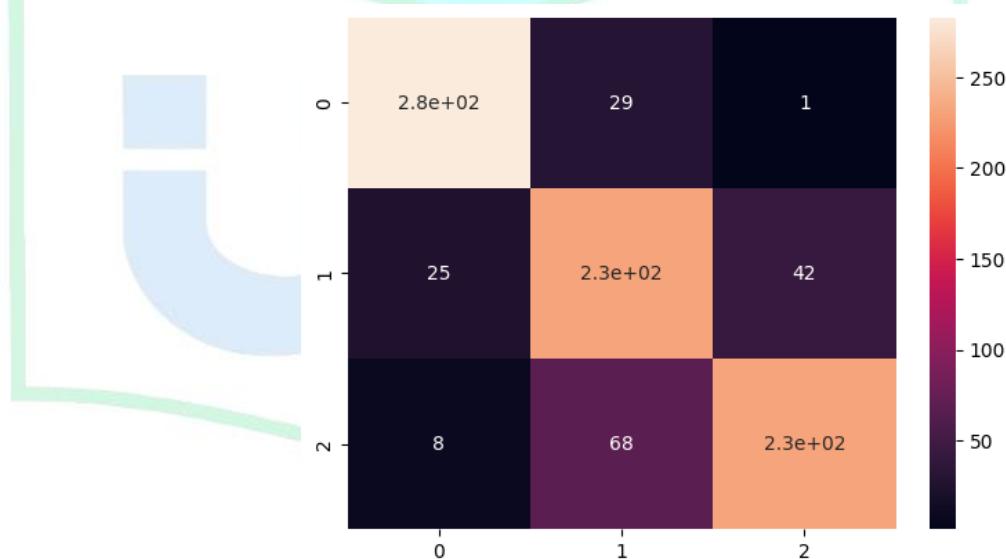
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$\text{Akurasi} = \frac{747}{920} \times 100\% = 81\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 81%.

b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 19 *Confusion Matrix* skenario 10

Berdasarkan Gambar 5.19 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2.

Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 280. Nilai *True Negative* (TN) sebesar  $230+68+42+230 = 570$ . Nilai *False Positive* (FP) sebesar  $25+8 = 33$ . Nilai *False Negative* (FN) sebesar  $29+1 = 30$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 230. Nilai *True Negative* (TN) sebesar  $280+8+1+230 = 519$ . Nilai *False Positive* (FP) sebesar  $29+68 = 97$ . Nilai *False Negative* (FN) sebesar  $25+42 = 67$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 230. Nilai *True Negative* (TN) sebesar  $280+25+29+230 = 564$ . Nilai *False Positive* (FP) sebesar  $1+42 = 43$ . Nilai *False Negative* (FN) sebesar  $8+68 = 76$ .

#### c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.90	0.90	0.90	313
2	0.71	0.78	0.74	299
3	0.84	0.75	0.80	308
accuracy			0.81	920
macro avg	0.81	0.81	0.81	920
weighted avg	0.82	0.81	0.81	920

Gambar 5. 20 *Classification Report* skenario 10

Berdasarkan Gambar 5.20 kita dapat melihat nilai presisi, *recall*, dan *f1-score* pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{280}{280 + (25 + 8)} = 0,90$$

$$P(\text{Reguler}) = \frac{230}{230 + (29 + 68)} = 0,71$$

$$P(\text{Reject}) = \frac{230}{230 + (1 + 42)} = 0,84$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 71%, dan kopra reject sebesar 84%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{280}{280 + (29 + 1)} = 0,94$$

$$R(\text{Reguler}) = \frac{230}{230 + (25 + 42)} = 0,78$$

$$R(\text{Reject}) = \frac{230}{230 + (8 + 68)} = 0,75$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 94%, kopra regular sebesar 78%, dan kopra reject sebesar 75%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1\text{-}score(\text{Edible}) = \frac{2 \times 0,90 \times 0,90}{0,90 + 0,90} = 0,90$$

$$F1\text{-}score(\text{Reguler}) = \frac{2 \times 0,71 \times 0,78}{0,71 + 0,78} = 0,74$$

$$F1\text{-}score(\text{Reject}) = \frac{2 \times 0,84 \times 0,75}{0,84 + 0,75} = 0,80$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 74%, dan kopra reject sebesar 80%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 11. Skenario 11

Pada skenario 11 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna, bentuk, dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 70:30 yaitu 70% untuk data *testing* dan 30% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

### a. Metrik Akurasi

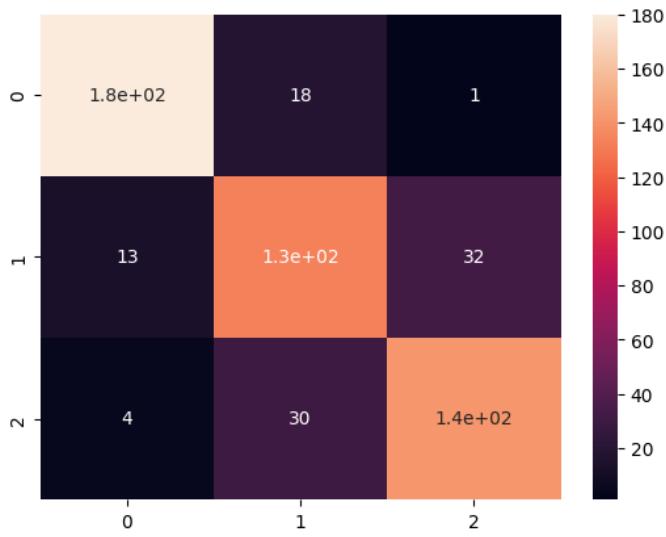
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$\text{Akurasi} = \frac{454}{552} \times 100\% = 82\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 82%.

### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 21 *Confusion Matrix* skenario 11

Berdasarkan Gambar 5.21 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edile, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 180. Nilai *True Negative* (TN) sebesar  $130 + 30 + 32 + 140 = 332$ . Nilai *False Positive* (FP) sebesar  $13 + 4 = 17$ . Nilai *False Negative* (FN) sebesar  $18 + 1 = 19$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 130. Nilai *True Negative* (TN) sebesar  $180 + 4 + 1 + 140 = 325$ . Nilai *False Positive* (FP) sebesar  $18 + 30 = 48$ . Nilai *False Negative* (FN) sebesar  $13 + 32 = 45$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 140. Nilai *True Negative* (TN) sebesar  $180 + 13 + 18 + 130 = 341$ . Nilai *False Positive*

(FP) sebesar  $1+32 = 33$ . Nilai *False Negative* (FN) sebesar  $4+30 = 34$ .

c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.91	0.90	0.91	199
2	0.73	0.75	0.74	178
3	0.81	0.81	0.81	175
accuracy			0.82	552
macro avg	0.82	0.82	0.82	552
weighted avg	0.82	0.82	0.82	552

Gambar 5. 22 *Classification Report* skenario 11

Berdasarkan Gambar 5.2 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

- 1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{180}{180 + (13 + 4)} = 0,91$$

$$P(\text{Reguler}) = \frac{130}{130 + (18 + 30)} = 0,73$$

$$P(\text{Reject}) = \frac{140}{140 + (1 + 32)} = 0,81$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 91%, kopra regular sebesar 73%, dan kopra reject sebesar 81%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R(\text{Edible}) = \frac{180}{180 + (18 + 1)} = 0,90$$

$$R (\text{Reguler}) = \frac{130}{130 + (13 + 32)} = 0,75$$

$$R (\text{Reject}) = \frac{140}{140 + (4 + 30)} = 0,81$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 90%, kopra regular sebesar 75%, dan kopra reject sebesar 81%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,91 \times 0,90}{0,91 + 0,90} = 0,91$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,73 \times 0,75}{0,73 + 0,75} = 0,74$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,81 \times 0,81}{0,81 + 0,81} = 0,81$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 91%, kopra regular sebesar 74%, dan kopra reject sebesar 81%. Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

## 12. Skenario 12

Pada skenario 12 dilakukan percobaan klasifikasi menggunakan ekstraksi fitur warna, bentuk, dan tekstur yang terdiri *mean R*, *mean G*, *mean B*, *mean H*, *mean S*, *mean V*, *mean Gray*, standar deviasi, variasi, luas, *perimeter*, *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation* dengan menggunakan perbandingan dataset 90:10 yaitu 90% untuk data *testing* dan 10% untuk data *training*. Berikut merupakan hasil pengujian tingkat akurasinya.

### a. Metrik Akurasi

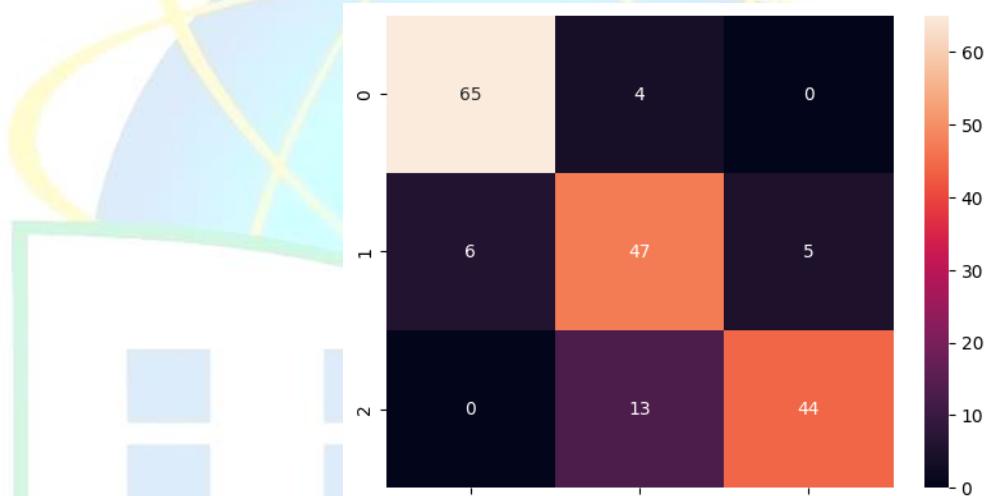
Perhitungan akurasi dengan rumus (2.14) sebagai berikut:

$$Akurasi = \frac{156}{184} \times 100\% = 85\%$$

Pada pengujian data testing menggunakan metrik akurasi didapatkan akurasi sebesar 85%.

### b. *Confusion Matrix*

Berikut merupakan hasil pengujian menggunakan *confusion matrix*.



Gambar 5. 23 *Confusion Matrix* skenario 12

Berdasarkan Gambar 5.23 *confusion matrix*  $3 \times 3$  terdiri atas dua bagian yaitu hasil prediksi dan hasil aktual. Hasil aktual terdiri atas tiga kelas yaitu kelas 0, 1, dan 2. Dimana 0 merepresentasikan jenis kopra edible, 1 merupakan jenis kopra regular, dan 2 merupakan jenis kopra reject. Sedangkan hasil prediksi juga terdiri dari kelas 0, 1, dan 2.

Dalam kasus kelas jenis kopra edible, kita memiliki nilai *True Positive* (TP) sebesar 65. Nilai *True*

*Negative* (TN) sebesar  $47+13+5+44 = 109$ . Nilai *False Positive* (FP) sebesar  $6+0 = 6$ . Nilai *False Negative* (FN) sebesar  $4+0 = 4$ . Sedangkan untuk kelas kopra regular, kita memiliki *True Positive* (TP) sebesar 47. Nilai *True Negative* (TN) sebesar  $65+0+0+44 = 109$ . Nilai *False Positive* (FP) sebesar  $4+13 = 17$ . Nilai *False Negative* (FN) sebesar  $6+5 = 11$ . Dan untuk kelas kopra reject, kita memiliki *True Positive* (TP) sebesar 44. Nilai *True Negative* (TN) sebesar  $65+6+4+47 = 122$ . Nilai *False Positive* (FP) sebesar  $5+0 = 5$ . Nilai *False Negative* (FN) sebesar  $0+13 = 13$ .

c. *Classification Report*

Berikut merupakan hasil pengujian menggunakan *classification report*.

	precision	recall	f1-score	support
1	0.92	0.94	0.93	69
2	0.73	0.81	0.77	58
3	0.90	0.77	0.83	57
accuracy			0.85	184
macro avg	0.85	0.84	0.84	184
weighted avg	0.85	0.85	0.85	184

Gambar 5. 24 *Classification Report* skenario 12

Berdasarkan Gambar 5.24 kita dapat melihat nilai presisi, *recall*, dan f1-score pada masing-masing kelas.

1) *Precision* dapat dihitung menggunakan rumus (2.15).

$$P(\text{Edible}) = \frac{65}{65 + (6 + 0)} = 0,92$$

$$P(\text{Reguler}) = \frac{47}{47 + (4 + 13)} = 0,73$$

$$P(\text{Reject}) = \frac{44}{44 + (5 + 0)} = 0,90$$

Nilai *precision* merupakan persentase kasus positif yang diidentifikasi dengan benar dari keseluruhan kasus yang diidentifikasi sebagai positif oleh model. Nilai *precision* yang diperoleh untuk kopra edible sebesar 92%, kopra regular sebesar 73%, dan kopra reject sebesar 90%. Nilai *precision* tertinggi diperoleh oleh kopra edible.

- 2) *Recall* dapat dihitung menggunakan rumus (2.16).

$$R (\text{Edible}) = \frac{65}{65 + (4 + 0)} = 0,94$$

$$R (\text{Reguler}) = \frac{47}{47 + (6 + 5)} = 0,81$$

$$R (\text{Reject}) = \frac{44}{44 + (0 + 13)} = 0,77$$

Nilai *recall* merupakan persentase kasus positif yang diidentifikasi dengan benar oleh model dari keseluruhan kasus positif yang sebenarnya. Nilai *recall* yang diperoleh untuk kopra edible sebesar 94%, kopra regular sebesar 81%, dan kopra reject sebesar 77%. Nilai *recall* tertinggi diperoleh oleh kopra edible.

- 3) *F1-score* dapat dihitung menggunakan rumus (2.17).

$$F1 - score (\text{Edible}) = \frac{2 \times 0,92 \times 0,94}{0,92 + 0,94} = 0,93$$

$$F1 - score (\text{Reguler}) = \frac{2 \times 0,73 \times 0,81}{0,73 + 0,81} = 0,77$$

$$F1 - score (\text{Reject}) = \frac{2 \times 0,90 \times 0,77}{0,90 + 0,77} = 0,83$$

*F1-score* merupakan rata-rata harmonik dari presisi dan recall untuk setiap kelas. Nilai *F1-score* yang diperoleh untuk kopra edible sebesar 93%, kopra

regular sebesar 77%, dan kopra reject sebesar 83%.

Nilai *F1-score* tertinggi diperoleh oleh kopra edible.

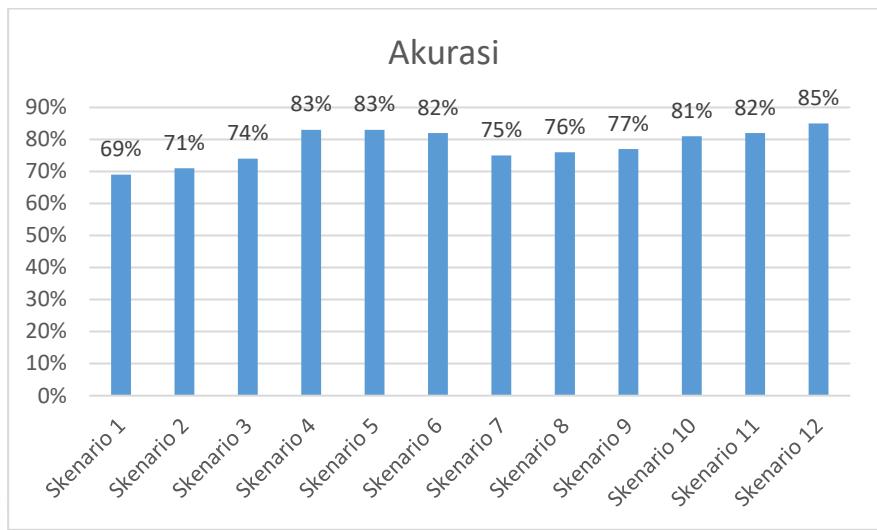
Setelah melakukan simulasi dengan 12 skenario, didapatkan hasil dari setiap skenario. Hasil eksperimen simulasi dapat dilihat pada tabel dibawah ini.

Tabel 5. 2 Hasil Akurasi Pengujian

No	Skenario	Jumlah Sampel	Akurasi	Precision	Recall	<i>F1-score</i>
1	Skenario 1	50:50	69%	70%	69%	69%
2	Skenario 2	70:30	71%	72%	72%	72%
3	Skenario 3	90:10	74%	75%	73%	74%
4	Skenario 4	50:50	83%	83%	83%	83%
5	Skenario 5	70:30	83%	83%	83%	83%
6	Skenario 6	90:10	82%	82%	81%	81%
7	Skenario 7	50:50	75%	75%	75%	75%
8	Skenario 8	70:30	76%	75%	75%	75%
9	Skenario 9	90:10	77%	77%	77%	77%
10	Skenario 10	50:50	81%	82%	81%	81%
11	Skenario 11	70:30	82%	82%	82%	82%
12	Skenario 12	90:10	85%	85%	84%	84%

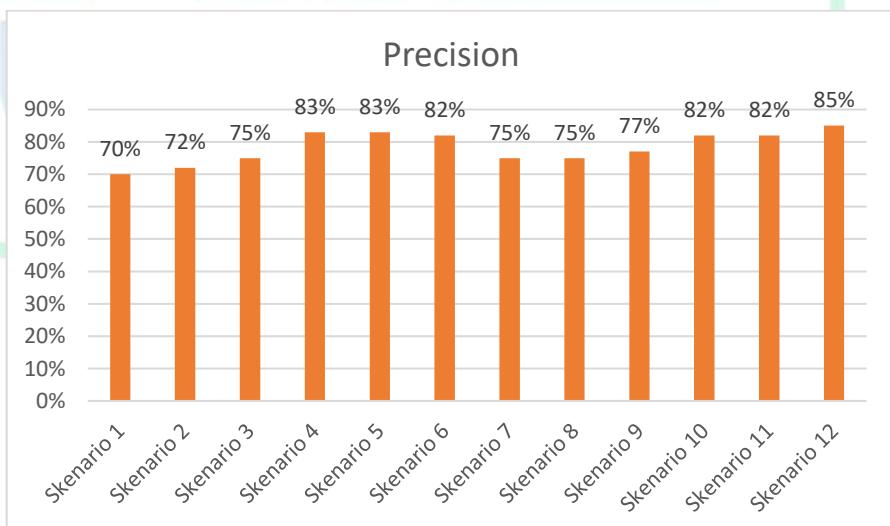
Hasil evaluasi model juga dapat dilihat dalam bentuk grafik.

Berikut merupakan hasil evaluasi akurasi terhadap model klasifikasi:



Hasil evaluasi pada grafik di atas menunjukkan tingkat akurasi terbesar yakni pada skenario 12 sebesar 85%, sementara tingkat akurasi terendah yakni skenario 1 sebesar 69%. Tingkat akurasi mengukur sejauh mana model prediksi memberikan hasil yang benar secara keseluruhan. Sehingga dapat disimpulkan bahwa, model mampu mengklasifikasikan data dengan baik.

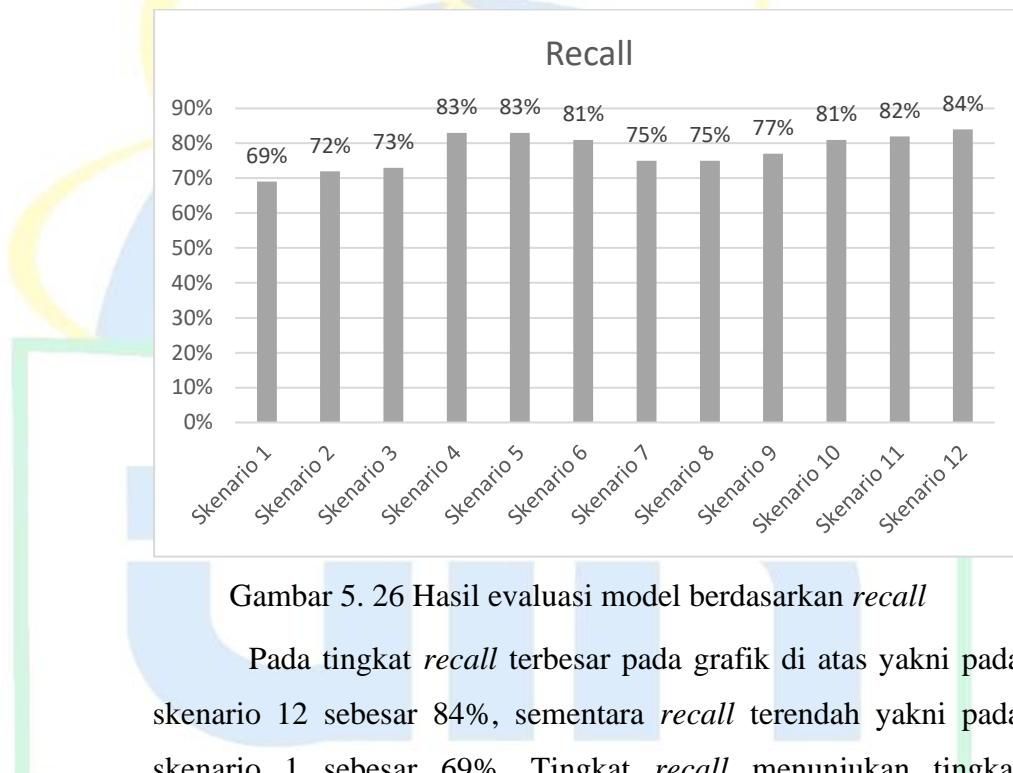
Berdasarkan evaluasi model yang telah dilakukan, berikut merupakan evaluasi *precision* terhadap model klasifikasi dalam bentuk grafik.



Gambar 5. 25 Hasil evaluasi model berdasarkan *precision*

Hasil evaluasi pada grafik di atas menunjukkan tingkat *precision* terbesar yakni pada skenario 12 sebesar 85%, sementara tingkat *precision* terendah yakni skenario 1 sebesar 70%. Tingkat *precision* menunjukkan hasil prediksi yang memberikan hasil yang benar atau relevan. Sehingga dapat disimpulkan bahwa, model mampu mengklasifikasikan data dengan baik.

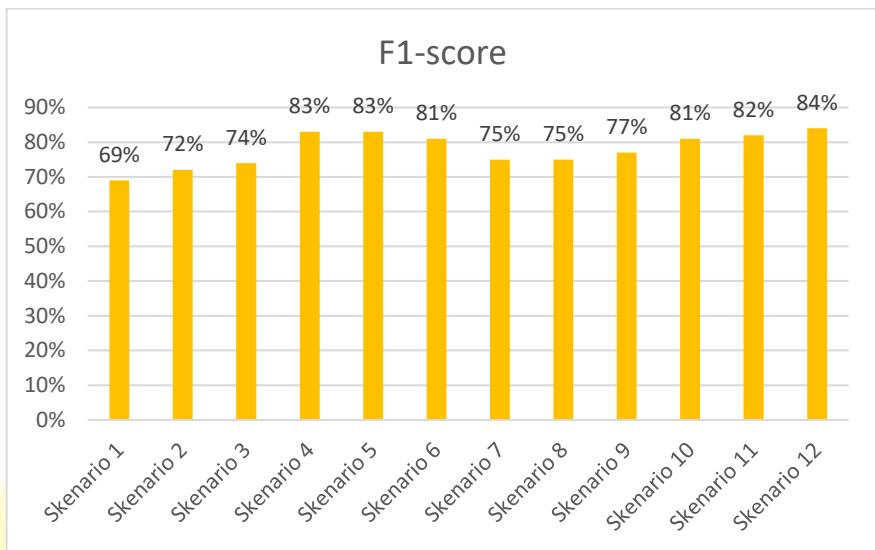
Berdasarkan evaluasi model yang telah dilakukan, berikut merupakan evaluasi *recall* terhadap model klasifikasi dalam bentuk grafik.



Gambar 5. 26 Hasil evaluasi model berdasarkan *recall*

Pada tingkat *recall* terbesar pada grafik di atas yakni pada skenario 12 sebesar 84%, sementara *recall* terendah yakni pada skenario 1 sebesar 69%. Tingkat *recall* menunjukkan tingkat keberhasilan model dalam menemukan kembali sebuah informasi. Dari nilai rata-rata *recall* yang dihasilkan, dapat disimpulkan bahwa model mampu mengklasifikasikan data dengan baik.

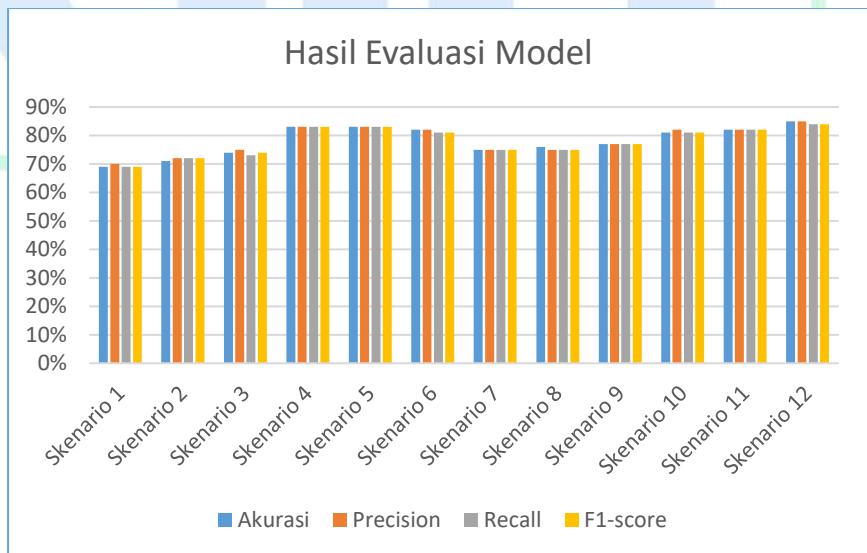
Berdasarkan evaluasi model yang telah dilakukan, maka berikut ini hasil evaluasi f1-score terhadap model klasifikasi dalam bentuk grafik:



Gambar 5. 27 Hasil evaluasi model berdasarkan *f1-score*

Pada tingkat *f1-score* terbesar pada grafik di atas yakni pada skenario 12 sebesar 84%, sementara *f1-score* terendah yaitu pada skenario 1 sebesar 69%. Tingkat *f1-score* menunjukkan perbandingan rata-rata *precision* dan *recall*. Dari rata-rata *f1-score* yang dihasilkan, dapat disimpulkan bahwa model memiliki performa yang baik dalam mengklasifikasikan model.

Berdasarkan evaluasi model yang telah dilakukan, berikut merupakan rekapitulasi hasil evaluasi terhadap model klasifikasi menggunakan metode *k-nearest neighbour* dalam bentuk grafik.



Gambar 5. 28 Hasil evaluasi model

Secara keseluruhan, hasil klasifikasi paling optimal terdapat pada penggunaan fitur warna, bentuk, dan tekstur dengan pembagian data 90:10 menghasilkan akurasi sebesar 85% dengan presisi 85%, recall 84%, dan f1-score 84%. Hasil pengujian juga menunjukkan perbedaan nilai akurasi dengan jumlah pembagian data, dimana rata-rata jumlah data dengan pembagian 90:10 menghasilkan nilai akurasi sedikit lebih tinggi dibandingkan pembagian data 70:30 dan 50:50.

### 5.3.2 Perbandingan Hasil dengan Penelitian Terdahulu

Penulis melakukan penelitian dengan topik yang serupa dengan penelitian yang dilakukan oleh (Adang et al., 2020) yaitu mengenai klasifikasi kopra namun pada fokus spesifik yang berbeda. Pada penelitian (Adang et al., 2020) bertujuan untuk memprediksi presentase kekeringan sampel daging kopra guna membedakan kualitas kopra yang masuk dalam kategori kopra standar, bagus, dan sangat bagus. Pada penelitian tersebut, dataset yang digunakan terdiri dari 40 citra yang terbagi menjadi 30 data *training* dan 10 data *testing*. Dan pengujian akurasi yang dilakukan hanya menggunakan metrik akurasi.

Sementara itu, penelitian yang dilakukan oleh penulis lebih berfokus pada klasifikasi jenis kopra, yaitu kopra edible, reguler, dan reject. Jumlah dataset yang digunakan dalam penelitian penulis sebanyak 1.839 citra. Dengan menggunakan jumlah dataset yang lebih besar, penulis memiliki lebih banyak data untuk melatih dan menguji model klasifikasi. Selain itu, penelitian penulis juga menggunakan pendekatan yang lebih variatif dalam membagi dataset menjadi data *training* dan data *testing*. Penulis menggunakan tiga skenario yang berbeda, yaitu 50:50, 70:30, dan 90:10. Hal ini memungkinkan penulis untuk melihat performa model klasifikasi

dalam berbagai proporsi. Selanjutnya, dalam menguji akurasi hasil klasifikasi, penelitian penulis menggunakan metrik akurasi, *confusion matrix*, dan *classification report*. Dengan penggunaan ketiga jenis pengujian tersebut, penelitian penulis dapat memberikan informasi yang lebih rinci mengenai performa model klasifikasi, termasuk *precision*, *recall*, dan *f1-score* untuk setiap kelas.

Penelitian yang dilakukan oleh (Suhendra et al., 2022) telah menggabungkan variasi fitur dan variasi pembagian dataset dalam eksperimennya. Dalam penelitian tersebut, dilakukan eksperimen dengan beberapa variasi fitur, seperti hanya fitur warna, hanya fitur tekstur, dan fitur gabungan antara fitur warna dan tekstur. Pembagian dataset dilakukan dengan rasio 80:20, 70:30, dan 60:40.

Di sisi lain, penelitian penulis juga melakukan eksperimen dengan penggabungan variasi fitur dan pembagian dataset, namun dengan variasi dan pembagian dataset yang berbeda. Variasi fitur yang diuji meliputi kombinasi warna dan bentuk, warna dan tekstur, bentuk dan tekstur, serta kombinasi warna, bentuk, dan tekstur. Pembagian data yang digunakan dalam penelitian ini adalah 50:50, 70:30, dan 90:10. Penelitian (Suhendra et al., 2022) dan penelitian penulis keduanya menguji pengaruh variasi fitur dan pembagian dataset terhadap hasil eksperimen. Dengan menggabungkan variasi fitur dan menggunakan pembagian dataset yang berbeda, diharapkan dapat menghasilkan pemahaman yang lebih komprehensif tentang efektivitas metode yang digunakan dalam analisis data.

Dalam penelitian sebelumnya, metode K-*Nearest Neighbor* (K-NN) telah digunakan untuk melakukan klasifikasi jenis pisang dan mendapat akurasi sebesar 82% (Lestari et al., 2019). Kemudian, (Prayogo et al., 2022) juga melakukan penelitian yang fokus pada pengembangan sistem klasifikasi untuk jenis bunga mawar menggunakan metode K-*Nearest Neighbor* (K-NN). Hasil klasifikasinya menunjukkan akurasi sebesar 92%. Selanjutnya,

penelitian yang dilakukan oleh (Halim & Anraeni, 2021) melakukan klasifikasi pada dataset citra penyakit pneumonia. Penelitian ini mencapai hasil klasifikasi dengan akurasi tinggi, yaitu sekitar 96%. Terakhir, (Rahayu Marlis et al., 2021) melakukan penelitian yang berfokus untuk memprediksi kualitas kopra putih menggunakan metode K-Nearest Neighbor (K-NN) dengan akurasi sebesar 83,3%.

Sementara itu, dalam penelitian penulis, metode *K-Nearest Neighbor* (K-NN) digunakan untuk klasifikasi jenis kopra berdasarkan ekstraksi fitur dan berhasil mencapai akurasi sebesar 85%. Meskipun akurasi penulis hanya mencapai 85%, namun model ini telah berhasil melakukan klasifikasi dengan baik. Perbedaan akurasi ini disebabkan oleh beberapa faktor, antara lain jumlah dataset, pengolahan data, dan penggunaan ekstraksi fitur yang berbeda.



## BAB VI

# PENUTUP

### 6.1 Kesimpulan

Berdasarkan hasil pembahasan dari penelitian yang telah penulis lakukan, berikut merupakan kesimpulan pada penelitian ini:

1. Pada penelitian ini, dilakukan ekstraksi fitur warna, bentuk, dan tekstur dalam analisis data citra. Fitur warna yang diekstraksi meliputi HSV, RGB, dan *grayscale*. Fitur bentuk yang diekstraksi meliputi luas dan *perimeter*, sedangkan fitur tekstur meliputi *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Hasil penelitian menunjukkan bahwa metode tersebut berhasil mencirikan data citra dengan baik. Penggunaan fitur warna, bentuk, dan tekstur memberikan informasi penting dan dapat diandalkan dalam menggambarkan karakteristik setiap citra yang diteliti. Pendekatan ini memberikan kontribusi yang baik dalam proses analisis data citra, dan juga mempersiapkan data citra untuk tahap selanjutnya, yaitu klasifikasi.
2. Klasifikasi jenis kopra menggunakan ekstraksi fitur dengan metode *k-nearest neighbour* menghasilkan akurasi terbaik pada penggunaan tiga fitur yaitu warna, bentuk, dan tekstur dengan pembagian sampel 90:10 yakni sebesar yakni sebesar 85% dengan presisi 85%, recall 84%, dan f1-score 84%.

### 6.2 Saran

Dalam penelitian ini, penulis menyadari masih banyak kekurangan dan keterbatasan yang perlu diperbaiki. Oleh sebab itu berdasarkan penelitian ini, penulis menyarankan untuk mengembangkan penelitian ini menjadi lebih baik dengan beberapa poin sebagai berikut:

1. Penelitian selanjutnya dapat mempertimbangkan penambahan fitur lainnya yang dapat memberikan kontribusi lebih lanjut dalam menggambarkan karakteristik citra.

2. Mencoba metode klasifikasi lainnya untuk membandingkan performa dan akurasi yang lebih tinggi.
3. Menggunakan jumlah data testing yang seimbang untuk masing-masing kelas agar dapat menghindari bias dalam evaluasi performa klasifikasi.
4. Mempertimbangkan penerapan teknik atau algoritma lain untuk meningkatkan kinerja klasifikasi jenis kopra, seperti penggunaan *ensemble learning* atau pengolahan data yang lebih canggih.



## DAFTAR PUSTAKA

- A. S. Abdul Kadir. (2013). Teori dan Aplikasi Pengolahan Citra. Yogyakarta:CV Andi Offset
- Abdullah, A., Usman, U., & Efendi, M. (2017). Sistem Klasifikasi Kualitas Kopra Berdasarkan Warna dan Tekstur Menggunakan Metode Nearest Mean Classifier (NMC). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 4(4), 297–303. <https://doi.org/10.25126/jtiik.201744479>
- Abdullah, & Usman. (2016). Sistem Cerdas untuk Klasifikasi Buah Kelapa Menggunakan Metode Backpropagation. *Selondang Mayang*, 87–94.
- Adang, Y., Rabi, A., & Arifuddin, R. (2020). Klasifikasi Tingkat Kematangan Kopra Menggunakan Metode naïve bayes. *Cyclotron*, 3(1). <https://doi.org/10.30651/cl.v3i1.4307>
- Adi Nugraha, K. (2019). Deteksi Area Parkir Mobil Berbasis Marker Menggunakan Moment Invariants dan K-NN. *Jurnal Teknik Informatika Dan Sistem Informasi*, 5(1), 112–121. <https://doi.org/10.28932/jutisi.v5i1.1586>
- Budi Putranto, B. Y., Hapsari, W., & Wijana, K. (2011). Segmentasi Warna Citra Dengan Deteksi Warna Hsv Untuk Mendeteksi Objek. *Jurnal Informatika*, 6(2). <https://doi.org/10.21460/inf.2010.62.81>
- Farokhah, L. (2020). Implementasi K-Nearest Neighbor untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 7(6), 1129. <https://doi.org/10.25126/jtiik.2020722608>
- Hadi, H. P., & Rachmawanto, E. H. (2022). Analisa Fitur Ekstraksi Ciri Dan Warna Dalam Proses Klasifikasi Kematangan Buah Rambutan Berbasis K-Nearest Neighbor. *Skanika*, 5(2), 177–189. <https://doi.org/10.36080/skanika.v5i2.2944>
- Halim, A. A. D., & Anraeni, S. (2021). Analisis Klasifikasi Dataset Citra Penyakit Pneumonia menggunakan Metode K-Nearest Neighbor (KNN). *Indonesian Journal of Data and Science*, 2(1), 01–12. <https://doi.org/10.33096/ijadas.v2i1.23>
- Heryadi, Y & Edy Irwansyah. (2020). Deep Learning: Aplikasinya di Bidang Geospasial. Depok: PT Artfisia Wahana Informa Teknologi
- Ilyas, M., Syahpawi, & Jonnius. (2022). Analisis Kelayakan Usaha Kopra Putih Di Kabupaten Indragiri Hilir Menurut Perspektif Ekonomi Syariah. *Islamic Business and Finance (IBF)*, 3(2), 158–178.
- Jenis Jenis Kopra, dan Teknik Pengeringan Kopra Asalan, Regular, Edible.* (n.d.). Retrieved May 13, 2023, from <https://rockrealms.com/2023/02/24/jenis-jenis-kopra/>
- Khatib, J., Dalam, S., Satria, B., Sidauruk, A., Wardhana, R., Akbar, A. Al, Ihsan,

- A., Gama, A. M., Yogyakarta, U. A., Bengkulu, U. D., Selatan, P. A., & Kunci, K. (2022). Implementasi Learning Vector Quantization untuk Klasifikasi Jenis Buah Kelapa menggunakan Image Processing. *Indonesian Journal of Computer Science*, 11(1), 566–576.
- Kirana, K. C. (2021). Pengolahan Citra Digital Teori Penerapan Pengolahan Citra Digital pada Deteksi Wajah. Malang: Ahlimedia Book
- Kusumanto, R. D., Tompunu, A. N., & Pambudi, S. (2011). Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV Abstrak. *Jurnal Ilmiah Teknik Elektro*, 2(2), 83–87.
- Panggabean, A. K., Syahfaridzah, A., & Ardiningih, N. A. (2021). Mendeteksi Objek Berdasarkan Warna Dengan Segmentasi Warna Hsv Menggunakan Aplikasi Matlab. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 4(2), 94–97. <https://doi.org/10.46880/jmika.vol4no2.pp94-97>
- Putra, Darma. (2010). Pengolahan Citra Digital (Edisi 1). Yogyakarta: Andi.
- Putri, A. R. (2016). Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya. *JIPI (Jurnal Ilmiah Penulisan Dan Pembelajaran Informatika)*, 1(01), 1–6. <https://doi.org/10.29100/jipi.v1i01.18>
- Rahayu Marlis, R., Abdullah, & Yunita, F. (2021). Sistem Prediksi Kualitas Kopra Putih Menggunakan k-Nearest Neighbor (k-NN). *SISTEMASI: Jurnal Sistem Informasi*, 10(2), 290–299. <http://sistemasi.ftik.unisi.ac.id>
- Rakhmawati, P. U., Pranoto, Y. M., & Setyati, E. (2018). Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur dan Fitur Warna Menggunakan Support Vector Machine. *Seminar Nasional Teknologi Dan Rekayasa (SENTRA)*, 1–8.
- Rilo Pambudi, A., Garno, & Purwantoro. (2020). JIP (Jurnal Informatika Polinema) DETEKSI KEASLIAN UANG KERTAS BERDASARKAN WATERMARK DENGAN PENGOLAHAN CITRA DIGITAL. *Jurnal Informatika Polinema*, 6(4), 69–74.
- Soepomo, P. (2013). SISTEM IDENTIFIKASI CITRA JENIS CABAI (Capsicum Annum L.) MENGGUNAKAN METODE KLASIFIKASI CITY BLOCK DISTANCE. *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)*, 1(2), 409–418.
- Salamah, U. G., & Risma E. (2021). Pengolahan Citra Digital. Bandung: Media Sains Indonesia.
- Susilowati, E. (2018). Konversi Citra RGB Ke Citra HSV Dan HCL Pada Citra Jeruk Medan. *Seminar Nasional Teknologi Informasi Dan Komunikasi STI&K (SeNTIK)*, 2, 67–71.
- Tingkat, U. M., Buah, K., Bintang, B., Shandy, Q., Panna, S. S., & Malago, Y. (2019). Penerapan Metode Grey Level Co-Occurrence Matriks (GLCM) dan

- K-Nearest Neighbor (K-NN). *Jurnal Nasional CosPhi*, 3(1), 2597–9329.
- W, Romy Budhi. (2022). Machine Learning Metode k-Nearest Neighbors Klasifikasi Angka Bahasa Isyarat. Malang: Media Nusa Creative.
- Wuryani, N., & Agustiani, S. (2021). Random Forest Classifier untuk Deteksi Penderita COVID-19 Berbasis Citra CT Scan. *Jurnal Teknik Komputer AMIK BSI*, 7(2), 187–193. <https://doi.org/10.31294/jtk.v4i2>



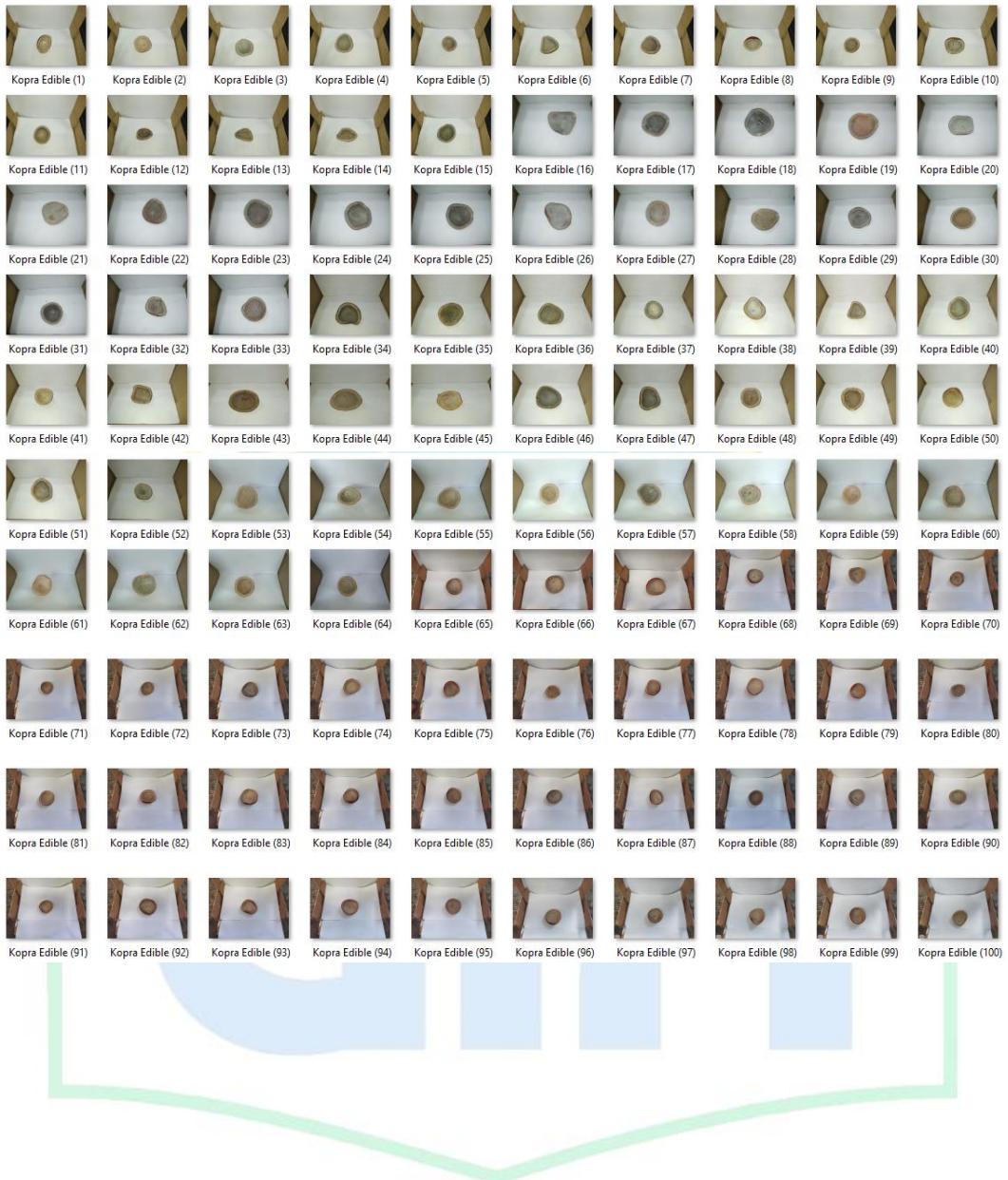
## LAMPIRAN

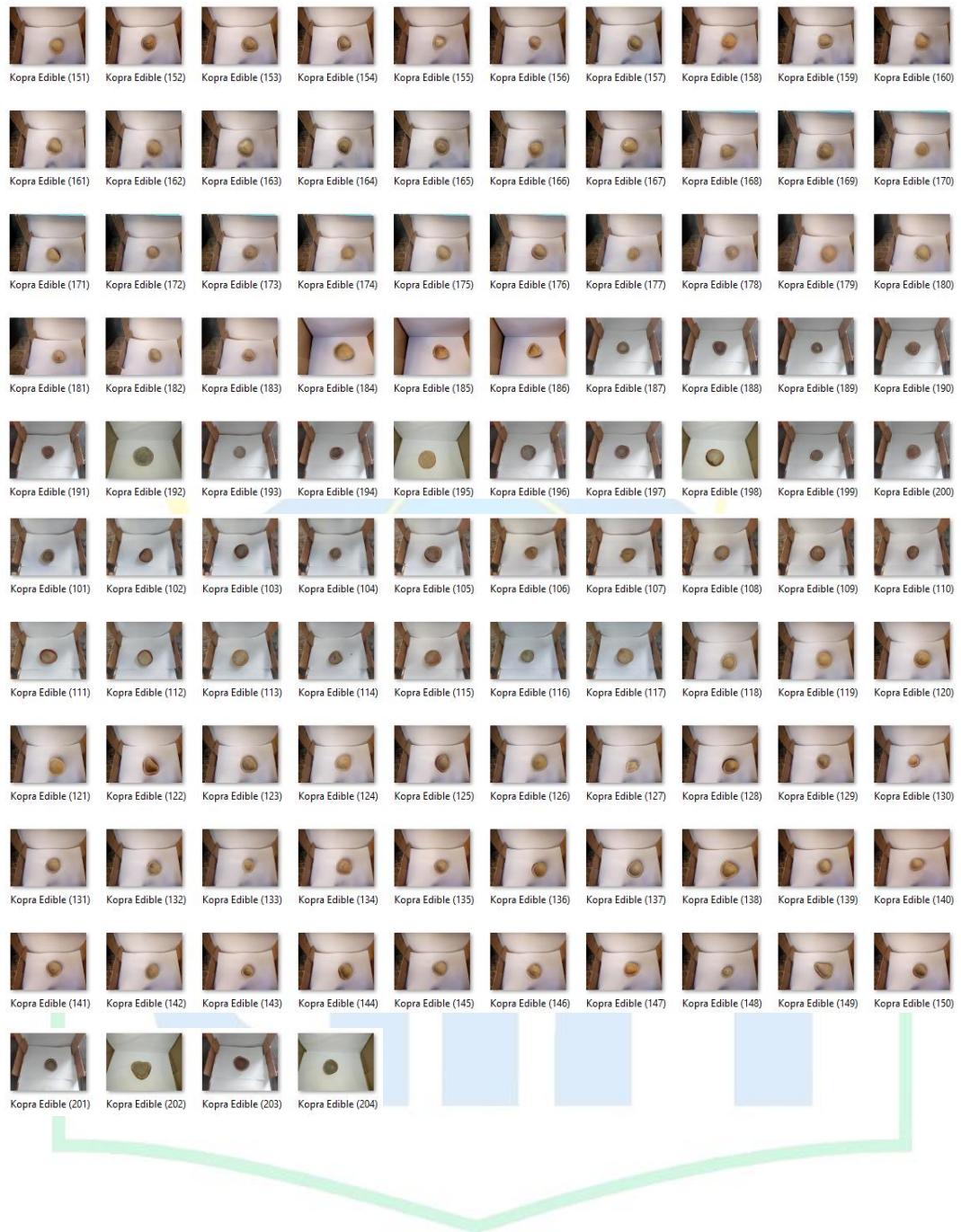
### Lampiran 1 Hasil Wawancara



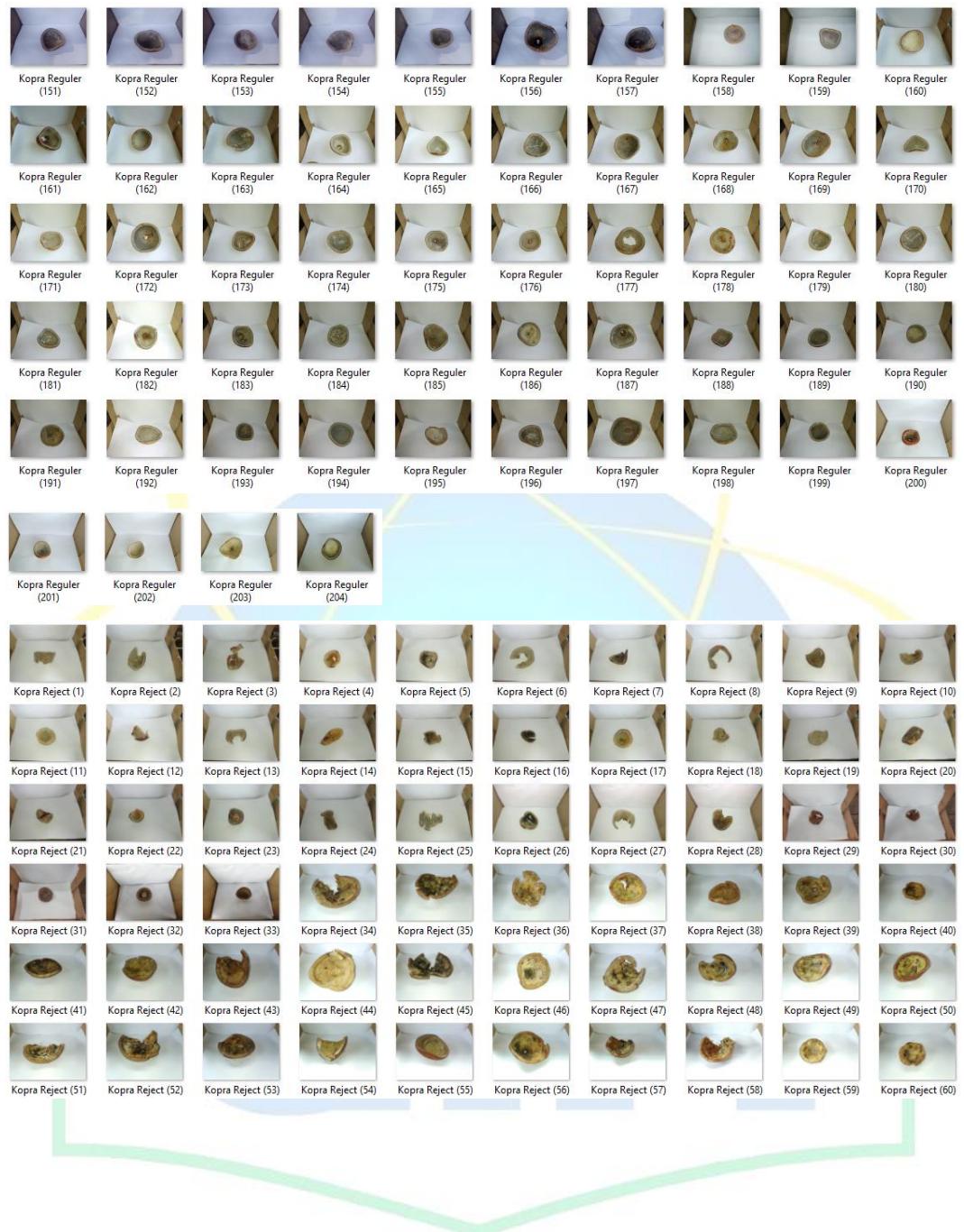
## Lampiran 2

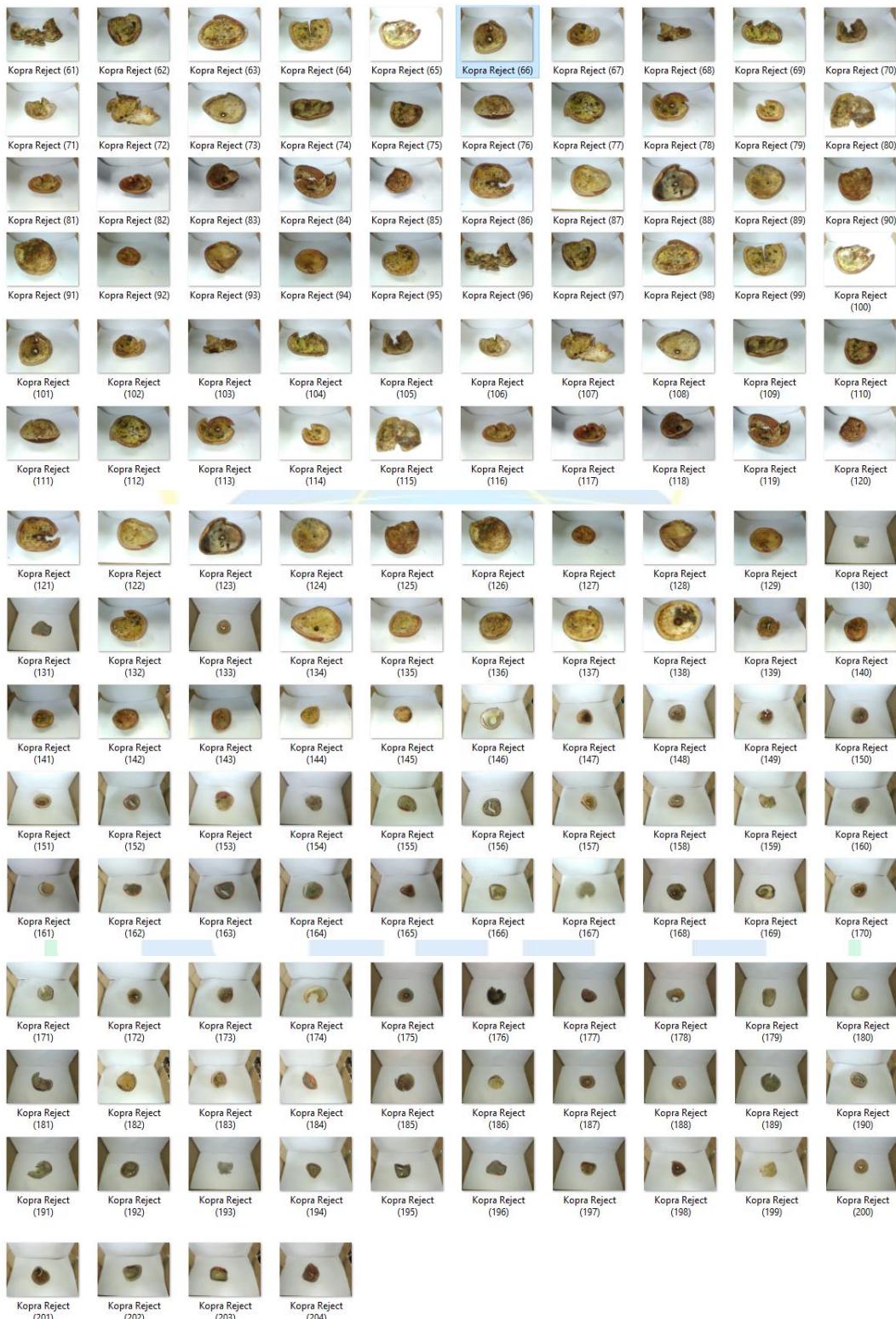
### Dataset Asli







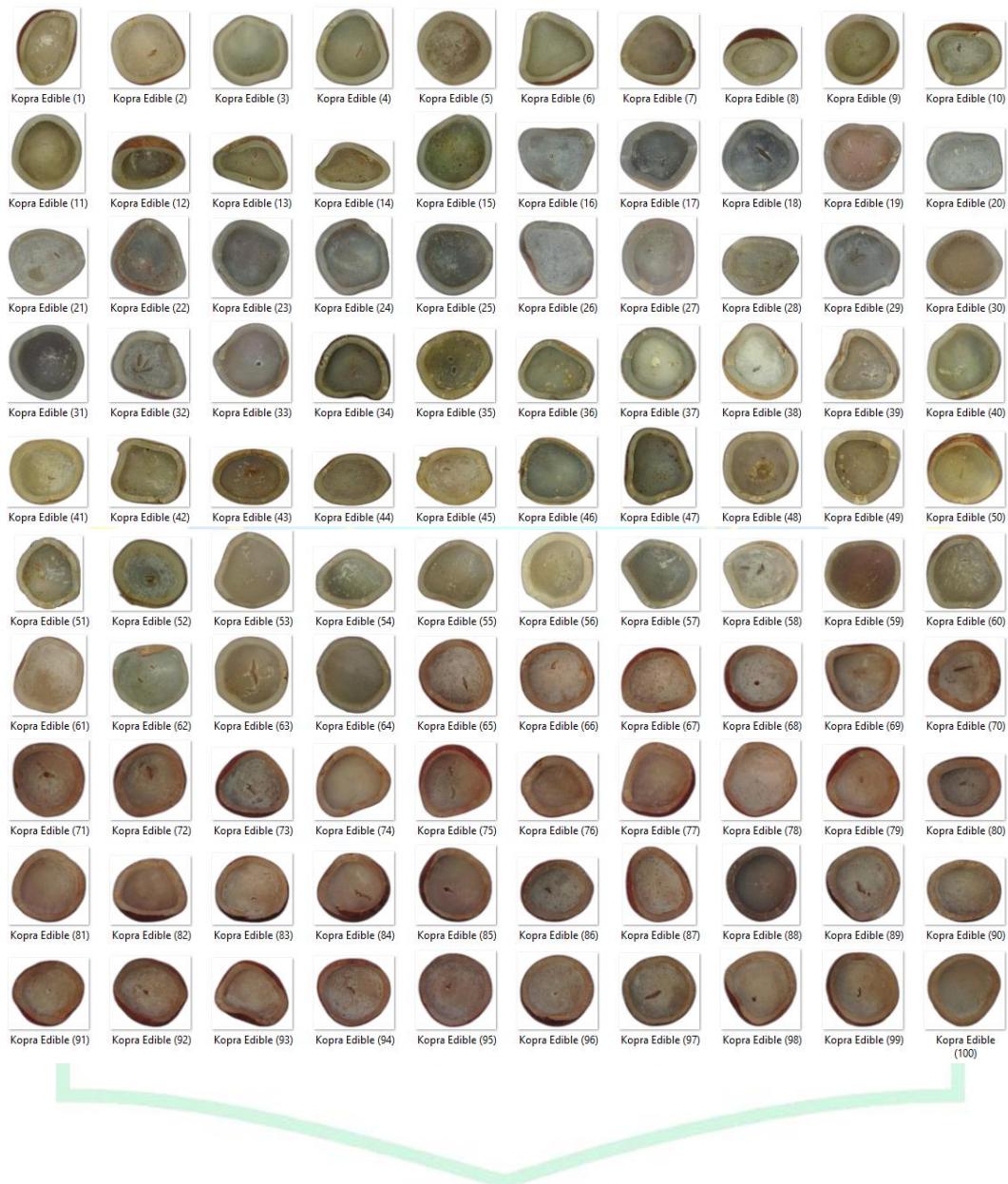




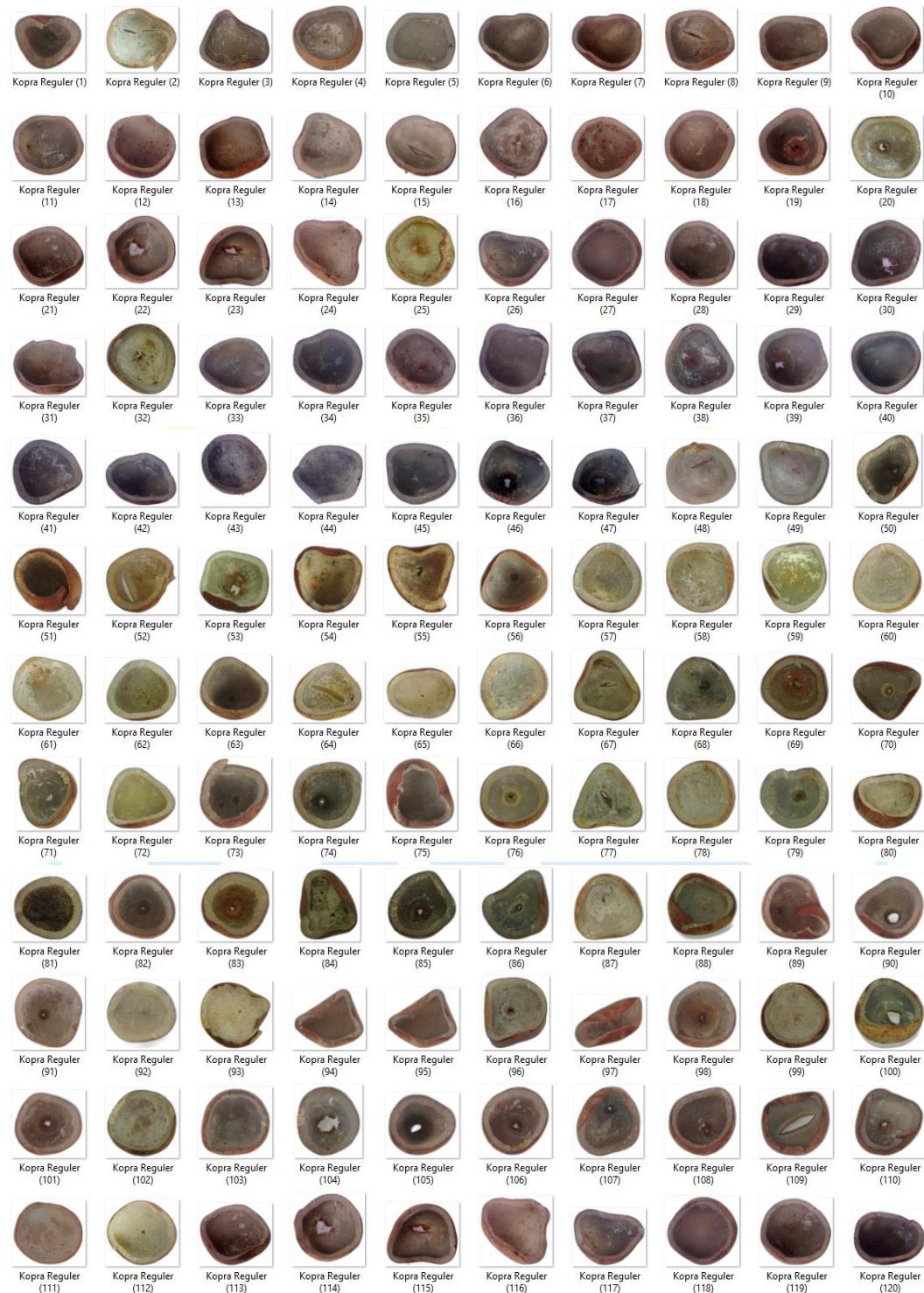
Note: Dataset yang terlampir hanya 1/3 dari dataset yang digunakan.

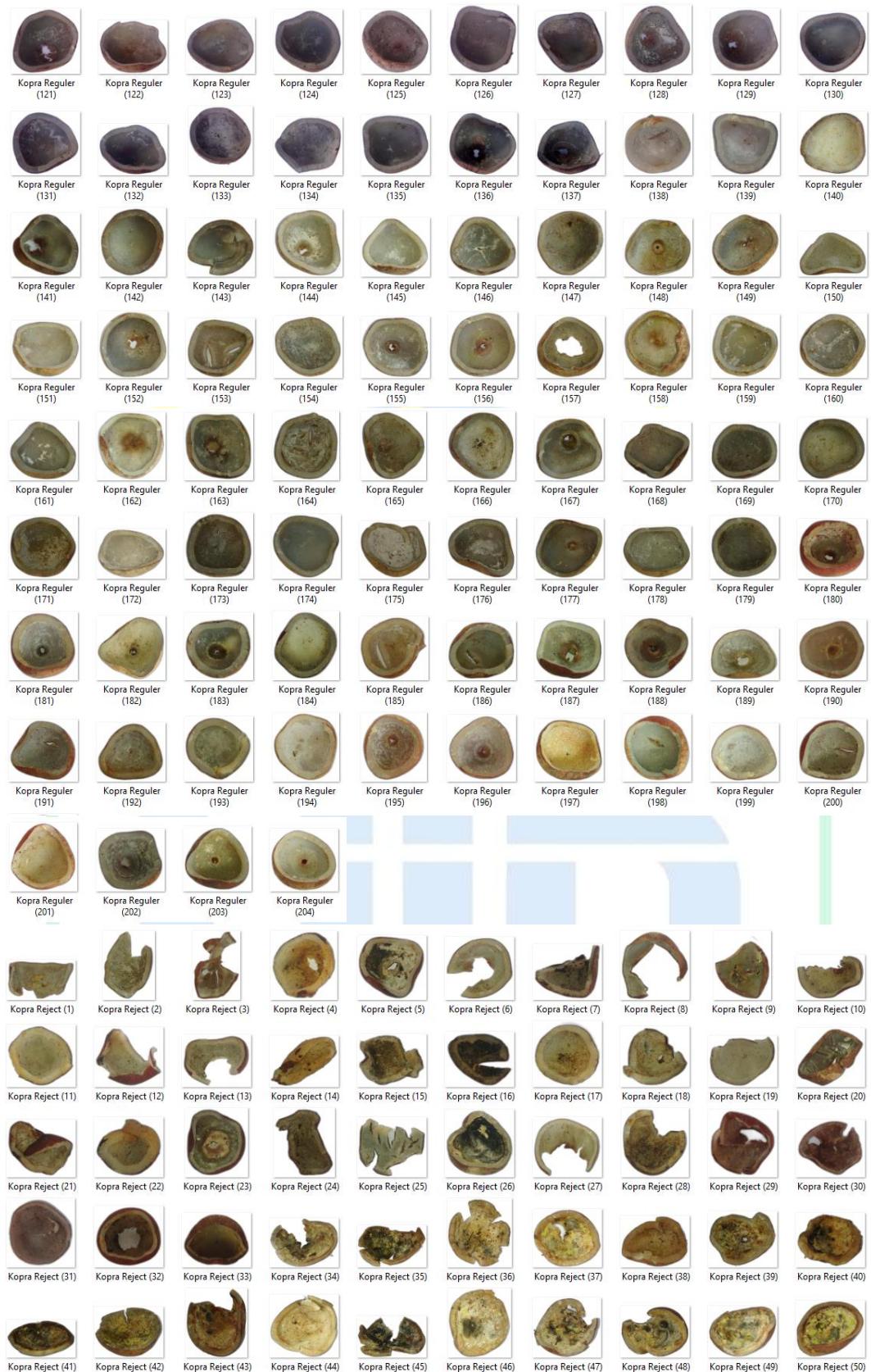
## Lampiran 3

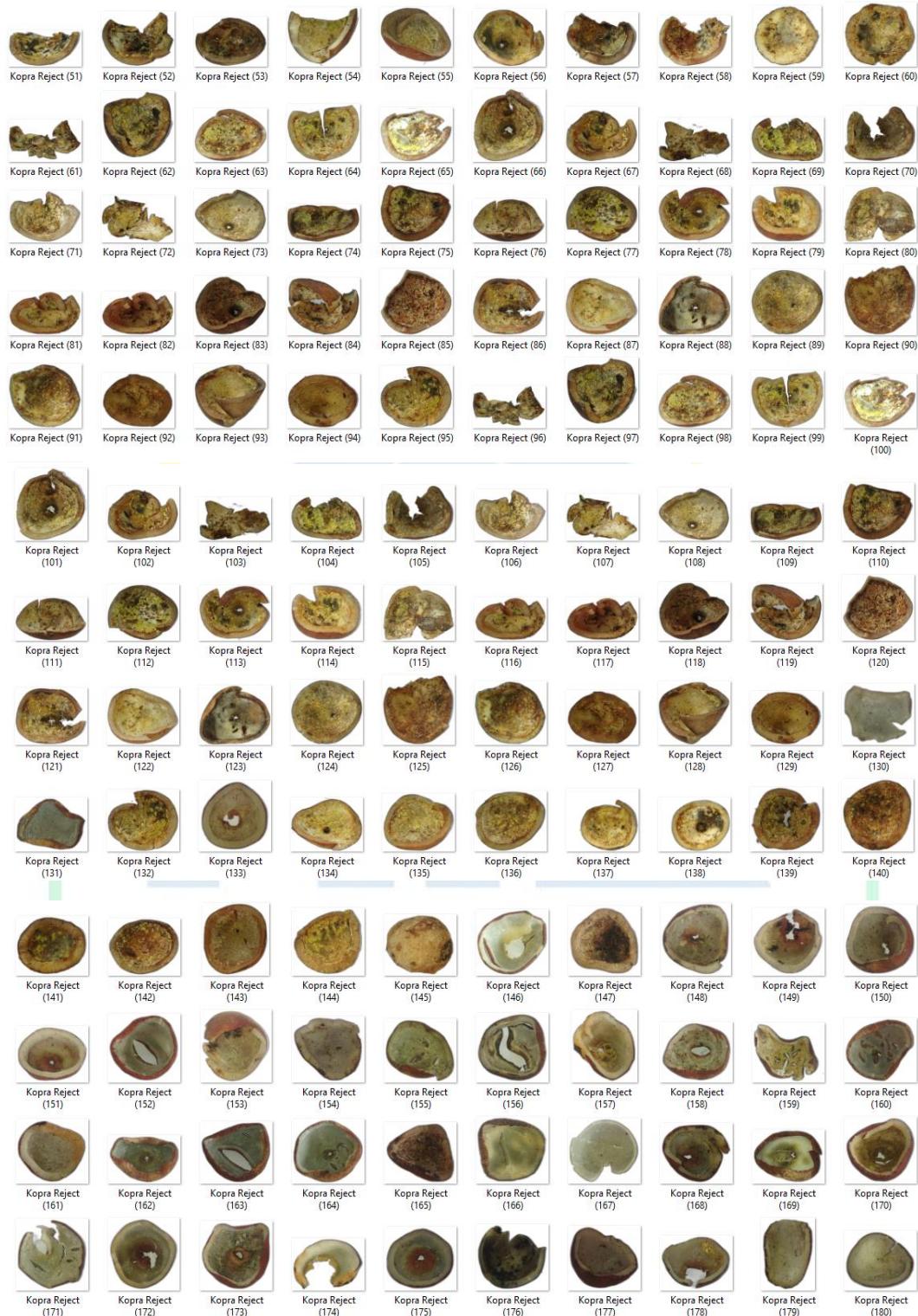
### Dataset setelah *preprocessing*













Note: Dataset yang terlampir hanya 1/3 dari dataset yang digunakan.



## Lampiran 4

### Nilai Hasil Ekstraksi Fitur

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
99,77	85,30	72,58	11,27	99,79	51,02	88,19	54,59	2979,99	7933,50	2053,30	43,45	3,34	0,48	0,23	0,99	1
98,32	96,61	85,83	19,92	98,69	23,79	95,88	63,31	4007,88	44291,00	1093,54	36,60	3,09	0,48	0,25	1,00	1
103,14	98,32	88,33	17,07	103,27	27,66	98,61	59,52	3542,94	47176,00	1126,71	41,85	3,04	0,49	0,21	0,99	1
89,48	80,33	62,88	15,04	89,53	53,26	81,09	55,19	3045,54	8433,00	1808,92	38,49	3,02	0,50	0,26	0,99	1
97,58	96,55	75,91	20,76	98,21	41,71	94,50	67,39	4540,79	17581,50	1164,90	56,45	3,80	0,47	0,25	0,99	1
114,16	110,42	91,15	19,00	114,83	36,87	109,36	79,94	6390,30	33489,00	1334,85	68,71	4,14	0,47	0,28	0,99	1
95,43	93,47	70,49	20,41	95,98	48,33	91,43	61,30	3757,34	35905,00	790,12	51,86	3,25	0,51	0,24	0,99	1
105,50	92,18	59,34	16,19	105,53	80,22	92,41	62,55	3912,56	19637,00	1561,76	62,84	3,97	0,48	0,23	0,99	1
97,58	96,55	75,91	20,76	98,21	41,71	94,50	67,39	4540,79	17581,50	1164,90	56,45	3,80	0,47	0,25	0,99	1
114,16	110,42	91,15	19,00	114,83	36,87	109,36	79,94	6390,30	33489,00	1334,85	68,71	4,14	0,47	0,28	0,99	1
95,43	93,47	70,49	20,41	95,98	48,33	91,43	61,30	3757,34	35905,00	790,12	51,86	3,25	0,51	0,24	0,99	1
105,50	92,18	59,34	16,19	105,53	80,22	92,41	62,55	3912,56	19637,00	1561,76	62,84	3,97	0,48	0,23	0,99	1
99,77	85,30	72,58	11,27	99,79	51,02	88,19	54,59	2979,99	7933,50	2053,30	43,45	3,34	0,48	0,23	0,99	1
98,32	96,61	85,83	19,92	98,69	23,79	95,88	63,31	4007,88	44291,00	1093,54	36,60	3,09	0,48	0,25	1,00	1
103,14	98,32	88,33	17,07	103,27	27,66	98,61	59,52	3542,94	47176,00	1126,71	41,85	3,04	0,49	0,21	0,99	1
89,48	80,33	62,88	15,04	89,53	53,26	81,09	55,19	3045,54	8433,00	1808,92	38,49	3,02	0,50	0,26	0,99	1
80,19	71,38	53,27	14,59	80,19	59,82	71,98	48,88	2389,16	722,50	259,66	39,56	2,85	0,54	0,23	0,99	1
68,42	58,46	43,20	12,79	68,42	66,00	59,72	42,91	1841,25	310,50	233,04	32,04	2,81	0,53	0,26	0,99	1
91,19	79,02	60,32	12,77	91,19	59,09	80,54	57,83	3344,14	21637,00	853,38	49,44	3,30	0,52	0,26	0,99	1
71,67	54,42	41,88	9,40	71,68	77,01	58,16	42,45	1802,04	155,50	139,44	53,78	3,32	0,56	0,23	0,99	1

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
96,60	84,66	67,75	13,04	96,61	53,95	86,30	58,90	3469,06	15980,00	2071,20	38,65	2,74	0,55	0,24	0,99	1
81,81	73,72	55,60	14,65	81,81	56,80	74,07	52,40	2745,51	7269,50	1599,03	53,33	3,26	0,53	0,26	0,99	1
77,52	68,09	49,09	13,56	77,52	62,68	68,76	51,79	2682,59	2963,00	671,35	44,02	3,19	0,51	0,29	0,99	1
106,77	92,93	72,23	13,85	106,77	61,01	94,71	58,79	3456,10	32614,00	1137,89	47,79	3,06	0,55	0,21	0,99	1
96,76	84,13	58,79	14,78	96,77	71,83	85,03	54,81	3004,39	6376,50	2090,64	47,07	3,57	0,44	0,22	0,99	1
82,71	61,56	44,57	10,01	82,71	84,20	65,96	43,96	1932,48	186,00	120,23	46,93	3,48	0,49	0,23	0,99	1
73,76	62,15	47,07	10,90	73,76	57,82	63,91	51,18	2618,92	1942,00	761,01	34,76	2,46	0,62	0,32	0,99	1
95,36	86,87	65,60	15,38	95,36	56,78	86,98	58,87	3465,22	11490,50	2049,73	48,31	3,47	0,48	0,24	0,99	1
93,17	76,05	58,45	11,31	93,17	68,91	79,17	50,84	2584,43	1389,00	536,90	48,83	3,41	0,49	0,21	0,99	1
69,91	65,95	50,10	17,50	69,96	49,08	65,31	46,06	2121,40	124,00	80,08	33,35	2,33	0,64	0,27	0,99	1
82,32	76,85	65,50	15,00	82,37	37,01	77,19	53,20	2830,22	6036,50	974,25	41,49	2,82	0,57	0,24	0,99	1
97,09	85,62	67,37	13,82	97,10	57,35	87,02	54,07	2923,93	36475,00	981,80	45,56	3,04	0,51	0,21	0,99	1
98,13	94,20	76,39	18,72	98,28	41,12	93,33	61,26	3752,97	28938,50	735,80	56,02	3,54	0,50	0,23	0,99	1
104,81	95,95	75,62	14,27	104,81	47,70	96,23	69,64	4849,26	23688,00	629,41	44,96	3,03	0,54	0,27	1,00	1
105,77	94,19	72,45	14,43	105,77	58,78	95,15	61,22	3747,95	31362,50	809,88	40,13	3,10	0,48	0,22	0,99	1
86,34	70,32	50,23	11,58	86,34	72,18	72,83	52,36	2741,54	1307,50	492,26	42,50	3,04	0,53	0,26	0,99	1
96,41	85,07	66,70	13,54	96,41	56,03	86,39	57,67	3325,35	26367,00	637,13	49,67	3,09	0,55	0,23	0,99	1
93,13	84,54	66,55	14,15	93,13	50,48	85,08	58,42	3412,97	14863,00	1821,67	36,25	2,42	0,61	0,26	0,99	1
91,20	81,59	64,20	14,20	91,20	54,30	82,49	54,89	3013,14	5836,00	1478,22	44,15	3,10	0,52	0,22	0,99	1
78,81	63,76	47,52	11,53	78,82	71,92	66,43	46,15	2129,60	880,50	470,32	49,22	3,23	0,51	0,24	0,99	1
82,86	76,94	58,46	16,82	82,87	53,12	76,58	53,89	2903,92	2957,00	542,84	45,10	3,25	0,49	0,24	0,99	1
95,77	91,35	72,04	17,84	95,78	45,73	90,44	60,26	3630,88	18051,50	1475,69	53,12	3,39	0,52	0,23	0,99	1
116,24	105,12	90,78	12,89	116,24	41,74	106,79	67,69	4581,63	32010,00	751,21	68,28	3,81	0,48	0,21	0,99	1

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
79,62	75,05	55,63	17,59	79,69	54,80	74,19	51,40	2642,23	1606,50	521,57	35,64	2,68	0,54	0,24	0,99	1
100,20	90,58	71,28	14,96	100,20	54,79	91,26	58,25	3393,39	16349,00	1530,50	43,19	2,91	0,54	0,21	0,99	1
79,09	71,99	55,98	15,00	79,10	51,24	72,28	52,80	2787,33	3436,50	663,45	55,86	3,48	0,51	0,25	0,99	1
100,95	91,64	74,01	13,70	100,96	45,49	92,42	67,36	4537,89	18117,50	551,29	69,16	3,82	0,51	0,26	0,99	1
86,76	77,10	58,22	14,37	86,76	58,79	77,83	52,09	2712,85	1958,50	573,85	37,74	2,92	0,52	0,24	0,99	1
80,40	71,18	55,91	12,92	80,40	52,58	72,24	52,31	2736,26	2908,00	862,97	34,47	2,61	0,56	0,26	0,99	1
77,13	73,60	62,09	15,74	77,23	32,78	73,31	56,81	3226,82	28582,50	775,13	54,92	3,30	0,55	0,30	0,99	1
84,46	82,08	64,16	19,53	85,28	47,25	80,75	60,34	3640,39	16814,50	1157,61	58,33	3,86	0,46	0,26	0,99	2
98,58	97,93	79,48	23,08	100,64	41,82	96,02	65,94	4348,03	28741,00	2170,54	69,65	4,14	0,43	0,23	0,99	2
93,32	87,80	70,41	19,66	94,33	50,01	87,46	60,17	3620,51	23952,50	1843,71	114,77	6,10	0,37	0,23	0,98	2
114,01	108,95	91,08	19,77	115,31	39,82	108,42	77,19	5958,02	49291,00	1722,69	95,19	5,52	0,40	0,26	0,99	2
77,43	77,73	62,74	24,27	80,02	44,32	75,93	55,40	3068,88	12269,50	1773,34	89,98	5,07	0,42	0,23	0,99	2
88,56	81,71	67,18	16,82	88,63	47,93	82,10	52,10	2714,51	6199,00	1690,41	93,32	5,61	0,37	0,21	0,98	2
95,42	90,63	68,57	20,55	97,05	56,67	89,54	62,81	3944,80	31263,50	1796,82	74,02	4,44	0,44	0,25	0,99	2
110,09	108,55	82,60	21,97	111,53	51,01	106,04	67,68	4580,79	54614,50	1946,13	47,09	3,32	0,50	0,23	0,99	2
69,75	63,97	51,05	17,85	70,62	51,99	64,23	52,88	2796,71	8491,00	1289,25	92,66	4,76	0,47	0,29	0,98	2
75,53	72,26	54,12	19,42	76,27	51,92	71,17	51,85	2688,48	1157,00	549,33	65,82	4,06	0,47	0,27	0,99	2
64,50	60,86	44,96	18,22	65,15	55,33	60,13	44,56	1985,58	376,00	266,59	58,44	3,91	0,48	0,27	0,99	2
108,91	104,28	81,12	19,68	109,08	50,49	103,01	67,71	4583,99	36432,00	865,49	105,89	5,68	0,39	0,22	0,99	2
101,23	98,33	78,37	19,10	101,97	41,25	96,92	71,88	5166,61	31117,50	1152,56	59,62	3,71	0,49	0,28	0,99	2
85,96	83,08	63,45	21,52	86,85	53,87	81,69	53,11	2820,40	11732,00	2944,69	114,78	6,16	0,37	0,21	0,98	2
75,84	70,03	49,68	17,72	75,87	67,66	69,44	50,34	2534,19	9424,50	2296,21	80,04	5,11	0,39	0,24	0,98	2
93,16	85,56	56,73	18,78	93,35	76,02	84,55	55,74	3106,74	12491,00	1385,68	87,66	5,49	0,37	0,22	0,99	2

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
82,52	74,73	54,46	17,15	82,92	61,37	74,76	53,06	2815,78	8657,50	1595,20	76,01	4,73	0,43	0,27	0,99	2
93,96	88,29	69,25	21,10	94,58	54,19	87,80	56,03	3139,32	76104,00	1444,28	93,14	5,19	0,39	0,21	0,99	2
81,38	71,83	50,32	15,64	81,41	71,92	72,25	49,73	2472,88	1173,50	353,75	83,45	4,92	0,40	0,24	0,98	2
84,83	81,14	61,76	19,36	85,27	50,94	80,03	54,63	2984,55	7954,00	2080,48	101,19	5,81	0,39	0,25	0,98	2
98,33	92,87	74,23	19,25	98,44	49,48	92,35	58,70	3445,98	39621,50	915,62	133,00	6,63	0,35	0,20	0,98	2
97,43	88,98	66,26	16,98	97,46	61,45	88,91	56,56	3198,62	13325,50	2305,41	78,80	4,45	0,43	0,22	0,99	2
90,81	85,53	60,69	18,62	90,94	61,84	84,26	57,63	3321,53	39693,00	1243,71	96,01	5,25	0,43	0,24	0,99	2
84,51	78,72	59,95	17,96	84,59	53,56	78,30	54,42	2961,18	9077,50	2314,83	122,86	6,11	0,40	0,25	0,98	2
84,51	78,72	59,95	17,96	84,59	53,56	78,30	54,42	2961,18	9077,50	2314,83	122,86	6,11	0,40	0,25	0,98	2
74,96	71,26	53,51	18,49	75,16	50,74	70,34	51,11	2612,31	1106,00	273,71	63,22	4,16	0,46	0,27	0,99	2
116,45	108,25	81,15	19,13	116,61	60,59	107,60	68,56	4699,96	49032,50	875,54	139,73	6,14	0,38	0,21	0,99	2
70,21	64,74	43,79	20,36	70,39	76,57	64,00	39,41	1553,37	42,50	48,04	79,44	5,27	0,35	0,17	0,97	2
73,70	68,99	48,53	18,97	73,80	66,67	68,06	44,43	1974,43	67,00	88,14	123,44	6,63	0,35	0,21	0,97	2
66,73	59,52	38,69	16,92	66,81	77,29	59,31	42,37	1794,81	40,50	62,53	69,79	4,95	0,40	0,26	0,98	2
84,12	78,53	56,13	18,99	84,31	62,17	77,63	53,92	2907,59	10682,00	2519,93	89,55	5,27	0,41	0,24	0,98	2
72,83	68,06	48,51	19,87	73,09	66,78	67,25	45,21	2044,17	1273,00	549,39	62,96	4,29	0,41	0,22	0,98	2
55,01	48,60	34,20	13,68	55,02	62,69	48,87	39,43	1554,89	0,00	0,00	37,69	3,28	0,49	0,31	0,99	2
55,28	51,17	36,11	17,76	55,30	66,78	50,68	35,62	1268,62	11,00	23,31	44,70	3,93	0,41	0,22	0,98	2
67,78	63,58	40,29	18,83	67,82	75,33	62,19	42,23	1783,13	6,00	21,31	57,63	4,32	0,41	0,22	0,98	2
64,21	55,63	34,17	15,99	64,22	88,29	55,74	37,97	1441,56	1,00	4,00	99,94	5,98	0,37	0,23	0,97	2
70,63	61,07	65,19	100,33	71,29	31,12	64,43	48,32	2334,89	2308,00	566,84	36,41	2,93	0,50	0,24	0,99	2
54,54	48,87	34,04	17,32	54,60	72,76	48,87	32,24	1039,50	0,00	0,00	56,88	4,51	0,38	0,19	0,97	2
63,25	58,89	64,03	90,34	66,19	23,71	60,78	43,90	1926,97	566,00	241,14	31,51	2,75	0,53	0,24	0,99	2

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
58,35	51,34	57,85	98,12	60,62	30,89	54,18	42,01	1764,64	102,50	96,33	43,89	3,55	0,47	0,26	0,99	2
57,51	51,38	57,44	98,36	60,16	31,61	53,90	42,52	1807,93	526,50	373,40	30,74	2,71	0,54	0,25	0,99	2
80,95	72,68	54,48	15,51	80,96	61,69	73,07	51,30	2632,17	7381,50	2354,55	80,29	5,29	0,38	0,24	0,98	2
63,55	55,72	62,72	95,69	65,57	29,55	58,85	44,06	1941,40	36,50	112,81	53,31	4,09	0,45	0,29	0,99	2
69,62	64,57	71,21	95,18	72,60	21,73	66,83	49,19	2419,35	3850,00	942,00	60,27	4,31	0,44	0,26	0,99	2
63,35	58,59	62,10	93,66	64,71	19,64	60,40	41,24	1700,73	63,50	48,87	28,99	2,68	0,51	0,21	0,99	2
63,92	57,46	41,89	15,60	63,93	63,32	57,60	41,44	1717,31	95,00	85,25	82,03	5,05	0,42	0,25	0,98	2
50,77	45,24	47,41	73,27	52,60	38,65	47,14	37,87	1434,19	239,00	110,23	52,36	3,74	0,48	0,23	0,98	2
51,17	42,16	45,62	86,85	51,91	41,49	45,26	36,95	1365,48	38,00	55,46	56,48	4,00	0,46	0,26	0,98	2
59,59	52,50	34,52	16,93	59,69	77,83	52,57	36,08	1301,75	109,00	138,37	43,85	3,43	0,47	0,24	0,98	2
76,13	72,77	49,92	20,36	76,21	65,32	71,19	44,88	2014,41	96,50	90,33	67,43	4,63	0,42	0,22	0,98	2
87,16	64,11	33,57	12,81	87,17	115,27	67,54	49,79	2479,14	1798,00	944,92	199,73	8,65	0,33	0,24	0,96	3
84,66	61,79	30,77	13,55	84,66	124,96	65,11	40,91	1673,97	33,00	43,46	120,71	6,82	0,33	0,19	0,96	3
101,33	79,78	36,79	14,35	101,34	112,41	81,34	58,71	3446,83	10579,00	2258,28	169,17	7,91	0,36	0,26	0,98	3
105,76	82,49	46,49	13,43	105,77	102,10	85,35	61,54	3786,86	14842,00	1745,41	185,33	8,74	0,32	0,24	0,98	3
85,47	79,17	60,33	18,17	85,80	58,35	78,91	58,32	3401,36	4135,00	752,32	151,60	6,64	0,37	0,22	0,98	3
53,36	46,75	32,07	15,00	53,38	72,30	47,08	33,69	1135,18	0,00	0,00	60,08	4,64	0,40	0,25	0,97	3
100,44	80,29	50,12	13,58	100,49	91,33	82,89	56,94	3242,34	14771,50	3009,01	108,93	6,15	0,37	0,24	0,98	3
60,03	50,41	33,22	14,58	60,06	82,66	51,33	38,77	1503,37	79,50	64,53	86,96	5,26	0,39	0,23	0,97	3
74,89	64,77	50,31	15,49	75,06	65,93	66,16	50,87	2587,51	4458,00	267,76	98,10	5,30	0,41	0,21	0,98	3
60,10	51,42	36,99	11,16	60,10	57,96	52,38	46,43	2155,96	51,50	41,21	58,46	3,96	0,50	0,34	0,99	3
65,70	53,85	38,52	13,25	65,72	78,25	55,65	38,13	1453,99	274,50	94,81	55,97	3,63	0,45	0,21	0,98	3
56,11	52,09	39,02	20,33	57,66	63,26	51,81	35,67	1272,11	513,00	153,82	87,99	5,20	0,37	0,21	0,97	3

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
57,39	46,50	29,44	12,42	57,39	83,30	47,83	37,25	1387,58	150,00	172,85	76,82	4,53	0,43	0,27	0,97	3
66,60	58,95	41,09	14,98	66,61	68,81	59,21	41,05	1684,93	342,00	171,05	68,50	4,40	0,43	0,24	0,98	3
58,94	49,47	30,53	13,44	58,94	82,28	50,14	39,01	1521,41	26,00	75,11	56,87	4,15	0,45	0,27	0,98	3
64,10	55,72	39,17	16,76	64,23	73,03	56,35	36,62	1341,16	124,00	61,94	89,76	5,54	0,36	0,20	0,97	3
90,74	73,37	46,53	13,39	90,74	89,51	75,54	52,56	2762,39	3703,00	994,17	78,53	4,90	0,41	0,23	0,99	3
56,83	42,71	30,24	11,26	56,90	89,69	45,51	36,98	1367,20	1123,00	130,91	58,44	3,68	0,45	0,23	0,98	3
43,92	36,17	27,44	9,60	43,96	51,60	37,49	40,12	1610,01	916,50	175,44	91,40	4,57	0,53	0,41	0,97	3
91,33	78,26	52,79	15,33	91,37	80,79	79,29	54,54	2974,91	2639,00	327,85	90,97	5,25	0,39	0,20	0,98	3
66,72	61,94	48,57	13,55	66,73	41,45	61,86	52,77	2784,60	127,00	113,88	82,22	4,57	0,48	0,34	0,99	3
54,23	51,13	37,63	13,12	54,28	38,32	50,52	53,12	2821,37	362,50	189,58	43,84	2,93	0,60	0,45	0,99	3
57,37	49,16	34,75	13,21	57,37	66,45	49,99	38,39	1473,86	67,00	61,60	62,11	3,90	0,46	0,27	0,98	3
44,33	33,62	23,35	10,25	44,33	76,69	35,64	29,98	898,70	39,00	28,14	48,49	3,63	0,48	0,31	0,97	3
73,32	63,04	45,19	14,33	73,33	72,05	64,09	46,17	2131,38	880,50	353,40	74,04	4,94	0,38	0,22	0,98	3
50,69	41,78	27,28	10,51	50,69	62,09	42,80	41,23	1700,28	3,00	8,83	41,56	3,04	0,57	0,41	0,99	3
46,20	36,37	24,94	11,96	46,27	87,34	38,01	34,91	1218,72	1415,00	201,14	57,46	3,84	0,44	0,25	0,98	3
60,66	51,07	37,88	13,83	60,73	71,53	52,43	35,98	1294,31	14,50	23,56	76,16	5,28	0,36	0,20	0,97	3
65,72	55,40	37,07	12,80	65,72	74,77	56,41	43,70	1910,00	27,50	42,04	48,11	3,78	0,47	0,28	0,99	3
52,52	47,49	35,97	13,84	52,52	52,71	47,67	38,47	1480,06	1,00	4,00	39,48	3,25	0,50	0,30	0,99	3
70,51	59,03	44,31	12,61	70,51	68,89	60,79	41,44	1716,94	44,00	30,97	64,65	4,37	0,42	0,22	0,98	3
64,85	53,42	38,14	11,99	64,88	72,91	55,10	42,62	1816,11	365,50	95,84	67,16	4,35	0,44	0,26	0,98	3
79,37	68,17	44,72	15,27	79,44	81,03	68,85	51,04	2604,81	3664,00	532,84	73,85	4,75	0,40	0,21	0,99	3
83,70	77,19	49,50	18,19	83,73	76,18	75,97	52,16	2721,12	2218,00	1040,23	57,30	4,29	0,42	0,23	0,99	3
79,16	68,28	47,81	13,41	79,16	70,36	69,22	51,73	2676,31	1882,50	416,17	72,55	4,45	0,44	0,26	0,99	3

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-S	Mean-Gray	Standar-Deviasi	Variasi	Luas	Perimeter	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	Jenis Kopra
62,29	56,50	46,61	14,88	62,45	47,41	57,10	45,66	2084,78	2242,00	191,97	58,93	3,84	0,47	0,27	0,99	3
73,45	66,74	42,73	16,13	73,46	70,84	66,02	51,08	2609,50	885,50	545,23	83,38	5,32	0,40	0,26	0,98	3
60,78	48,47	26,84	13,73	60,83	93,58	49,69	38,25	1462,92	36,00	43,80	57,19	4,53	0,41	0,26	0,98	3
52,70	44,42	31,07	13,65	52,72	78,69	45,38	37,34	1394,07	530,50	102,47	104,86	4,71	0,44	0,23	0,96	3
54,11	43,85	36,30	9,71	54,12	54,21	46,06	39,67	1573,82	86,50	54,04	49,31	3,86	0,46	0,30	0,98	3
62,98	47,41	38,45	8,26	62,98	76,50	51,04	49,83	2482,81	3695,50	572,54	76,96	4,87	0,44	0,31	0,98	3
81,23	62,19	52,67	8,15	81,23	65,72	66,80	65,59	4302,10	8645,50	1337,55	99,29	4,99	0,47	0,32	0,99	3
68,89	42,68	31,14	6,84	68,89	100,64	49,21	45,38	2059,34	2188,00	649,07	63,77	4,14	0,47	0,29	0,98	3
78,65	62,02	52,75	7,40	78,65	61,91	65,96	59,21	3505,36	8432,00	1729,78	77,17	4,41	0,46	0,29	0,99	3
88,74	61,27	50,86	8,04	88,74	89,83	68,30	57,12	3262,97	26424,50	746,22	97,96	5,41	0,38	0,21	0,98	3
62,67	35,24	27,06	6,30	62,67	107,74	42,54	42,12	1774,41	281,50	173,92	105,78	5,36	0,44	0,26	0,97	3
62,29	46,66	36,59	8,02	62,29	75,18	50,18	48,00	2304,36	2164,50	674,62	85,81	4,92	0,44	0,30	0,98	3
83,36	60,98	50,99	11,04	83,36	79,24	66,55	57,34	3287,56	4823,50	1350,04	87,63	4,99	0,43	0,24	0,99	3
72,16	53,52	44,02	6,81	72,16	80,76	58,01	59,01	3482,16	6709,50	1279,41	74,22	4,34	0,46	0,28	0,99	3
78,54	56,22	43,80	10,46	78,54	95,82	61,50	60,16	3619,14	29255,50	716,14	93,72	4,89	0,43	0,23	0,99	3

Note: Nilai ekstraksi fitur yang terlampir hanya 1/12 dari dataset yang digunakan, terdapat 50 citra untuk masing-masing jenis kopra.