

DOKUMEN PROYEK
12S3205 – PENAMBANGAN DATA

**DiTenun – *Ulos Image Classification Using Convolutional
Neural Networks (CNN)***

Disusun Oleh:

Kelompok 10

12S21047 Elshaday Prida Simamora
12S21048 Nussy Pentasonia Pangaribuan
12S21049 Jesika Audina Purba



PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
TAHUN 2024/2025

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR TABEL.....	4
DAFTAR GAMBAR	6
BAB 1 PENDAHULUAN	7
1.1 Latar Belakang	7
1.2 Tujuan.....	8
1.3 Manfaat.....	8
1.4 Ruang Lingkup.....	9
BAB 2 STUDI LITERATUR	11
2.1 <i>Convolutional Neural Networks</i> (CNN)	11
2.2 Penerapan CNN dalam Klasifikasi Gambar	11
BAB 3 METODE PENELITIAN	13
3. 1 <i>Cross-Industry Standard Process for Data Mining</i> (CRISP-DM).....	13
3.1.1 <i>Business Understanding</i>	13
3.1.2 <i>Data Understanding</i>	13
3.1.3 <i>Data Preparation</i>	73
3.1.4 <i>Modeling</i>	82
3.1.6 <i>Deployment</i>	94
3.2 <i>Timeline</i>	101
BAB 4 HASIL PENGUJIAN.....	103
4.1 Hasil Pelatihan Model CNN	103
BAB 5 ANALISIS	105
5.1 Analisis Hasil Evaluasi Model.....	105
5.1.1 Akurasi Model	105
5.1.2 <i>Precision, Recall, dan F1-Score</i>	105
5.2 Kelebihan Model CNN	106
5.2.1 Kemampuan untuk Mendeteksi Fitur Visual	106
5.2.3 Fleksibilitas dan Skalabilitas.....	106
5.2.4 Kemudahan Implementasi	106

5.3 Keterbatasan Model CNN.....	106
5.3.1 Kualitas Gambar	106
5.3.2 Variasi dalam Motif	107
5.3.3 Waktu Latih dan Sumber Daya.....	107
5.3.4 <i>Overfitting</i>	107
5.4 Saran untuk Penelitian Selanjutnya	107
BAB 6 KESIMPULAN.....	108
DAFTAR REFERENSI	109
BAB 7 PEMBAGIAN TUGAS	110

DAFTAR TABEL

Tabel 3-1 Kode Program untuk Mengumpulkan Data.....	14
Tabel 3-2 Kode Program Untuk Menampilkan Jumlah Data	14
Tabel 3-3 Jumlah Data	15
Tabel 3-4 Kode Program untuk Menampilkan Deskripsi Data	16
Tabel 3-5 Deskripsi Data	17
Tabel 3-6 Kode Program untuk Menampilkan Karakteristik Data	50
Tabel 3-7 Karakteristik Data.....	52
Tabel 3-8 Kode Program untuk Menampilkan Keterkaitan Antar Data	53
Tabel 3-9 Keterkaitan Antar Data.....	57
Tabel 3-10 Kode Program untuk Menampilkan Ukuran Data.....	58
Tabel 3-11 Ukuran Data.....	60
Tabel 3-12 Kode Program untuk Menampilkan Deskripsi Statistik Atribut Data	61
Tabel 3-13 Deskripsi Statistik Atribut Data.....	63
Tabel 3-14 Kode Program untuk Menampilkan Relasi Antar Atribut Data	64
Tabel 3-15 Relasi Antar Atribut Data.....	66
Tabel 3-16 Kode Program untuk Menampilkan Visualisasi Data	68
Tabel 3-17 Visualisasi Data	70
Tabel 3-18 Kode untuk Memilih dan Memilah Data.....	73
Tabel 3-19 Kode Program untuk Membersihkan Data	75
Tabel 3-20 Kode Program untuk Mengkonstruksi Data	78
Tabel 3-21 Kode Program untuk Mengintegrasikan Data	80
Tabel 3-22 Kode Program untuk Membangun Arsitektur CNN.....	82

Tabel 3-23 Kode Program untuk Mendefinisikan Arsitektur Model CNN	84
Tabel 3-24 Kode Program untuk Melatih Model CNN	85
Tabel 3-25 Kode Program untuk Melatih Model CNN (2).....	85
Tabel 3-26 Kode Program untuk Memuat Data Uji.....	86
Tabel 3-27 Kode Program untuk Membuat Plot Hasil Data Uji.....	86
Tabel 3-28 Kode Program untuk Menyaring Klasifikasi untuk Gambar Ulos	88
Tabel 3-29 Hasil Evaluasi Model	89
Tabel 3-30 Kode Program Untuk Membuat Matriks Hasil Akhir	90
Tabel 3-31 Kode Program untuk Menyimpan Model.....	94
Tabel 3-32 Kode Program untuk Membangun Streamlit.....	95
Tabel 3-33 <i>Timeline</i>	101

DAFTAR GAMBAR

Gambar 3-1 Grafik Hasil Test Anova.....	57
Gambar 3-2 Grafik Heatmap Korelasi Antar Atribut Numerik	66
Gambar 3-3 Grafik <i>Cross-Tabulation</i> Antara Kategori dan Format File	67
Gambar 3-4 Barchart Jumlah File per Kategori.....	70
Gambar 3-5 Doughnut Chart Distribusi Format File.....	71
Gambar 3-6 Gabungan Barchart dan Line Chart Distribusi Ukuran File	72
Gambar 3-7 Grafik Boxplot Ukuran File.....	72
Gambar 3-8 Hasil Plot Data Uji.....	87
Gambar 3-9 Confusion Matrix.....	92
Gambar 3-10 Grafik Heatmap	93

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai negara yang kaya akan seni dan budaya. Salah satu unsur budaya tersebut adalah menenun. Tenun merupakan kerajinan tangan berupa kain yang dibuat dari benang (katun, sutra, dan lain-lain) dengan cara memasukkan benang secara melintang pada suatu alat dan dibuat di daerah seperti Sumatera, Jawa, Palembang, dan Kalimantan [1]. Dengan berdasar kepada produk tenun dari berbagai daerah, penelitian ini mengkaji hasil tenun daerah Sumatera Utara, yaitu Ulos. Terdapat sebuah organisasi DiTenun yang didedikasikan untuk pelestarian dan promosi kain tenun tradisional Indonesia, khususnya Ulos, sebagai warisan budaya masyarakat Batak. Motif ulos yang digunakan oleh penenun saat ini adalah motif ulos yang lalu dan masih digunakan hingga saat ini [1]. Aplikasi Ditenun memiliki beberapa modul *generate*, yaitu modul Generate Kristik dan modul Ulos Editor. Terdapat juga model Verifikasi Kualitas Gambar dan Model Klasifikasi Ulos yang akan diintegrasikan dengan aplikasi DiTenun sebagai modul baru [2].

Untuk itu, perlu ada upaya untuk melestarikan dan mengenalkan lebih luas lagi keanekaragaman budaya Indonesia, salah satunya melalui pemanfaatan teknologi. Proyek DiTenun – *Ulos Image Classification* bertujuan untuk mengembangkan sebuah sistem yang dapat mengklasifikasikan jenis-jenis kain Ulos berdasarkan gambar, dengan memanfaatkan teknologi *machine learning* dan *image processing* menggunakan algoritma *Convolutional Neural Networks* (CNN). Algoritma CNN adalah salah satu algoritma yang mirip dengan arsitektur jaringan saraf tiruan yang telah terbukti sangat efektif dalam berbagai tugas pengolahan citra, termasuk pengenalan objek, pengenalan wajah, dan klasifikasi gambar. Algoritma ini bekerja dengan cara mengekstraksi fitur-fitur penting dari gambar melalui lapisan konvolusi, yang kemudian diproses untuk menghasilkan prediksi tentang isi gambar tersebut. Keunggulan algoritman CNN terletak pada kemampuannya untuk belajar secara otomatis mengenali pola-pola visual yang ada dalam gambar tanpa memerlukan intervensi manusia yang signifikan dalam hal pemrograman fitur. Oleh karena itu, algoritma CNN menjadi pilihan yang ideal untuk proyek DiTenun – *Ulos Image Classification* ini.

Melalui proyek ini, kami akan mengembangkan sebuah sistem yang dapat mengenali dan mengklasifikasikan berbagai jenis ulos Batak berdasarkan gambar. Sistem ini akan dilatih menggunakan data gambar ulos yang sudah diberi label sesuai dengan jenis dan motifnya. Dengan menggunakan algoritma CNN, sistem diharapkan dapat mengenali berbagai ciri khas pada kain ulos. Keberhasilan sistem ini tidak hanya akan mempermudah proses identifikasi ulos, tetapi juga akan berkontribusi pada pelestarian budaya, di mana masyarakat, peneliti, dan pelestari budaya dapat lebih mudah mengakses informasi mengenai ulos dan mengidentifikasi jenis-jenisnya.

1.2 Tujuan

1. Membuat Sistem Klasifikasi Gambar Ulos

Tujuan utama dari proyek ini adalah untuk membuat sebuah sistem yang dapat mengidentifikasi dan mengklasifikasikan berbagai jenis kain ulos Batak hanya dengan melihat gambar. Sistem ini menggunakan algoritma CNN yang memungkinkan komputer untuk "belajar" mengenali berbagai ciri khas dari setiap jenis ulos, seperti motif dan warna.

2. Meningkatkan Keakuratan dan Kecepatan Klasifikasi

Klasifikasi ulos secara manual membutuhkan banyak waktu dan pengetahuan mendalam. Dengan menggunakan CNN, proses ini menjadi jauh lebih cepat dan akurat. Sistem ini bisa mengenali berbagai motif ulos, bahkan yang cukup rumit, dengan tingkat ketepatan yang tinggi, sehingga membuat proses identifikasi lebih efisien dan dapat dilakukan dengan lebih mudah oleh siapa saja.

3. Mempermudah Akses dan Melestarikan Budaya Menggunakan Teknologi

Proyek ini juga bertujuan untuk menunjukkan bagaimana teknologi modern, seperti algoritma CNN, dapat digunakan untuk melestarikan dan mendokumentasikan warisan budaya. Kami berharap ini bisa menjadi inspirasi bagi penggunaan teknologi dalam melestarikan budaya-budaya lainnya di Indonesia, serta memotivasi riset-riset lebih lanjut di bidang ini.

1.3 Manfaat

1. Mempermudah Identifikasi Ulos

Salah satu manfaat terbesar dari proyek ini adalah mempermudah proses identifikasi berbagai jenis ulos. Dengan sistem yang dapat mengklasifikasikan ulos hanya dari gambar,

orang yang tidak memiliki pengetahuan khusus tentang ulos bisa dengan mudah mengetahui jenis dan makna kain tersebut. Ini akan sangat membantu dalam konteks edukasi, pameran budaya, atau bahkan bagi wisatawan yang tertarik mempelajari lebih lanjut tentang budaya Batak.

2. Meningkatkan Akses Pengetahuan tentang Ulos

Proyek ini memungkinkan siapa saja, di mana saja, untuk mengakses informasi tentang ulos dengan mudah. Misalnya, peneliti, pelestari budaya, atau bahkan masyarakat umum yang ingin mempelajari ulos dapat menggunakan sistem ini sebagai referensi cepat. Ini juga dapat digunakan di museum, galeri, atau tempat wisata sebagai alat bantu edukasi yang menarik.

3. Mengurangi Ketergantungan pada Ahli untuk Klasifikasi Manual

Sebelumnya, untuk mengenali jenis ulos, dibutuhkan keahlian khusus dari orang yang sudah berpengalaman dalam tenun ulos. Dengan adanya sistem berbasis CNN, kita tidak lagi perlu bergantung pada ahli untuk setiap klasifikasi. Sistem ini memungkinkan identifikasi yang lebih cepat, efisien, dan dapat diakses oleh lebih banyak orang tanpa batasan keahlian.

1.4 Ruang Lingkup

1. Koleksi dan Pengumpulan Data Gambar Ulos

Proyek ini akan mencakup pengumpulan gambar ulos dari berbagai jenis dan motif yang ada dalam budaya Batak. Gambar-gambar ini akan digunakan sebagai dataset untuk melatih algoritma CNN. Data yang dikumpulkan akan mencakup representasi dari berbagai jenis ulos yang sering digunakan.

2. Pengembangan Algoritma CNN untuk Klasifikasi

Fokus utama proyek ini adalah pengembangan dan pelatihan algoritma CNN untuk mengklasifikasikan jenis-jenis ulos berdasarkan gambar. Model akan dilatih dengan menggunakan dataset yang telah didapatkan untuk memastikan akurasi dalam mengenali berbagai jenis ulos berdasarkan fitur visualnya.

3. Evaluasi dan Pengujian Akurasi Sistem

Setelah model dikembangkan, tahap selanjutnya adalah evaluasi performa model melalui pengujian dengan data yang tidak digunakan dalam pelatihan. Pengujian ini bertujuan

untuk mengukur tingkat akurasi sistem dalam mengklasifikasikan ulos dengan benar. Hasil pengujian akan digunakan untuk memperbaiki dan menyempurnakan model agar lebih akurat.

4. Implementasi Sistem Klasifikasi

Sistem yang telah dikembangkan akan diimplementasikan dalam bentuk aplikasi atau platform berbasis web yang memungkinkan pengguna untuk mengunggah gambar ulos dan mendapatkan hasil klasifikasi secara otomatis. Sistem ini juga akan dilengkapi dengan penjelasan tentang jenis ulos yang terdeteksi beserta makna simbolisnya.

5. Penyusunan Dokumentasi

Sebagai bagian dari proyek, akan disusun dokumentasi yang menjelaskan mengenai proses pembuatan model pengklasifikasian ulos dengan algoritma CNN.

BAB 2

STUDI LITERATUR

2.1 *Convolutional Neural Networks* (CNN)

Convolutional Neural Networks (CNN) menjadi salah satu pendekatan yang paling menarik baru-baru ini dan telah menjadi faktor utama dalam berbagai keberhasilan baru-baru ini dan aplikasi yang menantang terkait dengan aplikasi pembelajaran mesin seperti deteksi objek ImageNet tantangan, klasifikasi gambar, dan pengenalan wajah. Algoritma CNN digunakan untuk segmentasi dan klasifikasi gambar dalam transaksi akademik. Algoritma ini dimanfaatkan untuk pengenalan gambar di berbagai area misalnya pengaturan gambar otomatis, fotografi stok, pengenalan wajah, dan banyak pekerjaan terkait lainnya [3].

Algoritma CNN dimanfaatkan untuk mengklasifikasikan gambar, mengelompokkannya berdasarkan kesamaan (pencarian foto), dan melakukan pengenalan objek dalam adegan. Pada algoritma ini, terdapat lapisan konvolusional sebagai blok bangunan inti dari CNN. Parameter lapisan terdiri dari satu set filter yang dapat dipelajari (atau kernel) yang memiliki reseptif kecil tetapi meluas melalui kedalaman penuh volume input. Selama *forward pass*, setiap filter dikonvolitasi melintasi lebar dan tinggi volume input, menghitung produk titik, dan menghasilkan peta aktivasi 2 dimensi dari filter tersebut. Akibatnya, jaringan akan dapat mempelajari tentang filter. Filter diaktifkan ketika akan dilakukan pencarian terhadap beberapa jenis fitur tertentu pada beberapa posisi spasial dalam *input*. Kemudian peta aktivasi dimasukkan ke dalam lapisan *downsampling*, dan seperti konvolusi, metode ini diterapkan satu tambalan pada satu waktu. Algoritma CNN juga memiliki lapisan yang terhubung sepenuhnya yang mengklasifikasikan output dengan satu label per node [3].

2.2 Penerapan CNN dalam Klasifikasi Gambar

Penerapan algoritma CNN dalam klasifikasi gambar melibatkan beberapa langkah utama. Pertama, dilakukan persiapan data, yaitu pengaturan data pelatihan dan pengujian, penggabungan data, penggabungan label, dan penyesuaian ukuran data yang sesuai. Algoritma CNN digunakan untuk mengekstraksi, menganalisis, dan memahami informasi dari gambar secara otomatis. Dalam sistem klasifikasi gambar otomatis, algoritma ini sering digunakan untuk memanfaatkan fitur dari lapisan atas untuk klasifikasi, meskipun dalam beberapa kasus, fitur dari lapisan bawah dapat

memiliki kekuatan diskriminatif yang lebih besar. CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi yang merupakan blok bangunan inti dari CNN [3].

Lapisan ini memiliki parameter berupa filter yang dapat dipelajari, yang memiliki bidang reseptif kecil tetapi meluas melalui kedalaman penuh dari volume input. Selama proses *forward pass*, setiap filter dikonvolusikan melintasi lebar dan tinggi volume input, menghasilkan peta aktivasi dua dimensi. Aktivasi ini kemudian digunakan untuk mengklasifikasikan gambar ke dalam kelas yang telah ditentukan, seperti pesawat, mobil, burung, kucing, dan lainnya [3].

BAB 3

METODE PENELITIAN

3.1 *Cross-Industry Standard Process for Data Mining (CRISP-DM)*

Berikut adalah struktur pemodelan untuk pemrosesan klasifikasi gambar motif ulos.

3.1.1 *Business Understanding*

Business understanding dilakukan untuk memastikan bahwa solusi yang dikembangkan benar-benar relevan dan bermanfaat bagi pihak-pihak yang terlibat. Dengan memahami kebutuhan dan tantangan yang ada misalnya, kesulitan dalam mengidentifikasi berbagai jenis ulos secara manual kita bisa mengarahkan pengembangan algoritma CNN untuk memenuhi tujuan praktis, seperti mempermudah proses identifikasi ulos. Dengan pendekatan ini, teknologi yang digunakan bisa lebih tepat sasaran dan memberikan manfaat yang maksimal.

Ulos adalah kain tradisional suku Batak yang memiliki berbagai motif sesuai dengan makna budayanya. Motif yang beragam dengan variasi dan pola yang rumit sering kali menyebabkan penggolongan ulos yang cukup rumit. Tugas analitik utama adalah klasifikasi motif ulos. Dengan menggunakan algoritma CNN, proyek ini bertujuan untuk memprediksi kelas atau kategori ulos berdasarkan gambar motifnya. Dataset gambar ulos diambil dari Kaggle, dan diharapkan dapat membantu proses identifikasi, pelabelan, dan dokumentasi motif ulos secara otomatis dan efisien. Tujuan dari proyek ini adalah mengembangkan algoritma CNN untuk melakukan klasifikasi terhadap motif ulos berdasarkan jenis motifnya. Target utamanya adalah mencapai akurasi sebesar $\geq 90\%$ pada data uji.

3.1.2 *Data Understanding*

Berikut adalah tujuan dari setiap langkah dalam data understanding untuk proyek klasifikasi ulos menggunakan model CNN:

a. Mengumpulkan Data

Tujuan utama dari mengumpulkan data adalah untuk mendapatkan gambar-gambar ulos yang representatif dari berbagai jenis dan motif. Data yang lengkap dan bervariasi akan membantu algoritma CNN dalam belajar mengenali berbagai pola dan fitur visual pada ulos. Pengumpulan data yang baik adalah dasar yang penting untuk membangun model

yang akurat. Berikut adalah untuk menampilkan dataset ulos yang diambil dari Kaggle, dapat dilihat pada Tabel 3-1.

Tabel 3-1 Kode Program untuk Mengumpulkan Data

```
import kagglehub

# Download dataset kain ulos
path = kagglehub.dataset_download("fthnaja/kain-ulos")

print("Path to dataset files:", path)
```

Selanjutnya, dilakukan proses pengumpulan sejumlah gambar ulos yang cukup banyak agar model CNN dapat dilatih dengan baik. Jumlah data yang memadai penting untuk memastikan model dapat generalisasi dengan baik. Biasanya, semakin banyak data, semakin baik kemampuan model untuk mengenali pola-pola dalam gambar ulos. Code untuk menampilkan jumlah data dapat dilihat pada Tabel 3-2.

Tabel 3-2 Kode Program Untuk Menampilkan Jumlah Data

```
import os
import pandas as pd
from tabulate import tabulate

# Path dataset yang telah diunduh
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk menghitung jumlah gambar di seluruh dataset dan menampilkan sebagai tabel
def count_images_per_category(dataset_path):
    category_counts = {}

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('.png', '.jpg', '.jpeg', '.tiff', '.bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                if category not in category_counts:
                    category_counts[category] = 0
```

```

category_counts[category] += 1

# Mengubah hasil ke dalam bentuk DataFrame untuk tampilan tabel
category_df = pd.DataFrame(list(category_counts.items()), columns=['Category', 'Image Count'])
return category_df

# Hitung jumlah gambar per kategori dan tampilkan
category_df = count_images_per_category(dataset_path)

# Menampilkan tabel jumlah gambar per kategori menggunakan tabulate
print(tabulate(category_df, headers='keys', tablefmt='pretty', showindex=False))

# Menampilkan jumlah total gambar
total_images = category_df['Image Count'].sum()
print(f"\nJumlah total gambar dalam dataset: {total_images}")

```

Jumlah data yang dihasilkan dapat dilihat pada Tabel 3-3.

Tabel 3-3 Jumlah Data

```

+-----+-----+
| Category | Image Count |
+-----+-----+
| Tumtuman | 206 |
| Pinunsaan | 201 |
| Ragi Hotang | 209 |
| Sibolang | 206 |
| Sadum | 204 |
| Ragi Hidup | 205 |
+-----+-----+

Jumlah total gambar dalam dataset: 1231

```

Selanjutnya, data yang dikumpulkan harus memiliki deskripsi yang jelas, seperti jenis ulos, motif, warna, dan kategori lainnya. Deskripsi ini penting agar setiap gambar dapat diberi

label dengan benar dan memudahkan proses pelatihan model. Berikut kode program untuk melihat deskripsi data yang digunakan, dan dapat dilihat pada Tabel 3-4.

Tabel 3-4 Kode Program untuk Menampilkan Deskripsi Data

```
import os
import pandas as pd
from tabulate import tabulate

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan deskripsi data
def describe_dataset(dataset_path):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('png', 'jpg', 'jpeg', 'tiff', 'bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_details.append({'Category': category, 'File Name': file, 'File Format': file_format})

    # Mengubah hasil ke dalam bentuk DataFrame
    file_df = pd.DataFrame(file_details)
    return file_df

# Mendapatkan deskripsi dataset
file_df = describe_dataset(dataset_path)

# Menampilkan semua deskripsi dataset menggunakan tabulate
print(tabulate(file_df, headers='keys', tablefmt='pretty', showindex=False)) # Menampilkan semua baris

# Menampilkan jumlah total file yang dianalisis
print(f"\nJumlah total file dalam dataset: {len(file_df)}")
```


Deskripsi data yang dihasilkan dapat dilihat pada Tabel 3-5.

Tabel 3-5 Deskripsi Data

Category	File Name	File Format
Tumtuman	ulos (12) - Copy.jpeg	jpeg
Tumtuman	IMG_20240611_202207_2.jpg	jpg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (28).jpeg	jpeg
Tumtuman	ulos (64) - Copy.jpeg	jpeg
Tumtuman	ulos (1).jpeg	jpeg
Tumtuman	ulos (68) - Copy.jpeg	jpeg
Tumtuman	ulos (4).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (11).jpeg	jpeg
Tumtuman	ulos (76).jpeg	jpeg
Tumtuman	IMG_20240611_202224.jpg	jpg
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (6).jpeg	jpeg
Tumtuman	IMG_20240611_202207_1.jpg	jpg
Tumtuman	IMG_20240611_202234.jpg	jpg
Tumtuman	IMG_20240611_202236.jpg	jpg
Tumtuman	ulos (11) - Copy.jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (14).jpeg	jpeg
Tumtuman	ulos (86).jpeg	jpeg
Tumtuman	ulos (88).jpeg	jpeg
Tumtuman	ulos (10) - Copy.jpeg	jpeg
Tumtuman	ulos (64).jpeg	jpeg
Tumtuman	ulos (59).jpeg	jpeg
Tumtuman	IMG_20240611_202200_3.jpg	jpg
Tumtuman	ulos (13) - Copy.jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (27).jpeg	jpeg
Tumtuman	IMG_20240611_202233.jpg	jpg
Tumtuman	ulos (14) - Copy.jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (24).jpeg	jpeg
Tumtuman	ulos (48) - Copy.jpeg	jpeg
Tumtuman	IMG_20240611_202201.jpg	jpg
Tumtuman	ulos (80) - Copy.jpeg	jpeg
Tumtuman	ulos (93).jpeg	jpeg

Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (19).jpeg	jpeg	
Tumtuman	IMG_20240611_202237.jpg	jpg	
Tumtuman	ulos (55) - Copy.jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (30).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (21).jpeg	jpeg	
Tumtuman	IMG_20240611_202240.jpg	jpg	
Tumtuman	ulos (62).jpeg	jpeg	
Tumtuman	IMG_20240611_202236_2.jpg	jpg	
Tumtuman	IMG_20240611_202225_1.jpg	jpg	
Tumtuman	ulos (95).jpeg	jpeg	
Tumtuman	IMG_20240611_202229.jpg	jpg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (10).jpeg	jpeg	
Tumtuman	IMG_20240611_202225.jpg	jpg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (26).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (20).jpeg	jpeg	
Tumtuman	ulos (7) - Copy.jpeg	jpeg	
Tumtuman	IMG_20240611_202224_1.jpg	jpg	
Tumtuman	IMG_20240611_202207.jpg	jpg	
Tumtuman	ulos (6).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (7).jpeg	jpeg	
Tumtuman	ulos (61).jpeg	jpeg	
Tumtuman	ulos (54) - Copy.jpeg	jpeg	
Tumtuman	IMG_20240611_202238_1.jpg	jpg	
Tumtuman	ulos (44) - Copy.jpeg	jpeg	
Tumtuman	ulos (51) - Copy.jpeg	jpeg	
Tumtuman	IMG_20240611_202159_1.jpg	jpg	
Tumtuman	ulos (75).jpeg	jpeg	
Tumtuman	IMG_20240611_202200_1.jpg	jpg	
Tumtuman	IMG_20240611_202238.jpg	jpg	
Tumtuman	ulos (78).jpeg	jpeg	
Tumtuman	ulos (74).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (15).jpeg	jpeg	
Tumtuman	ulos (81) - Copy.jpeg	jpeg	
Tumtuman	IMG_20240611_202237_1.jpg	jpg	
Tumtuman	ulos (63).jpeg	jpeg	
Tumtuman	ulos (92).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (4).jpeg	jpeg	

Tumtuman		ulos (52) - Copy.jpeg		jpeg	
Tumtuman		ulos (47) - Copy.jpeg		jpeg	
Tumtuman		ulos (6) - Copy.jpeg		jpeg	
Tumtuman		IMG_20240611_202239_1.jpg		jpg	
Tumtuman		ulos (5).jpeg		jpeg	
Tumtuman		ulos (94).jpeg		jpeg	
Tumtuman		IMG_20240611_202240_1.jpg		jpg	
Tumtuman		ulos (58).jpeg		jpeg	
Tumtuman		ulos (14).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (18).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (9).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 (3).jpeg		jpeg	
Tumtuman		ulos (3).jpeg		jpeg	
Tumtuman		ulos (90).jpeg		jpeg	
Tumtuman		ulos (8) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (12).jpeg		jpeg	
Tumtuman		ulos (46) - Copy.jpeg		jpeg	
Tumtuman		ulos (89).jpeg		jpeg	
Tumtuman		IMG_20240611_202231.jpg		jpg	
Tumtuman		ulos (43) - Copy.jpeg		jpeg	
Tumtuman		IMG_20240611_202236_1.jpg		jpg	
Tumtuman		IMG_20240611_202206.jpg		jpg	
Tumtuman		ulos (50) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 (5).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (22).jpeg		jpeg	
Tumtuman		ulos (80).jpeg		jpeg	
Tumtuman		ulos (65) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (25).jpeg		jpeg	
Tumtuman		ulos (66) - Copy.jpeg		jpeg	
Tumtuman		IMG_20240611_202159.jpg		jpg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 (8).jpeg		jpeg	
Tumtuman		ulos (79).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (23).jpeg		jpeg	
Tumtuman		IMG_20240611_202208.jpg		jpg	
Tumtuman		IMG_20240611_202226.jpg		jpg	
Tumtuman		ulos (16) - Copy.jpeg		jpeg	

Tumtuman		ulos (49) - Copy.jpeg		jpeg	
Tumtuman		IMG_20240611_202239.jpg		jpg	
Tumtuman		IMG_20240611_202200_2.jpg		jpg	
Tumtuman		ulos (17) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (17).jpeg		jpeg	
Tumtuman		ulos (13).jpeg		jpeg	
Tumtuman		ulos (57).jpeg		jpeg	
Tumtuman		IMG_20240611_202221.jpg		jpg	
Tumtuman		ulos (96).jpeg		jpeg	
Tumtuman		ulos (15) - Copy.jpeg		jpeg	
Tumtuman		IMG_20240611_202223.jpg		jpg	
Tumtuman		ulos (53) - Copy.jpeg		jpeg	
Tumtuman		ulos (83).jpeg		jpeg	
Tumtuman		ulos (67) - Copy.jpeg		jpeg	
Tumtuman		ulos (87).jpeg		jpeg	
Tumtuman		ulos (60).jpeg		jpeg	
Tumtuman		IMG_20240611_202235.jpg		jpg	
Tumtuman		IMG_20240611_202200.jpg		jpg	
Tumtuman		ulos (85).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (13).jpeg		jpeg	
Tumtuman		IMG_20240611_202235_1.jpg		jpg	
Tumtuman		ulos (91).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (16).jpeg		jpeg	
Tumtuman		ulos (2).jpeg		jpeg	
Tumtuman		ulos (45) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (29).jpeg		jpeg	
Tumtuman		ulos (84).jpeg		jpeg	
Tumtuman		ulos (77).jpeg		jpeg	
Tumtuman		ulos (9) - Copy.jpeg		jpeg	
Tumtuman		ulos (69) - Copy.jpeg		jpeg	
Tumtuman		ulos (97).jpeg		jpeg	
Pinuncaan		IMG_5373.PNG		PNG	
Pinuncaan		IMG_5331.PNG		PNG	
Pinuncaan		pinuncanaan (8).PNG		PNG	
Pinuncaan		IMG_20240611_201117.jpg		jpg	
Pinuncaan		IMG_5268.PNG		PNG	
Pinuncaan		IMG_20240611_201125.jpg		jpg	

Pinuncaan	IMG_20240611_2011166.jpg	jpg
Pinuncaan	IMG_E4593.JPG	JPG
Pinuncaan	IMG_20240611_201124.jpg	jpg
Pinuncaan	IMG_20240611_201120_22.jpg	jpg
Pinuncaan	IMG_5330.PNG	PNG
Pinuncaan	IMG_20240611_201115.jpg	jpg
Pinuncaan	IMG_20240611_2011100.jpg	jpg
Pinuncaan	IMG_5364.PNG	PNG
Pinuncaan	IMG_5340.PNG	PNG
Pinuncaan	IMG_20240611_2011222.jpg	jpg
Pinuncaan	IMG_5367.PNG	PNG
Pinuncaan	IMG_5389.PNG	PNG
Pinuncaan	IMG_20240611_201120_11.jpg	jpg
Pinuncaan	IMG_20240611_201121_2.jpg	jpg
Pinuncaan	IMG_5325.PNG	PNG
Pinuncaan	pinuncanaan (1).PNG	PNG
Pinuncaan	IMG_5337.PNG	PNG
Pinuncaan	IMG_5333.PNG	PNG
Pinuncaan	pinuncanaan (3).PNG	PNG
Pinuncaan	IMG_5357.PNG	PNG
Pinuncaan	IMG_20240611_201114_1.jpg	jpg
Pinuncaan	IMG_E5275.JPG	JPG
Pinuncaan	IMG_5313.PNG	PNG
Pinuncaan	pinuncanaan (9).PNG	PNG
Pinuncaan	IMG_E5236.JPG	JPG
Pinuncaan	IMG_5342.PNG	PNG
Pinuncaan	IMG_20240611_201119.jpg	jpg
Pinuncaan	IMG_5386.PNG	PNG
Pinuncaan	IMG_5332.PNG	PNG
Pinuncaan	pinuncanaan (2).JPG	JPG
Pinuncaan	IMG_5327.PNG	PNG
Pinuncaan	IMG_5363.PNG	PNG
Pinuncaan	IMG_5323.PNG	PNG
Pinuncaan	IMG_5328.PNG	PNG
Pinuncaan	IMG_5380.PNG	PNG
Pinuncaan	IMG_20240611_201122_1.jpg	jpg
Pinuncaan	IMG_5370.PNG	PNG

Pinuncaan	IMG_5273.PNG	PNG
Pinuncaan	IMG_5321.PNG	PNG
Pinuncaan	IMG_20240611_201116.jpg	jpg
Pinuncaan	IMG_20240611_201124_1.jpg	jpg
Pinuncaan	IMG_20240611_2011200.jpg	jpg
Pinuncaan	IMG_20240611_201120.jpg	jpg
Pinuncaan	pinuncanaan (7).PNG	PNG
Pinuncaan	IMG_5334.PNG	PNG
Pinuncaan	IMG_5272.PNG	PNG
Pinuncaan	IMG_20240611_201121_1.jpg	jpg
Pinuncaan	IMG_5274.PNG	PNG
Pinuncaan	IMG_5377.PNG	PNG
Pinuncaan	IMG_5372.PNG	PNG
Pinuncaan	IMG_E4592.JPG	JPG
Pinuncaan	IMG_5315.PNG	PNG
Pinuncaan	IMG_5378.PNG	PNG
Pinuncaan	IMG_5359.PNG	PNG
Pinuncaan	IMG_20240611_201114.jpg	jpg
Pinuncaan	IMG_20240611_2011255.jpg	jpg
Pinuncaan	IMG_5318.PNG	PNG
Pinuncaan	IMG_20240611_201120_2.jpg	jpg
Pinuncaan	IMG_5326.PNG	PNG
Pinuncaan	IMG_20240611_201121_22.jpg	jpg
Pinuncaan	IMG_5341.PNG	PNG
Pinuncaan	IMG_5320.PNG	PNG
Pinuncaan	IMG_5354.PNG	PNG
Pinuncaan	pinuncanaan (10).PNG	PNG
Pinuncaan	IMG_20240611_2011211.jpg	jpg
Pinuncaan	IMG_5387.PNG	PNG
Pinuncaan	pinuncanaan (6).PNG	PNG
Pinuncaan	IMG_5360.PNG	PNG
Pinuncaan	IMG_5339.PNG	PNG
Pinuncaan	IMG_5358.PNG	PNG
Pinuncaan	IMG_5385.PNG	PNG
Pinuncaan	IMG_20240611_201124_11.jpg	jpg
Pinuncaan	IMG_E5323.JPG	JPG
Pinuncaan	IMG_5322.PNG	PNG

Pinuncaan	IMG_20240611_2011277.jpg	jpg
Pinuncaan	IMG_5369.PNG	PNG
Pinuncaan	IMG_5335.PNG	PNG
Pinuncaan	IMG_5271.PNG	PNG
Pinuncaan	IMG_5319.PNG	PNG
Pinuncaan	IMG_20240611_2011177.jpg	jpg
Pinuncaan	IMG_5381.PNG	PNG
Pinuncaan	IMG_20240611_2011244.jpg	jpg
Pinuncaan	IMG_20240611_2011099.jpg	jpg
Pinuncaan	IMG_20240611_201122.jpg	jpg
Pinuncaan	pinuncanaan (11).PNG	PNG
Pinuncaan	IMG_5366.PNG	PNG
Pinuncaan	IMG_5379.PNG	PNG
Pinuncaan	IMG_5371.PNG	PNG
Pinuncaan	IMG_5352.PNG	PNG
Pinuncaan	IMG_E5313.JPG	JPG
Pinuncaan	pinuncanaan (2).PNG	PNG
Pinuncaan	IMG_E4591.JPG	JPG
Pinuncaan	IMG_5353.PNG	PNG
Pinuncaan	IMG_20240611_2011155.jpg	jpg
Pinuncaan	IMG_5314.PNG	PNG
Pinuncaan	IMG_20240611_2011088.jpg	jpg
Pinuncaan	IMG_20240611_201123_11.jpg	jpg
Pinuncaan	pinuncanaan (4).PNG	PNG
Pinuncaan	IMG_20240611_201121_111.jpg	jpg
Pinuncaan	IMG_5384.PNG	PNG
Pinuncaan	IMG_5355.PNG	PNG
Pinuncaan	IMG_20240611_201123.jpg	jpg
Pinuncaan	IMG_5270.PNG	PNG
Pinuncaan	IMG_5374.PNG	PNG
Pinuncaan	IMG_5336.PNG	PNG
Pinuncaan	pinuncanaan (5).PNG	PNG
Pinuncaan	IMG_5275.PNG	PNG
Pinuncaan	IMG_5329.PNG	PNG
Pinuncaan	IMG_20240611_2011199.jpg	jpg
Pinuncaan	IMG_20240611_2011077.jpg	jpg
Pinuncaan	IMG_5316.PNG	PNG

Pinuncaan	IMG_5388.PNG	PNG
Pinuncaan	IMG_5267.PNG	PNG
Pinuncaan	IMG_20240611_201122_11.jpg	jpg
Pinuncaan	IMG_E4595.JPG	JPG
Pinuncaan	IMG_20240611_201120_1.jpg	jpg
Pinuncaan	IMG_20240611_2011066.jpg	jpg
Pinuncaan	IMG_5375.PNG	PNG
Pinuncaan	pinuncanaan (1).JPG	JPG
Pinuncaan	IMG_5269.PNG	PNG
Pinuncaan	IMG_5361.PNG	PNG
Pinuncaan	IMG_20240611_201127.jpg	jpg
Pinuncaan	IMG_5365.PNG	PNG
Pinuncaan	IMG_20240611_201123_1.jpg	jpg
Pinuncaan	IMG_5368.PNG	PNG
Pinuncaan	IMG_20240611_2011233.jpg	jpg
Pinuncaan	IMG_5338.PNG	PNG
Pinuncaan	IMG_20240611_201121.jpg	jpg
Ragi Hotang	IMG_20240611_211522_1.jpg	jpg
Ragi Hotang	IMG_20240705_231339_1.jpg	jpg
Ragi Hotang	IMG_20240705_231128_44.jpg	jpg
Ragi Hotang	IMG_20240705_231355.jpg	jpg
Ragi Hotang	IMG_20240705_231143_9.jpg	jpg
Ragi Hotang	IMG_20240705_231359.jpg	jpg
Ragi Hotang	IMG_20240705_231413_1.jpg	jpg
Ragi Hotang	IMG_20240705_231128_24.jpg	jpg
Ragi Hotang	IMG_20240611_211534.jpg	jpg
Ragi Hotang	IMG_20240705_231128_49.jpg	jpg
Ragi Hotang	IMG_20240705_231128_35.jpg	jpg
Ragi Hotang	IMG_20240705_231436.jpg	jpg
Ragi Hotang	IMG_20240705_231341.jpg	jpg
Ragi Hotang	IMG_20240705_231143_1.jpg	jpg
Ragi Hotang	IMG_20240611_211452.jpg	jpg
Ragi Hotang	IMG_20240705_231143_3.jpg	jpg
Ragi Hotang	IMG_20240705_231128_47.jpg	jpg
Ragi Hotang	IMG_20240611_211520.jpg	jpg
Ragi Hotang	IMG_20240611_211524.jpg	jpg
Ragi Hotang	IMG_20240705_231358.jpg	jpg

Ragi Hotang	IMG_20240611_211529.jpg	jpg
Ragi Hotang	IMG_20240611_211522.jpg	jpg
Ragi Hotang	IMG_20240705_231410_1.jpg	jpg
Ragi Hotang	IMG_20240705_231417.jpg	jpg
Ragi Hotang	IMG_20240705_231333.jpg	jpg
Ragi Hotang	IMG_20240705_231128_23.jpg	jpg
Ragi Hotang	IMG_20240611_211412.jpg	jpg
Ragi Hotang	IMG_20240611_211404.jpg	jpg
Ragi Hotang	IMG_20240705_231413.jpg	jpg
Ragi Hotang	IMG_20240705_231128_48.jpg	jpg
Ragi Hotang	IMG_20240611_211429.jpg	jpg
Ragi Hotang	IMG_20240611_211411.jpg	jpg
Ragi Hotang	IMG_20240705_231143_4.jpg	jpg
Ragi Hotang	IMG_20240611_211406.jpg	jpg
Ragi Hotang	IMG_20240705_231143_2.jpg	jpg
Ragi Hotang	IMG_20240705_231356.jpg	jpg
Ragi Hotang	IMG_20240705_231418_1.jpg	jpg
Ragi Hotang	IMG_20240611_211418.jpg	jpg
Ragi Hotang	IMG_20240611_211521.jpg	jpg
Ragi Hotang	IMG_20240705_231143_30.jpg	jpg
Ragi Hotang	IMG_20240705_231438.jpg	jpg
Ragi Hotang	IMG_20240611_211410.jpg	jpg
Ragi Hotang	IMG_20240705_231143_47.jpg	jpg
Ragi Hotang	IMG_20240611_211435.jpg	jpg
Ragi Hotang	IMG_20240705_231143_32.jpg	jpg
Ragi Hotang	IMG_20240611_211526.jpg	jpg
Ragi Hotang	IMG_20240705_231422.jpg	jpg
Ragi Hotang	IMG_20240705_231143_6.jpg	jpg
Ragi Hotang	IMG_20240705_231143_5.jpg	jpg
Ragi Hotang	IMG_20240705_231143_27.jpg	jpg
Ragi Hotang	IMG_20240705_231408.jpg	jpg
Ragi Hotang	IMG_20240705_231353_1.jpg	jpg
Ragi Hotang	IMG_20240705_231128_50.jpg	jpg
Ragi Hotang	IMG_20240705_231407.jpg	jpg
Ragi Hotang	IMG_20240611_211501.jpg	jpg
Ragi Hotang	IMG_20240705_231443_2.jpg	jpg
Ragi Hotang	IMG_20240611_211528.jpg	jpg

Ragi Hotang	IMG_20240705_231332.jpg	jpg
Ragi Hotang	IMG_20240705_231143_10.jpg	jpg
Ragi Hotang	IMG_20240705_231143_33.jpg	jpg
Ragi Hotang	IMG_20240705_231143_49.jpg	jpg
Ragi Hotang	IMG_20240611_211426.jpg	jpg
Ragi Hotang	IMG_20240611_211416_1.jpg	jpg
Ragi Hotang	IMG_20240611_211402.jpg	jpg
Ragi Hotang	IMG_20240705_231438_1.jpg	jpg
Ragi Hotang	IMG_20240705_231128_28.jpg	jpg
Ragi Hotang	IMG_20240705_231410.jpg	jpg
Ragi Hotang	IMG_20240705_231443_1.jpg	jpg
Ragi Hotang	IMG_20240705_231444.jpg	jpg
Ragi Hotang	IMG_20240705_231426.jpg	jpg
Ragi Hotang	IMG_20240705_231409.jpg	jpg
Ragi Hotang	IMG_20240611_211414.jpg	jpg
Ragi Hotang	IMG_20240705_231420.jpg	jpg
Ragi Hotang	IMG_20240705_231401.jpg	jpg
Ragi Hotang	IMG_20240611_211423.jpg	jpg
Ragi Hotang	IMG_20240611_211519.jpg	jpg
Ragi Hotang	IMG_20240705_231128_38.jpg	jpg
Ragi Hotang	IMG_20240611_211454.jpg	jpg
Ragi Hotang	IMG_20240705_231427.jpg	jpg
Ragi Hotang	IMG_20240611_211530.jpg	jpg
Ragi Hotang	IMG_20240705_231437_1.jpg	jpg
Ragi Hotang	IMG_20240705_231416.jpg	jpg
Ragi Hotang	IMG_20240611_211523.jpg	jpg
Ragi Hotang	IMG_20240611_211401.jpg	jpg
Ragi Hotang	IMG_20240611_211533.jpg	jpg
Ragi Hotang	IMG_20240705_231331.jpg	jpg
Ragi Hotang	IMG_20240705_231143_28.jpg	jpg
Ragi Hotang	IMG_20240705_231421.jpg	jpg
Ragi Hotang	IMG_20240611_211413.jpg	jpg
Ragi Hotang	IMG_20240705_231428_1.jpg	jpg
Ragi Hotang	IMG_20240611_211520_1.jpg	jpg
Ragi Hotang	IMG_20240611_211416.jpg	jpg
Ragi Hotang	IMG_20240705_231443.jpg	jpg
Ragi Hotang	IMG_20240705_231143_8.jpg	jpg

Ragi Hotang	IMG_20240611_211532.jpg	jpg
Ragi Hotang	IMG_20240705_231357.jpg	jpg
Ragi Hotang	IMG_20240705_231404.jpg	jpg
Ragi Hotang	IMG_20240705_231143_50.jpg	jpg
Ragi Hotang	IMG_20240705_231143_48.jpg	jpg
Ragi Hotang	IMG_20240705_231334.jpg	jpg
Ragi Hotang	IMG_20240611_211527.jpg	jpg
Ragi Hotang	IMG_20240611_211357.jpg	jpg
Ragi Hotang	IMG_20240611_211354.jpg	jpg
Ragi Hotang	IMG_20240705_231143_29.jpg	jpg
Ragi Hotang	IMG_20240611_211403.jpg	jpg
Ragi Hotang	IMG_20240611_211415.jpg	jpg
Ragi Hotang	IMG_20240705_231128_36.jpg	jpg
Ragi Hotang	IMG_20240611_211458.jpg	jpg
Ragi Hotang	IMG_20240705_231423.jpg	jpg
Ragi Hotang	IMG_20240705_231143_13.jpg	jpg
Ragi Hotang	IMG_20240705_231425.jpg	jpg
Ragi Hotang	IMG_20240705_231426_1.jpg	jpg
Ragi Hotang	IMG_20240705_231339.jpg	jpg
Ragi Hotang	IMG_20240611_211538.jpg	jpg
Ragi Hotang	IMG_20240611_211450.jpg	jpg
Ragi Hotang	IMG_20240705_231421_1.jpg	jpg
Ragi Hotang	IMG_20240705_231406.jpg	jpg
Ragi Hotang	IMG_20240705_231444_1.jpg	jpg
Ragi Hotang	IMG_20240705_231403.jpg	jpg
Ragi Hotang	IMG_20240611_211405.jpg	jpg
Ragi Hotang	IMG_20240611_211535.jpg	jpg
Ragi Hotang	IMG_20240705_231415.jpg	jpg
Ragi Hotang	IMG_20240705_231437.jpg	jpg
Ragi Hotang	IMG_20240705_231428.jpg	jpg
Ragi Hotang	IMG_20240705_231411.jpg	jpg
Ragi Hotang	IMG_20240611_211535_1.jpg	jpg
Ragi Hotang	IMG_20240705_231412.jpg	jpg
Ragi Hotang	IMG_20240611_211407.jpg	jpg
Ragi Hotang	IMG_20240705_231418.jpg	jpg
Ragi Hotang	IMG_20240705_231128_37.jpg	jpg
Ragi Hotang	IMG_20240705_231128_25.jpg	jpg

Ragi Hotang	IMG_20240611_211524_1.jpg	jpg
Ragi Hotang	IMG_20240705_231128_27.jpg	jpg
Ragi Hotang	IMG_20240705_231442.jpg	jpg
Ragi Hotang	IMG_20240705_231143_46.jpg	jpg
Ragi Hotang	IMG_20240611_211525.jpg	jpg
Ragi Hotang	IMG_20240611_211531.jpg	jpg
Ragi Hotang	IMG_20240611_211417.jpg	jpg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (25).jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (17) - Copy.jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (12).jpeg	jpeg
Sibolang	1000104041.jpg	jpg
Sibolang	1000104024.jpg	jpg
Sibolang	1000104052.jpg	jpg
Sibolang	1000104043.jpg	jpg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (26).jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (27) - Copy.jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (28).jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (15) - Copy.jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (28) - Copy.jpeg	jpeg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (25) - Copy.jpeg	jpeg
Sibolang	IMG-20240705-WA0103.jpg	jpg
Sibolang	1000104058.jpg	jpg
Sibolang	IMG-20240705-WA0041.jpg	jpg
Sibolang	IMG-20240705-WA0045.jpg	jpg
Sibolang	IMG-20240705-WA0102.jpg	jpg
Sibolang	WhatsApp Image 2024-07-05 at 23.50.14 (1) - Copy.jpeg	jpeg
Sibolang	1000104030.jpg	jpg
Sibolang	1000104044.jpg	jpg
Sibolang	1000104056.jpg	jpg
Sibolang	1000104055.jpg	jpg
Sibolang	IMG-20240705-WA0120.jpg	jpg
Sibolang	1000104059.jpg	jpg
Sibolang	1000104045.jpg	jpg
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (22) - Copy.jpeg	jpeg
Sibolang	1000104034.jpg	jpg
Sibolang	IMG-20240705-WA0043.jpg	jpg
Sibolang	IMG-20240705-WA0026.jpg	jpg

Sibolang		1000104061.jpg		jpg	
Sibolang		1000104023.jpg		jpg	
Sibolang		sibolang2.jpeg		jpeg	
Sibolang		IMG-20240705-WA0031.jpg		jpg	
Sibolang		1000104035.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.50.14 (3) - Copy.jpeg		jpeg	
Sibolang		IMG-20240705-WA0104.jpg		jpg	
Sibolang		1000104054.jpg		jpg	
Sibolang		IMG-20240705-WA0122.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.50.14.jpeg		jpeg	
Sibolang		1000104050.jpg		jpg	
Sibolang		1000104046.jpg		jpg	
Sibolang		sibolang7.jpeg		jpeg	
Sibolang		1000104064.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (15).jpeg		jpeg	
Sibolang		IMG-20240705-WA0117.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (21).jpeg		jpeg	
Sibolang		1000104029.jpg		jpg	
Sibolang		sibolang9.jpg		jpg	
Sibolang		sibolang8.jpeg		jpeg	
Sibolang		IMG-20240705-WA0121.jpg		jpg	
Sibolang		sibolang12.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.50.14 (2) - Copy.jpeg		jpeg	
Sibolang		IMG-20240705-WA0106.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (23) - Copy.jpeg		jpeg	
Sibolang		IMG-20240705-WA0047.jpg		jpg	
Sibolang		1000104048.jpg		jpg	
Sibolang		1000104051.jpg		jpg	
Sibolang		1000104053.jpg		jpg	
Sibolang		1000104037.jpg		jpg	
Sibolang		sibolang5.jpeg		jpeg	
Sibolang		1000104042.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (24) - Copy.jpeg		jpeg	
Sibolang		IMG-20240705-WA0027.jpg		jpg	
Sibolang		IMG-20240705-WA0113.jpg		jpg	
Sibolang		sibolang13.jpg		jpg	
Sibolang		1000104039.jpg		jpg	

Sibolang	sibolang4.jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23.jpeg	jpeg	
Sibolang	IMG-20240705-WA0115.jpg	jpg	
Sibolang	sibolang14.jpeg	jpeg	
Sibolang	1000104062.jpg	jpg	
Sibolang	IMG-20240705-WA0107.jpg	jpg	
Sibolang	1000104060.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (14).jpeg	jpeg	
Sibolang	IMG-20240705-WA0046.jpg	jpg	
Sibolang	1000104033.jpg	jpg	
Sibolang	1000104040.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.51.15 (1).jpeg	jpeg	
Sibolang	1000104027.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (13).jpeg	jpeg	
Sibolang	1000104063.jpg	jpg	
Sibolang	IMG-20240705-WA0040.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.50.14 (15).jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (10).jpeg	jpeg	
Sibolang	IMG-20240705-WA0123.jpg	jpg	
Sibolang	IMG-20240705-WA0029.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (26) - Copy.jpeg	jpeg	
Sibolang	sibolang15.jpeg	jpeg	
Sibolang	sibolang10.jpg	jpg	
Sibolang	IMG-20240705-WA0105.jpg	jpg	
Sibolang	IMG-20240705-WA0111.jpg	jpg	
Sibolang	1000104028.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.50.14 (16) - Copy.jpeg	jpeg	
Sibolang	sibolang3.jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (8).jpeg	jpeg	
Sibolang	IMG-20240705-WA0030.jpg	jpg	
Sibolang	IMG-20240705-WA0044.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (22).jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (27).jpeg	jpeg	
Sibolang	IMG-20240705-WA0101.jpg	jpg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (20).jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.50.14 (17) - Copy.jpeg	jpeg	
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (29) - Copy.jpeg	jpeg	

Sibolang		IMG-20240705-WA0108.jpg		jpg	
Sibolang		IMG-20240705-WA0028.jpg		jpg	
Sibolang		IMG-20240705-WA0116.jpg		jpg	
Sibolang		1000104026.jpg		jpg	
Sibolang		IMG-20240705-WA0112.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (11).jpeg		jpeg	
Sibolang		1000104057.jpg		jpg	
Sibolang		IMG-20240705-WA0119.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (16) - Copy.jpeg		jpeg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (18) - Copy.jpeg		jpeg	
Sibolang		sibolang16.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (23).jpeg		jpeg	
Sibolang		WhatsApp Image 2024-07-05 at 23.50.14 (1).jpeg		jpeg	
Sibolang		IMG-20240705-WA0118.jpg		jpg	
Sibolang		IMG-20240705-WA0110.jpg		jpg	
Sibolang		IMG-20240705-WA0039.jpg		jpg	
Sibolang		IMG-20240705-WA0109.jpg		jpg	
Sibolang		1000104038.jpg		jpg	
Sibolang		IMG-20240705-WA0114.jpg		jpg	
Sibolang		sibolang.jpeg		jpeg	
Sibolang		WhatsApp Image 2024-07-05 at 23.50.14 (15) - Copy.jpeg		jpeg	
Sibolang		1000104022.jpg		jpg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (24).jpeg		jpeg	
Sibolang		WhatsApp Image 2024-07-05 at 23.49.23 (9).jpeg		jpeg	
Sibolang		1000104032.jpg		jpg	
Sibolang		1000104047.jpg		jpg	
Sibolang		1000104036.jpg		jpg	
Sibolang		1000104025.jpg		jpg	
Sibolang		sibolang11.jpg		jpg	
Sibolang		1000104031.jpg		jpg	
Sibolang		sibolang6.jpeg		jpeg	
Sadum		IMG_20240705_231538_9.jpg		jpg	
Sadum		IMG_20240705_231538_5.jpg		jpg	
Sadum		IMG_20240705_231538_36.jpg		jpg	
Sadum		IMG_20240705_231538_11.jpg		jpg	
Sadum		IMG_20240705_231538_17.jpg		jpg	
Sadum		IMG_20240705_231550_2.jpg		jpg	

	Sadum		IMG_20240705_231530_2.jpg		jpg	
	Sadum		IMG_20240705_231550_1.jpg		jpg	
	Sadum		IMG_20240705_231538_14.jpg		jpg	
	Sadum		IMG_20240705_231550_17.jpg		jpg	
	Sadum		IMG_20240705_231559_37.jpg		jpg	
	Sadum		IMG_20240705_231538_35.jpg		jpg	
	Sadum		IMG_20240705_231550_14.jpg		jpg	
	Sadum		IMG_20240705_231550_13.jpg		jpg	
	Sadum		IMG_20240705_231538_12.jpg		jpg	
	Sadum		IMG_20240611_193359.jpg		jpg	
	Sadum		IMG_20240611_193405.jpg		jpg	
	Sadum		IMG_20240705_231559_8.jpg		jpg	
	Sadum		IMG_20240705_231538_34.jpg		jpg	
	Sadum		IMG_20240705_231538_28.jpg		jpg	
	Sadum		IMG_20240705_231530_10.jpg		jpg	
	Sadum		IMG_20240705_231824.jpg		jpg	
	Sadum		IMG_20240705_231559_40.jpg		jpg	
	Sadum		IMG_20240705_231538_4.jpg		jpg	
	Sadum		IMG_20240705_231538_39.jpg		jpg	
	Sadum		IMG_20240705_231559_3.jpg		jpg	
	Sadum		IMG_20240705_231559_10.jpg		jpg	
	Sadum		IMG_20240705_231550_6.jpg		jpg	
	Sadum		IMG_20240611_193403.jpg		jpg	
	Sadum		IMG_20240705_231559_43.jpg		jpg	
	Sadum		IMG_20240705_231530_31.jpg		jpg	
	Sadum		IMG_20240705_231559_6.jpg		jpg	
	Sadum		IMG_20240611_193358.jpg		jpg	
	Sadum		IMG_20240705_231550_3.jpg		jpg	
	Sadum		IMG_20240705_231530_15.jpg		jpg	
	Sadum		IMG_20240705_231822.jpg		jpg	
	Sadum		IMG_20240705_231550_9.jpg		jpg	
	Sadum		IMG_20240705_231538_32.jpg		jpg	
	Sadum		IMG_20240705_231538_29.jpg		jpg	
	Sadum		IMG_20240611_193247.jpg		jpg	
	Sadum		IMG_20240611_193411.jpg		jpg	
	Sadum		IMG_20240705_231821.jpg		jpg	
	Sadum		IMG_20240705_231527_5.jpg		jpg	

	Sadum		IMG_20240705_231538_33.jpg		jpg	
	Sadum		IMG_20240705_231538_26.jpg		jpg	
	Sadum		IMG_20240705_231550_15.jpg		jpg	
	Sadum		IMG_20240705_231559_9.jpg		jpg	
	Sadum		IMG_20240705_231530_17.jpg		jpg	
	Sadum		IMG_20240705_231550_12.jpg		jpg	
	Sadum		IMG_20240705_231538_23.jpg		jpg	
	Sadum		IMG_20240611_193239.jpg		jpg	
	Sadum		IMG_20240705_231550_8.jpg		jpg	
	Sadum		IMG_20240705_231530_8.jpg		jpg	
	Sadum		IMG_20240705_231559_33.jpg		jpg	
	Sadum		IMG_20240705_231530_14.jpg		jpg	
	Sadum		IMG_20240611_193242.jpg		jpg	
	Sadum		IMG_20240705_231559_38.jpg		jpg	
	Sadum		IMG_20240705_231538_24.jpg		jpg	
	Sadum		IMG_20240705_231559_5.jpg		jpg	
	Sadum		IMG_20240705_231820_1.jpg		jpg	
	Sadum		IMG_20240705_231538_8.jpg		jpg	
	Sadum		IMG_20240705_231825_1.jpg		jpg	
	Sadum		IMG_20240705_231559_39.jpg		jpg	
	Sadum		IMG_20240705_231550_4.jpg		jpg	
	Sadum		IMG_20240705_231530_12.jpg		jpg	
	Sadum		IMG_20240611_193306.jpg		jpg	
	Sadum		IMG_20240705_231538_18.jpg		jpg	
	Sadum		IMG_20240705_231530_18.jpg		jpg	
	Sadum		IMG_20240611_193342.jpg		jpg	
	Sadum		IMG_20240705_231538_22.jpg		jpg	
	Sadum		IMG_20240611_193353.jpg		jpg	
	Sadum		IMG_20240705_231550_5.jpg		jpg	
	Sadum		IMG_20240611_193351.jpg		jpg	
	Sadum		IMG_20240705_231559_31.jpg		jpg	
	Sadum		IMG_20240705_231559_44.jpg		jpg	
	Sadum		IMG_20240705_231530_11.jpg		jpg	
	Sadum		IMG_20240705_231826.jpg		jpg	
	Sadum		IMG_20240611_193245.jpg		jpg	
	Sadum		IMG_20240705_231822_1.jpg		jpg	
	Sadum		IMG_20240705_231538_31.jpg		jpg	

	Sadum		IMG_20240611_193355.jpg		jpg	
	Sadum		IMG_20240705_231538_6.jpg		jpg	
	Sadum		IMG_20240705_231538_15.jpg		jpg	
	Sadum		IMG_20240705_231559_1.jpg		jpg	
	Sadum		IMG_20240705_231530_34.jpg		jpg	
	Sadum		IMG_20240611_193409.jpg		jpg	
	Sadum		IMG_20240705_231538_10.jpg		jpg	
	Sadum		IMG_20240705_231530_35.jpg		jpg	
	Sadum		IMG_20240705_231538_25.jpg		jpg	
	Sadum		IMG_20240705_231559_34.jpg		jpg	
	Sadum		IMG_20240705_231527_6.jpg		jpg	
	Sadum		IMG_20240611_193257.jpg		jpg	
	Sadum		IMG_20240705_231559_7.jpg		jpg	
	Sadum		IMG_20240705_231550_11.jpg		jpg	
	Sadum		IMG_20240705_231559_35.jpg		jpg	
	Sadum		IMG_20240705_231559_32.jpg		jpg	
	Sadum		IMG_20240705_231550_16.jpg		jpg	
	Sadum		IMG_20240705_231559_41.jpg		jpg	
	Sadum		IMG_20240705_231559_45.jpg		jpg	
	Sadum		IMG_20240705_231538_37.jpg		jpg	
	Sadum		IMG_20240705_231530_7.jpg		jpg	
	Sadum		IMG_20240611_193349.jpg		jpg	
	Sadum		IMG_20240705_231559_4.jpg		jpg	
	Sadum		IMG_20240705_231530_3.jpg		jpg	
	Sadum		IMG_20240611_193347.jpg		jpg	
	Sadum		IMG_20240705_231530_27.jpg		jpg	
	Sadum		IMG_20240611_193316.jpg		jpg	
	Sadum		IMG_20240705_231825.jpg		jpg	
	Sadum		IMG_20240705_231538_19.jpg		jpg	
	Sadum		IMG_20240705_231538_30.jpg		jpg	
	Sadum		IMG_20240705_231538_38.jpg		jpg	
	Sadum		IMG_20240705_231538_20.jpg		jpg	
	Sadum		IMG_20240611_193241.jpg		jpg	
	Sadum		IMG_20240705_231559_2.jpg		jpg	
	Sadum		IMG_20240611_193305.jpg		jpg	
	Sadum		IMG_20240611_193303.jpg		jpg	
	Sadum		IMG_20240705_231559_42.jpg		jpg	

Sadum	IMG_20240705_231530_13.jpg	jpg
Sadum	IMG_20240705_231538_27.jpg	jpg
Sadum	IMG_20240705_231527_4.jpg	jpg
Sadum	IMG_20240705_231530_19.jpg	jpg
Sadum	IMG_20240705_231538_7.jpg	jpg
Sadum	IMG_20240705_231550_10.jpg	jpg
Sadum	IMG_20240705_231559_36.jpg	jpg
Sadum	IMG_20240611_193308.jpg	jpg
Sadum	IMG_20240705_231538_16.jpg	jpg
Sadum	IMG_20240705_231530_23.jpg	jpg
Sadum	IMG_20240611_193345.jpg	jpg
Sadum	IMG_20240705_231550_7.jpg	jpg
Sadum	IMG_20240611_193313.jpg	jpg
Sadum	IMG_20240611_193250.jpg	jpg
Sadum	IMG_20240705_231538_21.jpg	jpg
Sadum	IMG_20240705_231538_13.jpg	jpg
Sadum	IMG_20240705_231527_3.jpg	jpg
Sadum	IMG_20240705_231530_9.jpg	jpg
Sadum	IMG_20240611_193401.jpg	jpg
Ragi Hidup	IMG_5178.PNG	PNG
Ragi Hidup	IMG_5136.PNG	PNG
Ragi Hidup	IMG_20240611_201519.jpg	jpg
Ragi Hidup	IMG_5199.PNG	PNG
Ragi Hidup	IMG_5184.PNG	PNG
Ragi Hidup	IMG_5161.PNG	PNG
Ragi Hidup	IMG_20240611_201524.jpg	jpg
Ragi Hidup	IMG_5196.PNG	PNG
Ragi Hidup	IMG_20240611_201513.jpg	jpg
Ragi Hidup	IMG_5173.PNG	PNG
Ragi Hidup	IMG_5152.PNG	PNG
Ragi Hidup	IMG_5226.PNG	PNG
Ragi Hidup	IMG_5211.PNG	PNG
Ragi Hidup	IMG_5163.PNG	PNG
Ragi Hidup	IMG_5146.PNG	PNG
Ragi Hidup	IMG_20240611_201516.jpg	jpg
Ragi Hidup	IMG_20240611_201520.jpg	jpg
Ragi Hidup	IMG_5164.PNG	PNG

Ragi Hidup	IMG_5231.PNG	PNG
Ragi Hidup	IMG_5182.PNG	PNG
Ragi Hidup	IMG_5207.PNG	PNG
Ragi Hidup	IMG_5141.PNG	PNG
Ragi Hidup	IMG_5224.PNG	PNG
Ragi Hidup	IMG_20240611_201511_1.jpg	jpg
Ragi Hidup	IMG_5203.PNG	PNG
Ragi Hidup	IMG_5129.PNG	PNG
Ragi Hidup	IMG_5162.PNG	PNG
Ragi Hidup	IMG_20240611_201514_1.jpg	jpg
Ragi Hidup	IMG_5133.PNG	PNG
Ragi Hidup	IMG_5142.PNG	PNG
Ragi Hidup	IMG_5157.PNG	PNG
Ragi Hidup	IMG_5130.PNG	PNG
Ragi Hidup	IMG_5138.PNG	PNG
Ragi Hidup	IMG_5160.PNG	PNG
Ragi Hidup	IMG_5154.PNG	PNG
Ragi Hidup	IMG_5209.PNG	PNG
Ragi Hidup	IMG_20240611_201533.jpg	jpg
Ragi Hidup	IMG_5170.PNG	PNG
Ragi Hidup	IMG_5234.PNG	PNG
Ragi Hidup	IMG_5132.PNG	PNG
Ragi Hidup	IMG_20240611_201512.jpg	jpg
Ragi Hidup	IMG_5128.PNG	PNG
Ragi Hidup	IMG_5185.PNG	PNG
Ragi Hidup	IMG_5126.PNG	PNG
Ragi Hidup	IMG_5193.PNG	PNG
Ragi Hidup	IMG_5168.PNG	PNG
Ragi Hidup	IMG_5204.PNG	PNG
Ragi Hidup	IMG_5153.PNG	PNG
Ragi Hidup	IMG_20240611_201521_1.jpg	jpg
Ragi Hidup	IMG_5202.PNG	PNG
Ragi Hidup	IMG_5137.PNG	PNG
Ragi Hidup	IMG_5214.PNG	PNG
Ragi Hidup	IMG_5172.PNG	PNG
Ragi Hidup	IMG_5235.PNG	PNG
Ragi Hidup	IMG_5131.PNG	PNG

Ragi Hidup	IMG_5183.PNG	PNG
Ragi Hidup	IMG_5143.PNG	PNG
Ragi Hidup	IMG_5217.PNG	PNG
Ragi Hidup	IMG_5220.PNG	PNG
Ragi Hidup	IMG_5159.PNG	PNG
Ragi Hidup	IMG_5134.PNG	PNG
Ragi Hidup	IMG_20240611_201521_3.jpg	jpg
Ragi Hidup	IMG_5151.PNG	PNG
Ragi Hidup	IMG_5201.PNG	PNG
Ragi Hidup	IMG_20240611_201522.jpg	jpg
Ragi Hidup	IMG_5179.PNG	PNG
Ragi Hidup	IMG_5191.PNG	PNG
Ragi Hidup	IMG_5225.PNG	PNG
Ragi Hidup	IMG_5192.PNG	PNG
Ragi Hidup	IMG_20240611_201521_2.jpg	jpg
Ragi Hidup	IMG_5139.PNG	PNG
Ragi Hidup	IMG_5148.PNG	PNG
Ragi Hidup	IMG_5223.PNG	PNG
Ragi Hidup	IMG_20240611_201515.jpg	jpg
Ragi Hidup	IMG_5150.PNG	PNG
Ragi Hidup	IMG_5176.PNG	PNG
Ragi Hidup	IMG_5045.PNG	PNG
Ragi Hidup	IMG_5144.PNG	PNG
Ragi Hidup	IMG_5171.PNG	PNG
Ragi Hidup	IMG_20240611_201527.jpg	jpg
Ragi Hidup	IMG_5215.PNG	PNG
Ragi Hidup	IMG_5186.PNG	PNG
Ragi Hidup	IMG_20240611_201521.jpg	jpg
Ragi Hidup	IMG_20240611_201520_1.jpg	jpg
Ragi Hidup	IMG_5124.PNG	PNG
Ragi Hidup	IMG_5158.PNG	PNG
Ragi Hidup	IMG_20240611_201532.jpg	jpg
Ragi Hidup	IMG_5127.PNG	PNG
Ragi Hidup	IMG_5197.PNG	PNG
Ragi Hidup	IMG_5166.PNG	PNG
Ragi Hidup	IMG_5212.PNG	PNG
Ragi Hidup	IMG_5195.PNG	PNG

Ragi Hidup	IMG_5175.PNG	PNG
Ragi Hidup	IMG_5135.PNG	PNG
Ragi Hidup	IMG_5187.PNG	PNG
Ragi Hidup	IMG_20240611_201525.jpg	jpg
Ragi Hidup	IMG_20240611_201513_1.jpg	jpg
Ragi Hidup	IMG_5167.PNG	PNG
Ragi Hidup	IMG_5228.PNG	PNG
Ragi Hidup	IMG_5198.PNG	PNG
Ragi Hidup	IMG_5125.PNG	PNG
Ragi Hidup	IMG_5232.PNG	PNG
Ragi Hidup	IMG_5213.PNG	PNG
Ragi Hidup	IMG_20240611_201514.jpg	jpg
Ragi Hidup	IMG_5219.PNG	PNG
Ragi Hidup	IMG_5155.PNG	PNG
Ragi Hidup	IMG_5216.PNG	PNG
Ragi Hidup	IMG_5177.PNG	PNG
Ragi Hidup	IMG_5180.PNG	PNG
Ragi Hidup	IMG_5222.PNG	PNG
Ragi Hidup	IMG_5140.PNG	PNG
Ragi Hidup	IMG_5044.PNG	PNG
Ragi Hidup	IMG_5200.PNG	PNG
Ragi Hidup	IMG_5227.PNG	PNG
Ragi Hidup	IMG_5190.PNG	PNG
Ragi Hidup	IMG_5189.PNG	PNG
Ragi Hidup	IMG_5206.PNG	PNG
Ragi Hidup	IMG_5188.PNG	PNG
Ragi Hidup	IMG_5165.PNG	PNG
Ragi Hidup	IMG_5147.PNG	PNG
Ragi Hidup	IMG_5149.PNG	PNG
Ragi Hidup	IMG_5145.PNG	PNG
Ragi Hidup	IMG_5181.PNG	PNG
Ragi Hidup	IMG_5169.PNG	PNG
Ragi Hidup	IMG_20240611_201511.jpg	jpg
Ragi Hidup	IMG_5174.PNG	PNG
Ragi Hidup	IMG_5210.PNG	PNG
Ragi Hidup	IMG_5233.PNG	PNG
Ragi Hidup	IMG_5218.PNG	PNG

Ragi Hidup	IMG_5208.PNG	PNG
Ragi Hidup	IMG_5194.PNG	PNG
Ragi Hidup	IMG_5205.PNG	PNG
Ragi Hidup	IMG_5221.PNG	PNG
Ragi Hidup	IMG_5156.PNG	PNG
Ragi Hidup	IMG_20240611_201529.jpg	jpg
Ragi Hidup	IMG_20240611_201501.jpg	jpg
Tumtuman	IMG_20240611_202207_2.jpg	jpg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (5).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (22).jpeg	jpeg
Tumtuman	ulos (31).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.58.54 (11).jpeg	jpeg
Tumtuman	ulos (22) - Copy.jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 - Copy (3).jpeg	jpeg
Tumtuman	ulos (101).jpeg	jpeg
Tumtuman	IMG_20240611_202207_1.jpg	jpg
Tumtuman	ulos (35).jpeg	jpeg
Tumtuman	ulos (18) - Copy.jpeg	jpeg
Tumtuman	ulos (24).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (3).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.58.54 (15).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (4).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.58.54 (13).jpeg	jpeg
Tumtuman	IMG_20240611_202200_3.jpg	jpg
Tumtuman	ulos (100).jpeg	jpeg
Tumtuman	IMG_20240611_202201.jpg	jpg
Tumtuman	ulos (37) - Copy.jpeg	jpeg
Tumtuman	ulos (40) - Copy.jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.58.54 (18).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (21).jpeg	jpeg
Tumtuman	ulos (99).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.58.54 (19).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (6).jpeg	jpeg
Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 - Copy (2).jpeg	jpeg
Tumtuman	ulos (19) - Copy.jpeg	jpeg
Tumtuman	ulos (41) - Copy.jpeg	jpeg
Tumtuman	ulos (98).jpeg	jpeg

Tumtuman		ulos (28).jpeg		jpeg	
Tumtuman		IMG_20240611_202207.jpg		jpg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (17).jpeg		jpeg	
Tumtuman		ulos (27).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (8).jpeg		jpeg	
Tumtuman		ulos (39) - Copy.jpeg		jpeg	
Tumtuman		ulos (102).jpeg		jpeg	
Tumtuman		ulos (36).jpeg		jpeg	
Tumtuman		ulos (32).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (7).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (20).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 (20).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (14).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-06 at 00.02.57 (1).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (21).jpeg		jpeg	
Tumtuman		ulos (25) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 (23).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27.jpeg		jpeg	
Tumtuman		IMG_20240611_202206.jpg		jpg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.54.27 - Copy (4).jpeg		jpeg	
Tumtuman		ulos (38) - Copy.jpeg		jpeg	
Tumtuman		ulos (29).jpeg		jpeg	
Tumtuman		ulos (26).jpeg		jpeg	
Tumtuman		IMG_20240611_202208.jpg		jpg	
Tumtuman		ulos (34).jpeg		jpeg	
Tumtuman		ulos (20) - Copy.jpeg		jpeg	
Tumtuman		ulos (25).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (12).jpeg		jpeg	
Tumtuman		IMG_20240611_202200_2.jpg		jpg	
Tumtuman		ulos (30).jpeg		jpeg	
Tumtuman		ulos (21) - Copy.jpeg		jpeg	
Tumtuman		ulos (24) - Copy.jpeg		jpeg	
Tumtuman		ulos (36) - Copy.jpeg		jpeg	
Tumtuman		ulos (42) - Copy.jpeg		jpeg	
Tumtuman		ulos (23) - Copy.jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54 (16).jpeg		jpeg	
Tumtuman		WhatsApp Image 2024-07-05 at 23.58.54.jpeg		jpeg	

Tumtuman	WhatsApp Image 2024-07-05 at 23.54.27 (24).jpeg	jpeg	
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (2).jpeg	jpeg	
Tumtuman	ulos (33).jpeg	jpeg	
Pinuncaan	IMG_20240611_201106.jpg	jpg	
Pinuncaan	IMG_5382.PNG	PNG	
Pinuncaan	IMG_20240611_201117.jpg	jpg	
Pinuncaan	IMG_4595.PNG	PNG	
Pinuncaan	IMG_5266.PNG	PNG	
Pinuncaan	IMG_20240611_201110.jpg	jpg	
Pinuncaan	IMG_4592.PNG	PNG	
Pinuncaan	IMG_20240611_201115.jpg	jpg	
Pinuncaan	IMG_5238.PNG	PNG	
Pinuncaan	IMG_5244.PNG	PNG	
Pinuncaan	IMG_5246.PNG	PNG	
Pinuncaan	IMG_5257.PNG	PNG	
Pinuncaan	IMG_20240611_201114_1.jpg	jpg	
Pinuncaan	pinuncanaan (15).PNG	PNG	
Pinuncaan	IMG_5243.PNG	PNG	
Pinuncaan	IMG_5253.PNG	PNG	
Pinuncaan	pinuncanaan (18).PNG	PNG	
Pinuncaan	IMG_20240611_201119.jpg	jpg	
Pinuncaan	IMG_5383.PNG	PNG	
Pinuncaan	IMG_5254.PNG	PNG	
Pinuncaan	IMG_5263.PNG	PNG	
Pinuncaan	pinuncanaan (23).PNG	PNG	
Pinuncaan	IMG_5248.PNG	PNG	
Pinuncaan	IMG_5258.PNG	PNG	
Pinuncaan	IMG_5259.PNG	PNG	
Pinuncaan	IMG_5237.PNG	PNG	
Pinuncaan	IMG_20240611_201116.jpg	jpg	
Pinuncaan	IMG_5252.PNG	PNG	
Pinuncaan	IMG_5251.PNG	PNG	
Pinuncaan	IMG_20240611_201114.jpg	jpg	
Pinuncaan	pinuncanaan (14).PNG	PNG	
Pinuncaan	IMG_5260.PNG	PNG	
Pinuncaan	IMG_20240611_201109.jpg	jpg	
Pinuncaan	pinuncanaan (12).PNG	PNG	

Pinuncaan	IMG_20240611_201107.jpg	jpg
Pinuncaan	IMG_5255.PNG	PNG
Pinuncaan	pinuncanaan (17).PNG	PNG
Pinuncaan	IMG_5247.PNG	PNG
Pinuncaan	IMG_5250.PNG	PNG
Pinuncaan	IMG_20240611_201108.jpg	jpg
Pinuncaan	IMG_5236.PNG	PNG
Pinuncaan	IMG_5241.PNG	PNG
Pinuncaan	IMG_20240611_201114 4.jpg	jpg
Pinuncaan	IMG_5245.PNG	PNG
Pinuncaan	IMG_5264.PNG	PNG
Pinuncaan	IMG_5239.PNG	PNG
Pinuncaan	IMG_20240611_201114_11.jpg	jpg
Pinuncaan	IMG_5261.PNG	PNG
Pinuncaan	IMG_20240611_201108_2.jpg	jpg
Pinuncaan	IMG_5240.PNG	PNG
Pinuncaan	IMG_5256.PNG	PNG
Pinuncaan	IMG_5242.PNG	PNG
Pinuncaan	IMG_5249.PNG	PNG
Pinuncaan	pinuncanaan (20).PNG	PNG
Pinuncaan	pinuncanaan (22).PNG	PNG
Pinuncaan	IMG_20240611_201108_11.jpg	jpg
Pinuncaan	IMG_20240611_201108_22.jpg	jpg
Pinuncaan	pinuncanaan (16).PNG	PNG
Pinuncaan	pinuncanaan (21).PNG	PNG
Pinuncaan	pinuncanaan (19).PNG	PNG
Pinuncaan	IMG_20240611_201108_1.jpg	jpg
Pinuncaan	IMG_20240611_201107_11.jpg	jpg
Pinuncaan	IMG_5262.PNG	PNG
Pinuncaan	IMG_20240611_201107_1.jpg	jpg
Pinuncaan	IMG_4594.PNG	PNG
Pinuncaan	IMG_4591.PNG	PNG
Pinuncaan	IMG_4593.PNG	PNG
Ragi Hotang	IMG_20240611_211522_1.jpg	jpg
Ragi Hotang	IMG_20240705_231038_8.jpg	jpg
Ragi Hotang	IMG_20240705_231038_39.jpg	jpg
Ragi Hotang	IMG_20240705_231038_24.jpg	jpg

Ragi Hotang	IMG_20240705_231038_29.jpg	jpg
Ragi Hotang	IMG_20240611_211534.jpg	jpg
Ragi Hotang	IMG_20240705_231038_10.jpg	jpg
Ragi Hotang	IMG_20240705_231038_28.jpg	jpg
Ragi Hotang	IMG_20240705_231038_30.jpg	jpg
Ragi Hotang	IMG_20240705_231038_17.jpg	jpg
Ragi Hotang	IMG_20240705_231038_32.jpg	jpg
Ragi Hotang	IMG_20240705_231038_26.jpg	jpg
Ragi Hotang	IMG_20240611_211524.jpg	jpg
Ragi Hotang	IMG_20240611_211529.jpg	jpg
Ragi Hotang	IMG_20240611_211522.jpg	jpg
Ragi Hotang	IMG_20240705_231038_11.jpg	jpg
Ragi Hotang	IMG_20240705_231038_42.jpg	jpg
Ragi Hotang	IMG_20240611_211412.jpg	jpg
Ragi Hotang	IMG_20240611_211429.jpg	jpg
Ragi Hotang	IMG_20240611_211411.jpg	jpg
Ragi Hotang	IMG_20240705_231038_20.jpg	jpg
Ragi Hotang	IMG_20240611_211410.jpg	jpg
Ragi Hotang	IMG_20240611_211435.jpg	jpg
Ragi Hotang	IMG_20240705_231038_16.jpg	jpg
Ragi Hotang	IMG_20240611_211526.jpg	jpg
Ragi Hotang	IMG_20240705_231038_15.jpg	jpg
Ragi Hotang	IMG_20240705_231038_1.jpg	jpg
Ragi Hotang	IMG_20240705_231038_22.jpg	jpg
Ragi Hotang	IMG_20240705_231038_27.jpg	jpg
Ragi Hotang	IMG_20240611_211528.jpg	jpg
Ragi Hotang	IMG_20240705_231038_41.jpg	jpg
Ragi Hotang	IMG_20240611_211426.jpg	jpg
Ragi Hotang	IMG_20240705_231038_33.jpg	jpg
Ragi Hotang	IMG_20240705_231349.jpg	jpg
Ragi Hotang	IMG_20240705_231350.jpg	jpg
Ragi Hotang	IMG_20240705_231038_9.jpg	jpg
Ragi Hotang	IMG_20240705_231038_5.jpg	jpg
Ragi Hotang	IMG_20240611_211530.jpg	jpg
Ragi Hotang	IMG_20240705_231038_7.jpg	jpg
Ragi Hotang	IMG_20240705_231038_13.jpg	jpg
Ragi Hotang	IMG_20240705_231038_34.jpg	jpg

Ragi Hotang	IMG_20240611_211523.jpg	jpg
Ragi Hotang	IMG_20240705_231351.jpg	jpg
Ragi Hotang	IMG_20240611_211533.jpg	jpg
Ragi Hotang	IMG_20240611_211532.jpg	jpg
Ragi Hotang	IMG_20240705_231038_19.jpg	jpg
Ragi Hotang	IMG_20240705_231038_18.jpg	jpg
Ragi Hotang	IMG_20240705_231038_25.jpg	jpg
Ragi Hotang	IMG_20240705_231038_2.jpg	jpg
Ragi Hotang	IMG_20240611_211527.jpg	jpg
Ragi Hotang	IMG_20240705_231352.jpg	jpg
Ragi Hotang	IMG_20240705_231038_3.jpg	jpg
Ragi Hotang	IMG_20240705_231038_6.jpg	jpg
Ragi Hotang	IMG_20240705_231038_21.jpg	jpg
Ragi Hotang	IMG_20240705_231038_38.jpg	jpg
Ragi Hotang	IMG_20240611_211538.jpg	jpg
Ragi Hotang	IMG_20240611_211450.jpg	jpg
Ragi Hotang	IMG_20240705_231038_4.jpg	jpg
Ragi Hotang	IMG_20240705_231038_35.jpg	jpg
Ragi Hotang	IMG_20240705_231038_40.jpg	jpg
Ragi Hotang	IMG_20240611_211535.jpg	jpg
Ragi Hotang	IMG_20240705_231038_12.jpg	jpg
Ragi Hotang	IMG_20240611_211535_1.jpg	jpg
Ragi Hotang	IMG_20240611_211407.jpg	jpg
Ragi Hotang	IMG_20240705_231353.jpg	jpg
Ragi Hotang	IMG_20240611_211524_1.jpg	jpg
Ragi Hotang	IMG_20240705_231038_23.jpg	jpg
Ragi Hotang	IMG_20240705_231038_14.jpg	jpg
Ragi Hotang	IMG_20240611_211525.jpg	jpg
Ragi Hotang	IMG_20240611_211531.jpg	jpg
Ragi Hotang	IMG_20240705_231038_31.jpg	jpg
Sibolang	IMG-20240705-WA0094.jpg	jpg
Sibolang	1000104024.jpg	jpg
Sibolang	1000104047 - Copy.jpg	jpg
Sibolang	IMG-20240705-WA0049.jpg	jpg
Sibolang	IMG-20240705-WA0093 - Copy.jpg	jpg
Sibolang	IMG-20240705-WA0050 - Copy.jpg	jpg
Sibolang	IMG-20240705-WA0048.jpg	jpg

Sibolang		IMG-20240705-WA0076.jpg		jpg	
Sibolang		IMG-20240705-WA0093.jpg		jpg	
Sibolang		IMG-20240705-WA0061.jpg		jpg	
Sibolang		1000104046 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0064.jpg		jpg	
Sibolang		IMG-20240705-WA0051.jpg		jpg	
Sibolang		IMG-20240705-WA0071.jpg		jpg	
Sibolang		IMG-20240705-WA0050.jpg		jpg	
Sibolang		IMG-20240705-WA0058.jpg		jpg	
Sibolang		1000104023.jpg		jpg	
Sibolang		IMG-20240705-WA0097.jpg		jpg	
Sibolang		IMG-20240705-WA0072.jpg		jpg	
Sibolang		IMG-20240705-WA0080.jpg		jpg	
Sibolang		IMG-20240705-WA0089.jpg		jpg	
Sibolang		IMG-20240705-WA0062.jpg		jpg	
Sibolang		IMG-20240705-WA0067.jpg		jpg	
Sibolang		1000104050.jpg		jpg	
Sibolang		1000104046.jpg		jpg	
Sibolang		IMG-20240705-WA0059.jpg		jpg	
Sibolang		IMG-20240705-WA0054.jpg		jpg	
Sibolang		IMG-20240705-WA0100.jpg		jpg	
Sibolang		1000104025 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0092.jpg		jpg	
Sibolang		IMG-20240705-WA0082.jpg		jpg	
Sibolang		1000104048.jpg		jpg	
Sibolang		IMG-20240705-WA0086.jpg		jpg	
Sibolang		IMG-20240705-WA0049 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0099.jpg		jpg	
Sibolang		IMG-20240705-WA0065.jpg		jpg	
Sibolang		IMG-20240705-WA0053.jpg		jpg	
Sibolang		IMG-20240705-WA0094 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0056.jpg		jpg	
Sibolang		IMG-20240705-WA0095.jpg		jpg	
Sibolang		IMG-20240705-WA0077.jpg		jpg	
Sibolang		IMG-20240705-WA0084.jpg		jpg	
Sibolang		IMG-20240705-WA0075.jpg		jpg	
Sibolang		IMG-20240705-WA0052.jpg		jpg	

Sibolang		IMG-20240705-WA0081.jpg		jpg	
Sibolang		IMG-20240705-WA0074.jpg		jpg	
Sibolang		IMG-20240705-WA0078.jpg		jpg	
Sibolang		IMG-20240705-WA0069.jpg		jpg	
Sibolang		IMG-20240705-WA0083.jpg		jpg	
Sibolang		IMG-20240705-WA0073.jpg		jpg	
Sibolang		IMG-20240705-WA0060.jpg		jpg	
Sibolang		IMG-20240705-WA0048 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0068.jpg		jpg	
Sibolang		IMG-20240705-WA0087.jpg		jpg	
Sibolang		IMG-20240705-WA0055.jpg		jpg	
Sibolang		IMG-20240705-WA0063.jpg		jpg	
Sibolang		IMG-20240705-WA0057.jpg		jpg	
Sibolang		1000104024 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0092 - Copy.jpg		jpg	
Sibolang		IMG-20240705-WA0088.jpg		jpg	
Sibolang		IMG-20240705-WA0079.jpg		jpg	
Sibolang		IMG-20240705-WA0066.jpg		jpg	
Sibolang		IMG-20240705-WA0098.jpg		jpg	
Sibolang		IMG-20240705-WA0091.jpg		jpg	
Sibolang		IMG-20240705-WA0070.jpg		jpg	
Sibolang		IMG-20240705-WA0090.jpg		jpg	
Sibolang		1000104022.jpg		jpg	
Sibolang		1000104047.jpg		jpg	
Sibolang		1000104025.jpg		jpg	
Sibolang		IMG-20240705-WA0096.jpg		jpg	
Sibolang		IMG-20240705-WA0085.jpg		jpg	
Sadum		IMG_20240705_231559_46.jpg		jpg	
Sadum		sadum7.jpg		jpg	
Sadum		IMG_20240705_231538_40.jpg		jpg	
Sadum		IMG_20240705_231538_50.jpg		jpg	
Sadum		IMG_20240705_231818_1.jpg		jpg	
Sadum		sadum15.jpg		jpg	
Sadum		IMG_20240705_231816_1.jpg		jpg	
Sadum		sadum11.jpg		jpg	
Sadum		IMG_20240705_231538_41.jpg		jpg	
Sadum		sadum14.jpg		jpg	

	Sadum		IMG_20240705_231559_49.jpg		jpg	
	Sadum		IMG_20240705_231538_46.jpg		jpg	
	Sadum		IMG_20240705_231817.jpg		jpg	
	Sadum		IMG_20240705_231530_39.jpg		jpg	
	Sadum		sadum3.jpg		jpg	
	Sadum		sadum4.jpg		jpg	
	Sadum		IMG_20240705_231538_1.jpg		jpg	
	Sadum		IMG_20240705_231550_20.jpg		jpg	
	Sadum		sadum6.jpg		jpg	
	Sadum		IMG_20240611_193247.jpg		jpg	
	Sadum		sadum8.jpg		jpg	
	Sadum		IMG_20240705_231538_2.jpg		jpg	
	Sadum		IMG_20240705_231538_48.jpg		jpg	
	Sadum		sadum16.jpg		jpg	
	Sadum		IMG_20240705_231813.jpg		jpg	
	Sadum		IMG_20240705_231530_42.jpg		jpg	
	Sadum		IMG_20240705_231819.jpg		jpg	
	Sadum		IMG_20240611_193239.jpg		jpg	
	Sadum		IMG_20240705_231538_3.jpg		jpg	
	Sadum		IMG_20240705_231538_49.jpg		jpg	
	Sadum		sadum2.jpg		jpg	
	Sadum		sadum10.jpg		jpg	
	Sadum		IMG_20240705_231538_45.jpg		jpg	
	Sadum		IMG_20240705_231530_46.jpg		jpg	
	Sadum		sadum5.jpg		jpg	
	Sadum		IMG_20240705_231812.jpg		jpg	
	Sadum		IMG_20240705_231559_47.jpg		jpg	
	Sadum		IMG_20240611_193242.jpg		jpg	
	Sadum		sadum12.jpg		jpg	
	Sadum		IMG_20240705_231530_43.jpg		jpg	
	Sadum		IMG_20240705_231559_48.jpg		jpg	
	Sadum		IMG_20240705_231530_49.jpg		jpg	
	Sadum		IMG_20240705_231818.jpg		jpg	
	Sadum		IMG_20240611_193245.jpg		jpg	
	Sadum		IMG_20240705_231538_43.jpg		jpg	
	Sadum		sadum13.jpg		jpg	
	Sadum		IMG_20240705_231815_1.jpg		jpg	

Sadum	IMG_20240705_231530_45.jpg	jpg
Sadum	IMG_20240705_231816.jpg	jpg
Sadum	IMG_20240705_231814.jpg	jpg
Sadum	IMG_20240705_231538_47.jpg	jpg
Sadum	IMG_20240705_231530_47.jpg	jpg
Sadum	IMG_20240705_231814_1.jpg	jpg
Sadum	IMG_20240705_231530_50.jpg	jpg
Sadum	IMG_20240611_193241.jpg	jpg
Sadum	IMG_20240705_231815.jpg	jpg
Sadum	IMG_20240705_231550_18.jpg	jpg
Sadum	sadum9.jpg	jpg
Sadum	IMG_20240705_231530_40.jpg	jpg
Sadum	IMG_20240705_231550_19.jpg	jpg
Sadum	IMG_20240705_231559_50.jpg	jpg
Sadum	IMG_20240705_231538_44.jpg	jpg
Sadum	IMG_20240705_231530_38.jpg	jpg
Sadum	sadum.jpg	jpg
Sadum	IMG_20240705_231820.jpg	jpg
Sadum	IMG_20240611_193250.jpg	jpg
Sadum	IMG_20240705_231538_42.jpg	jpg
Sadum	IMG_20240705_231530_48.jpg	jpg
Ragi Hidup	IMG_5018.PNG	PNG
Ragi Hidup	IMG_5008.PNG	PNG
Ragi Hidup	IMG_5009.PNG	PNG
Ragi Hidup	IMG_4996.PNG	PNG
Ragi Hidup	IMG_4992.PNG	PNG
Ragi Hidup	IMG_4633.PNG	PNG
Ragi Hidup	IMG_4671.PNG	PNG
Ragi Hidup	IMG_20240611_201519.jpg	jpg
Ragi Hidup	IMG_4637.PNG	PNG
Ragi Hidup	IMG_5015.PNG	PNG
Ragi Hidup	IMG_5001.PNG	PNG
Ragi Hidup	IMG_4675.PNG	PNG
Ragi Hidup	IMG_5006.PNG	PNG
Ragi Hidup	IMG_5007.PNG	PNG
Ragi Hidup	IMG_4632.PNG	PNG
Ragi Hidup	IMG_4988.PNG	PNG

Ragi Hidup	IMG_20240611_201516.jpg	jpg
Ragi Hidup	IMG_20240611_201520.jpg	jpg
Ragi Hidup	IMG_5021.PNG	PNG
Ragi Hidup	IMG_4664.PNG	PNG
Ragi Hidup	IMG_4994.PNG	PNG
Ragi Hidup	IMG_4666.PNG	PNG
Ragi Hidup	IMG_4991.PNG	PNG
Ragi Hidup	IMG_4641.PNG	PNG
Ragi Hidup	IMG_5002.PNG	PNG
Ragi Hidup	IMG_5005.PNG	PNG
Ragi Hidup	IMG_5013.PNG	PNG
Ragi Hidup	IMG_4995.PNG	PNG
Ragi Hidup	IMG_5014.PNG	PNG
Ragi Hidup	IMG_4998.PNG	PNG
Ragi Hidup	IMG_5022.PNG	PNG
Ragi Hidup	IMG_4665.PNG	PNG
Ragi Hidup	IMG_4669.PNG	PNG
Ragi Hidup	IMG_4639.PNG	PNG
Ragi Hidup	IMG_4993.PNG	PNG
Ragi Hidup	IMG_4630.PNG	PNG
Ragi Hidup	IMG_4631.PNG	PNG
Ragi Hidup	IMG_4667.PNG	PNG
Ragi Hidup	IMG_4989.PNG	PNG
Ragi Hidup	IMG_5003.PNG	PNG
Ragi Hidup	IMG_4674.PNG	PNG
Ragi Hidup	IMG_20240611_201515.jpg	jpg
Ragi Hidup	IMG_4636.PNG	PNG
Ragi Hidup	IMG_4673.PNG	PNG
Ragi Hidup	IMG_4634.PNG	PNG
Ragi Hidup	IMG_4643.PNG	PNG
Ragi Hidup	IMG_5020.PNG	PNG
Ragi Hidup	IMG_4672.PNG	PNG
Ragi Hidup	IMG_5012.PNG	PNG
Ragi Hidup	IMG_4997.PNG	PNG
Ragi Hidup	IMG_4676.PNG	PNG
Ragi Hidup	IMG_5011.PNG	PNG
Ragi Hidup	IMG_4661.PNG	PNG

Ragi Hidup	IMG_5016.PNG	PNG
Ragi Hidup	IMG_4990.PNG	PNG
Ragi Hidup	IMG_4668.PNG	PNG
Ragi Hidup	IMG_5004.PNG	PNG
Ragi Hidup	IMG_4663.PNG	PNG
Ragi Hidup	IMG_4635.PNG	PNG
Ragi Hidup	IMG_4987.PNG	PNG
Ragi Hidup	IMG_4640.PNG	PNG
Ragi Hidup	IMG_4638.PNG	PNG
Ragi Hidup	IMG_5019.PNG	PNG
Ragi Hidup	IMG_5017.PNG	PNG
Ragi Hidup	IMG_4670.PNG	PNG
Ragi Hidup	IMG_5010.PNG	PNG
Ragi Hidup	IMG_4662.PNG	PNG
Ragi Hidup	IMG_4999.PNG	PNG
Ragi Hidup	IMG_4642.PNG	PNG
+-----+-----+-----+		
Jumlah total file dalam dataset: 1231		

b. Menelaah Data

Proses menelaah data ini dilakukan untuk menganalisa data secara eksploratif (*Exploratory Data Analysis*). Dalam tahap ini, kita akan mempelajari karakteristik gambar ulos yang ada untuk membantu memahami deskripsi data yang digunakan. Kode program untuk menampilkan karakteristik data dapat dilihat pada Tabel 3-6.

Tabel 3-6 Kode Program untuk Menampilkan Karakteristik Data

```
import os
import pandas as pd
from tabulate import tabulate

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan deskripsi data lengkap dengan karakteristik
def describe_dataset(dataset_path):
    file_details = []
```

```

# Menelusuri seluruh file dan folder
for root, dirs, files in os.walk(dataset_path):
    for file in files:
        if not file.startswith('.') and file.lower().endswith(('png', '.jpg', '.jpeg', '.tiff', '.bmp')):
            # Menentukan kategori berdasarkan nama folder
            category = os.path.basename(root)
            file_format = file.split('.')[-1] # Mendapatkan ekstensi file
            file_path = os.path.join(root, file) # Mendapatkan path file lengkap

            # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
            file_size = os.path.getsize(file_path) # Ukuran file dalam byte
            last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi

            # Menyimpan informasi dalam bentuk dictionary
            file_details.append({
                'Category': category,
                'File Name': file,
                'File Format': file_format,
                'File Size (bytes)': file_size,
                'Last Modified': last_modified
            })

# Mengubah hasil ke dalam bentuk DataFrame
file_df = pd.DataFrame(file_details)
return file_df

# Mendapatkan deskripsi dataset
file_df = describe_dataset(dataset_path)

# Menampilkan semua deskripsi dataset menggunakan tabulate
print(tabulate(file_df, headers='keys', tablefmt='pretty', showindex=False)) # Menampilkan semua baris

# Menampilkan jumlah total file yang dianalisis
print(f"\nJumlah total file dalam dataset: {len(file_df)}")

```

Karakteristik data yang didapatkan adalah kategori ulos, nama file, format file, ukuran file,

waktu modifikasi terakhir. Daftar karakteristik dapat dilihat pada Tabel 3-7.

Tabel 3-7 Karakteristik Data

Ragi Hidup				
Category	File Name	File Format	File Size (bytes)	Last Modified
Ragi Hidup	IMG_5125.PNG	PNG	671778	1734074881.1773717
Ragi Hidup	IMG_5158.PNG	PNG	1121774	1734074881.4493902
Sadum				
Category	File Name	File Format	File Size (bytes)	Last Modified
Sadum	IMG_20240705_231559_5.jpg	jpg	2255616	1734074887.4697974
Sadum	IMG_20240705_231559_36.jpg	jpg	2288496	1734074887.3857918
Sibolang				
Category	File Name	File Format	File Size (bytes)	Last Modified
Sibolang	WhatsApp Image 2024-07-05 at 23.49.23 (15) - Copy.jpeg	jpeg	64152	1734074887.7598171
Sibolang	1000104052.jpg	jpg	26437	1734074887.7358155
Ragi Hotang				
Category	File Name	File Format	File Size (bytes)	Last Modified
Ragi Hotang	IMG_20240705_231428.jpg	jpg	6093424	1734074885.395657
Ragi Hotang	IMG_20240611_211520_1.jpg	jpg	5211182	1734074882.92249

Pinuncaan				
Category	File Name	File Format	File Size (bytes)	Last Modified
Pinuncaan	IMG_20240611_2011255.jpg	jpg	5040439	1734074879.8712835
Pinuncaan	IMG_5327.PNG	PNG	633971	1734074880.1072996
Tumtuman				
Category	File Name	File Format	File Size (bytes)	Last Modified
Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (18).jpeg	jpeg	99543	1734074889.6109424
Tumtuman	ulos (16) - Copy.jpeg	jpeg	72031	1734074889.6359441
Jumlah total file dalam dataset: 1231				

Memahami keterkaitan antar data juga sangat penting. Menganalisis keterkaitan antar data yang digunakan ini membantu memastikan bahwa data yang digunakan tidak mengandung bias dan mewakili seluruh keragaman jenis ulos. Keterkaitan antar data pada proyek ini dianalisis dengan menggunakan Anova, Kendall's Tau, dan Chi-Squared. Kode untuk menampilkan keterkaitan data dapat dilihat pada Tabel 3-8.

Tabel 3-8 Kode Program untuk Menampilkan Keterkaitan Antar Data

```
import os
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency, kendalltau, f_oneway
from sklearn.feature_selection import mutual_info_classif
import seaborn as sns
import matplotlib.pyplot as plt
from tabulate import tabulate
```

```

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan deskripsi data lengkap dengan karakteristik
def describe_dataset(dataset_path):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('.png', '.jpg', '.jpeg', '.tiff', '.bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_path = os.path.join(root, file) # Mendapatkan path file lengkap

                # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
                file_size = os.path.getsize(file_path) # Ukuran file dalam byte
                last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi

                # Menyimpan informasi dalam bentuk dictionary
                file_details.append({
                    'Category': category,
                    'File Name': file,
                    'File Format': file_format,
                    'File Size (bytes)': file_size,
                    'Last Modified': last_modified
                })

    # Mengubah hasil ke dalam bentuk DataFrame
    file_df = pd.DataFrame(file_details)
    return file_df

# Mendapatkan deskripsi dataset
file_df = describe_dataset(dataset_path)

```

```

# Uji ANOVA: Uji perbedaan ukuran file antar kategori
categories = file_df['Category'].unique()
file_sizes_by_category = [file_df[file_df['Category'] == category]['File Size (bytes)'] for category in
categories]

# Melakukan uji ANOVA
f_stat, p_value = f_oneway(*file_sizes_by_category)

# Menampilkan hasil uji ANOVA
print(f"\nANOVA Test Results:")
print(f"F-statistic: {f_stat}, p-value: {p_value}")

if p_value < 0.05:
    print("Ada perbedaan signifikan dalam ukuran file antar kategori.")
else:
    print("Tidak ada perbedaan signifikan dalam ukuran file antar kategori.")

# Kendall's Tau: Mengukur asosiasi antara dua variabel ordinal (misalnya ukuran file dan kategori
numerik)
# Mengonversi kategori menjadi angka untuk analisis
category_mapping = {category: idx for idx, category in enumerate(file_df['Category'].unique())}
file_df['Category Numeric'] = file_df['Category'].map(category_mapping)

kendall_corr, kendall_p_value = kendalltau(file_df['File Size (bytes)'], file_df['Category Numeric'])

# Menampilkan hasil Kendall's Tau
print(f"\nKendall's Tau Test Results:")
print(f"Tau: {kendall_corr}, p-value: {kendall_p_value}")

if kendall_p_value < 0.05:
    print("Ada asosiasi signifikan antara ukuran file dan kategori.")
else:
    print("Tidak ada asosiasi signifikan antara ukuran file dan kategori.")

# Uji Chi-Squared: Uji independensi antara kategori dan format file
contingency_table = pd.crosstab(file_df['Category'], file_df['File Format'])
chi2, p, dof, expected = chi2_contingency(contingency_table)

```

```

print(f"\nChi-Squared Test Results:")
print(f"Chi-Squared: {chi2}, p-value: {p}")

if p < 0.05:
    print("Ada asosiasi signifikan antara kategori dan format file.")
else:
    print("Tidak ada asosiasi signifikan antara kategori dan format file.")

# Mutual Information: Mengukur seberapa banyak informasi yang dibagikan antara dua variabel
# Memerlukan konversi kategori menjadi numerik untuk ukuran file
from sklearn.preprocessing import LabelEncoder

# Encode kategori dan format file
label_encoder = LabelEncoder()
file_df['Category Encoded'] = label_encoder.fit_transform(file_df['Category'])
file_df['File Format Encoded'] = label_encoder.fit_transform(file_df['File Format'])

# Menghitung Mutual Information antara kategori dan ukuran file
X = file_df[['Category Encoded', 'File Format Encoded']]
y = file_df['File Size (bytes)']

mutual_info = mutual_info_classif(X, y)
print(f"\nMutual Information between 'Category' and 'File Size (bytes)': {mutual_info[0]}")

# Visualisasi untuk ANOVA dan Chi-Squared
# ANOVA Visualisasi: Distribusi ukuran file antar kategori
sns.boxplot(x='Category', y='File Size (bytes)', data=file_df)
plt.title('ANOVA: Ukuran File Antar Kategori')
plt.xticks(rotation=45)
plt.show()

# Chi-Squared Visualisasi: Tabel kontingensi antara kategori dan format file
sns.heatmap(contingency_table, annot=True, cmap="Blues", fmt="d")
plt.title('Chi-Squared: Kategori vs Format File')
plt.show()

```


Hasil dari keterkaitan antar data dapat dilihat pada Tabel 3-9.

Tabel 3-9 Keterkaitan Antar Data

ANOVA Test Results:

F-statistic: 173.15445738956262, p-value: 1.9097527024196994e-139

Ada perbedaan signifikan dalam ukuran file antar kategori.

Kendall's Tau Test Results:

Tau: -0.04412125023041999, p-value: 0.031794602025439316

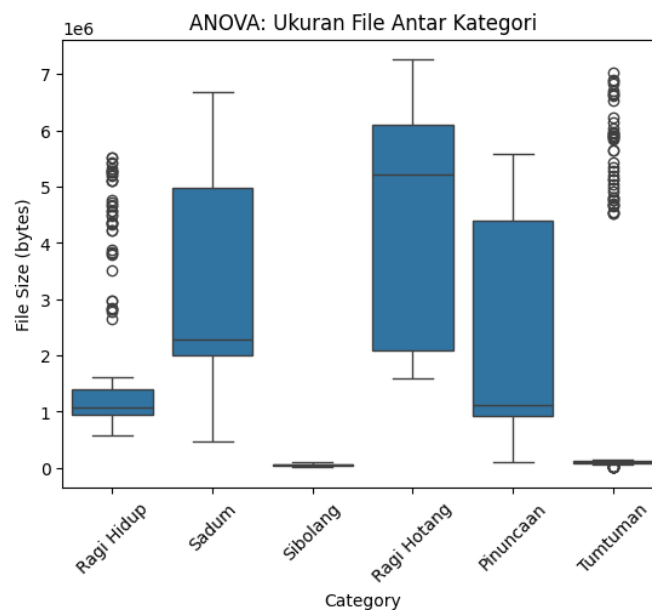
Ada asosiasi signifikan antara ukuran file dan kategori.

Chi-Squared Test Results:

Chi-Squared: 1575.0099207161215, p-value: 0.0

Ada asosiasi signifikan antara kategori dan format file.

Mutual Information between 'Category' and 'File Size (bytes)': 2.8846400016536338



Gambar 3-1 Grafik Hasil Test Anova

Grafik hasil uji ANOVA diatas menunjukkan distribusi ukuran file untuk setiap kategori ulos: Ragi Hidup, Sadum, Sibolang, Ragi Hotang, Pinunsaan, dan Tuntuman. Dari boxplot, terlihat bahwa kategori Ragi Hotang dan Sadum memiliki variasi ukuran file yang lebih

besar dibandingkan kategori lain, dengan rentang nilai yang luas dan beberapa outlier. Sementara itu, kategori Sibolang dan Tumtuman memiliki ukuran file yang cenderung lebih kecil dan stabil, dengan sedikit *outlier*. Ragi Hidup dan Pinuncaan menunjukkan ukuran file yang bervariasi, namun dengan distribusi yang lebih terpusat. Hasil ini menunjukkan adanya perbedaan signifikan dalam ukuran file antar kategori ulos.

c. Memvalidasi Data

Validasi data dilakukan untuk menilai kesesuaian kualitas data yang digunakan untuk klasifikasi motif ulos.

Untuk proses validasi data, dilakukan validasi terhadap ukuran data bertujuan untuk memastikan dataset cukup besar untuk pelatihan model yang efektif. Kode program untuk menampilkan ukuran data dapat dilihat pada Tabel 3-10.

Tabel 3-10 Kode Program untuk Menampilkan Ukuran Data

```
import os
import pandas as pd

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan deskripsi dataset dan validasi ukuran file
def validate_file_sizes(dataset_path, min_size=0, max_size=float('inf')):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('png', 'jpg', 'jpeg', 'tiff', 'bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_path = os.path.join(root, file) # Mendapatkan path file lengkap

                # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
                file_size = os.path.getsize(file_path) # Ukuran file dalam byte
                last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi
```

```

# Memvalidasi ukuran file apakah sesuai dengan batasan yang diberikan
size_status = "Valid"
if file_size < min_size:
    size_status = f"Too small ({file_size} bytes)"
elif file_size > max_size:
    size_status = f"Too large ({file_size} bytes)"

# Menyimpan informasi dalam bentuk dictionary
file_details.append({
    'Category': category,
    'File Name': file,
    'File Format': file_format,
    'File Size (bytes)': file_size,
    'Size Status': size_status,
    'Last Modified': last_modified
})

# Mengubah hasil ke dalam bentuk DataFrame
file_df = pd.DataFrame(file_details)
return file_df

# Menentukan batasan ukuran file (misalnya: min_size = 1000 bytes, max_size = 5 MB)
min_size = 1000 # minimal 1 KB
max_size = 5 * 1024 * 1024 # maksimal 5 MB

# Mendapatkan deskripsi dataset dan validasi ukuran file
file_df = validate_file_sizes(dataset_path, min_size, max_size)

# Menampilkan data yang valid (ukuran file sesuai) dan yang tidak valid
valid_files = file_df[file_df['Size Status'] == 'Valid']
invalid_files = file_df[file_df['Size Status'] != 'Valid']

# Menampilkan hasil
print("Valid Files:")
print(valid_files[['Category', 'File Name', 'File Size (bytes)', 'Size Status']])

```

```
print("\nInvalid Files:")
print(invalid_files[['Category', 'File Name', 'File Size (bytes)', 'Size Status']])
```

Hasil dari ukuran data dapat dilihat pada Tabel 3-11.

Tabel 3-11 Ukuran Data

Valid Files:

	Category	File Name \
0	Tumtuman	ulos (12) - Copy.jpeg
1	Tumtuman	IMG_20240611_202207_2.jpg
2	Tumtuman	WhatsApp Image 2024-07-06 at 00.02.57 (28).jpeg
3	Tumtuman	ulos (64) - Copy.jpeg
4	Tumtuman	ulos (1).jpeg
...
1226	Ragi Hidup	IMG_4670.PNG
1227	Ragi Hidup	IMG_5010.PNG
1228	Ragi Hidup	IMG_4662.PNG
1229	Ragi Hidup	IMG_4999.PNG
1230	Ragi Hidup	IMG_4642.PNG

	File Size (bytes)	Size Status
0	86566	Valid
1	4771524	Valid
2	103057	Valid
3	87360	Valid
4	124150	Valid
...
1226	973752	Valid
1227	1037800	Valid
1228	1178340	Valid
1229	1418735	Valid
1230	1071129	Valid

[1044 rows x 4 columns]

Invalid Files:

	Category	File Name	File Size (bytes) \
9	Tumtuman	IMG_20240611_202224.jpg	7022682
12	Tumtuman	IMG_20240611_202234.jpg	6836530
13	Tumtuman	IMG_20240611_202236.jpg	5870673
24	Tumtuman	IMG_20240611_202233.jpg	6651663
36	Tumtuman	IMG_20240611_202240.jpg	5280267
...
1146	Sadum	IMG_20240705_231814_1.jpg	5878053
1148	Sadum	IMG_20240611_193241.jpg	5275509
1149	Sadum	IMG_20240705_231815.jpg	5851359
1158	Sadum	IMG_20240705_231820.jpg	5855280
1169	Ragi Hidup	IMG_20240611_201519.jpg	5521263
Size Status			
9	Too large (7022682 bytes)		
12	Too large (6836530 bytes)		
13	Too large (5870673 bytes)		
24	Too large (6651663 bytes)		
36	Too large (5280267 bytes)		
...	...		
1146	Too large (5878053 bytes)		
1148	Too large (5275509 bytes)		
1149	Too large (5851359 bytes)		
1158	Too large (5855280 bytes)		
1169	Too large (5521263 bytes)		
[187 rows x 4 columns]			

Deskripsi statistik atribut yang ada pada proyek ini dicari dengan menggunakan kode yang ada pada Tabel 3-12.

Tabel 3-12 Kode Program untuk Menampilkan Deskripsi Statistik Atribut Data

```
import os
import pandas as pd

# Path dataset
```

```

dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan deskripsi dataset dan validasi ukuran file
def get_file_statistics(dataset_path):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('png', 'jpg', 'jpeg', 'tiff', 'bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_path = os.path.join(root, file) # Mendapatkan path file lengkap

                # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
                file_size = os.path.getsize(file_path) # Ukuran file dalam byte
                last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi

                # Menyimpan informasi dalam bentuk dictionary
                file_details.append({
                    'Category': category,
                    'File Name': file,
                    'File Format': file_format,
                    'File Size (bytes)': file_size,
                    'Last Modified': last_modified
                })

    # Mengubah hasil ke dalam bentuk DataFrame
    file_df = pd.DataFrame(file_details)
    return file_df

# Mendapatkan deskripsi dataset dan validasi ukuran file
file_df = get_file_statistics(dataset_path)

# Menampilkan deskripsi statistik dari atribut numerik
print("Deskripsi Statistik Atribut (Ukuran File dalam Byte):")

```

```

print(file_df['File Size (bytes)'].describe())

# Menampilkan statistik deskriptif kategori (kategori gambar)
print("\nFrekuensi Kategori:")
print(file_df['Category'].value_counts())

# Menampilkan statistik deskriptif terkait format file (ekstensi file)
print("\nFrekuensi Format File:")
print(file_df['File Format'].value_counts())

```

Deskripsi statistik atribut data yang dihasilkan dapat dilihat pada Tabel 3-13.

Tabel 3-13 Deskripsi Statistik Atribut Data

Deskripsi Statistik Atribut (Ukuran File dalam Byte):

```

count    1.231000e+03
mean     2.152272e+06
std      2.175859e+06
min      8.131000e+03
25%     1.017080e+05
50%     1.195456e+06
75%     4.300208e+06
max      7.253876e+06

```

Name: File Size (bytes), dtype: float64

Frekuensi Kategori:

Category

Ragi Hotang 209

Tumtuman 206

Sibolang 206

Ragi Hidup 205

Sadum 204

Pinuncaan 201

Name: count, dtype: int64

Frekuensi Format File:

File Format

jpg 703

PNG 306

```
jpeg    212
JPG      10
Name: count, dtype: int64
```

Selanjutnya untuk validasi data adalah melakukan analisis terhadap relasi antar atribut. Menganalisis relasi antar atribut (misalnya, hubungan antara warna dan jenis ulos) dapat membantu memahami bagaimana atribut-atribut tersebut memengaruhi klasifikasi. Relasi antar atribut dicari dengan kode pada Tabel 3-14.

Tabel 3-14 Kode Program untuk Menampilkan Relasi Antar Atribut Data

```
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan statistik file
def get_file_statistics(dataset_path):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('png', 'jpg', 'jpeg', 'tiff', 'bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)
                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_path = os.path.join(root, file) # Mendapatkan path file lengkap

                # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
                file_size = os.path.getsize(file_path) # Ukuran file dalam byte
                last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi

                # Menyimpan informasi dalam bentuk dictionary
                file_details.append({
```



```

        'Category': category,
        'File Name': file,
        'File Format': file_format,
        'File Size (bytes)': file_size,
        'Last Modified': last_modified
    })

# Mengubah hasil ke dalam bentuk DataFrame
file_df = pd.DataFrame(file_details)
return file_df

# Mendapatkan deskripsi dataset dan validasi ukuran file
file_df = get_file_statistics(dataset_path)

# Menampilkan korelasi antar atribut numerik
print("\nKorelasi antar atribut numerik (hanya ukuran file dan waktu modifikasi):")
numeric_columns = ['File Size (bytes)', 'Last Modified']
correlation_matrix = file_df[numeric_columns].corr()

# Menampilkan heatmap korelasi
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title('Heatmap Korelasi Antar Atribut Numerik')
plt.show()

# Menampilkan relasi antara kategori dan format file menggunakan Cross Tabulation
print("\nCross-Tabulation antara Kategori dan Format File:")
category_format_ct = pd.crosstab(file_df['Category'], file_df['File Format'])
print(category_format_ct)

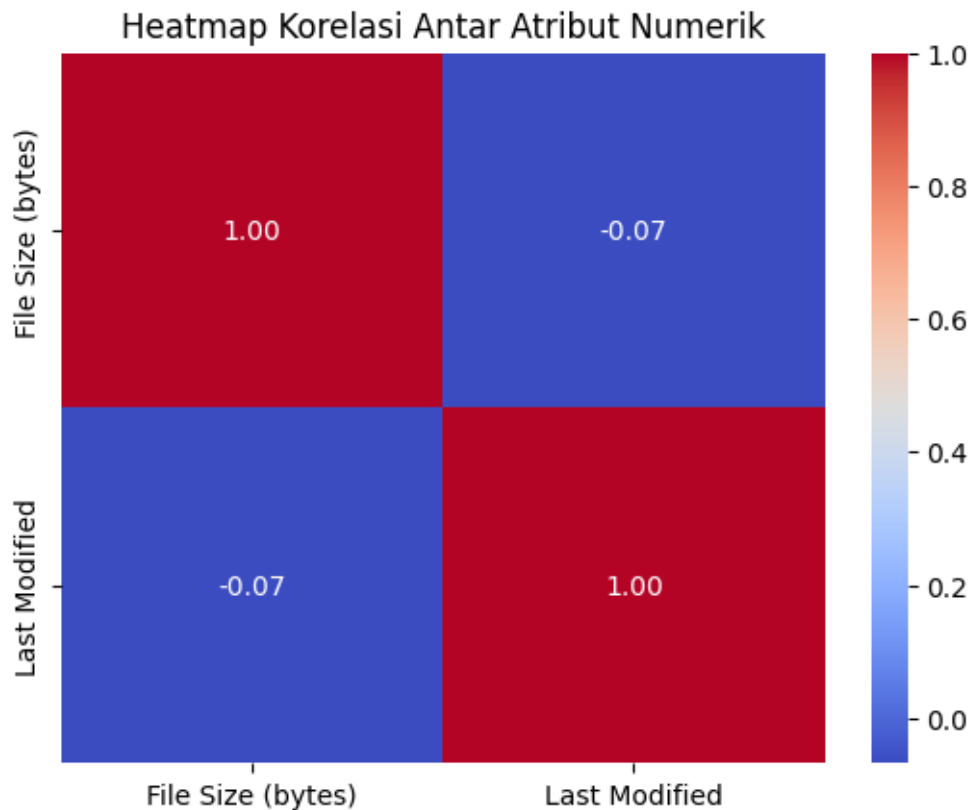
# Visualisasi Cross-Tabulation menggunakan Heatmap
sns.heatmap(category_format_ct, annot=True, cmap="Blues", fmt="d")
plt.title('Cross-Tabulation antara Kategori dan Format File')
plt.ylabel('Category')
plt.xlabel('File Format')
plt.show()

```

Hasil relasi antar atribut data pada proyek ini dapat dilihat pada Tabel 3-15.

Tabel 3-15 Relasi Antar Atribut Data

Korelasi antar atribut numerik (hanya ukuran file dan waktu modifikasi):



Gambar 3-2 Grafik Heatmap Korelasi Antar Atribut Numerik

Heatmap diatas menunjukkan korelasi antar atribut numerik dalam dataset, yaitu *File Size* (bytes) dan *Last Modified*. Nilai korelasi -0.07 antara kedua atribut menunjukkan korelasi yang sangat lemah dan negatif, artinya perubahan ukuran file tidak memiliki hubungan signifikan dengan waktu terakhir file dimodifikasi. Nilai diagonal 1.00 menunjukkan bahwa setiap atribut memiliki korelasi sempurna dengan dirinya sendiri. Secara keseluruhan, tidak ada keterkaitan yang berarti antara ukuran file dan waktu modifikasi dalam dataset ini.

Cross-Tabulation antara Kategori dan Format File:

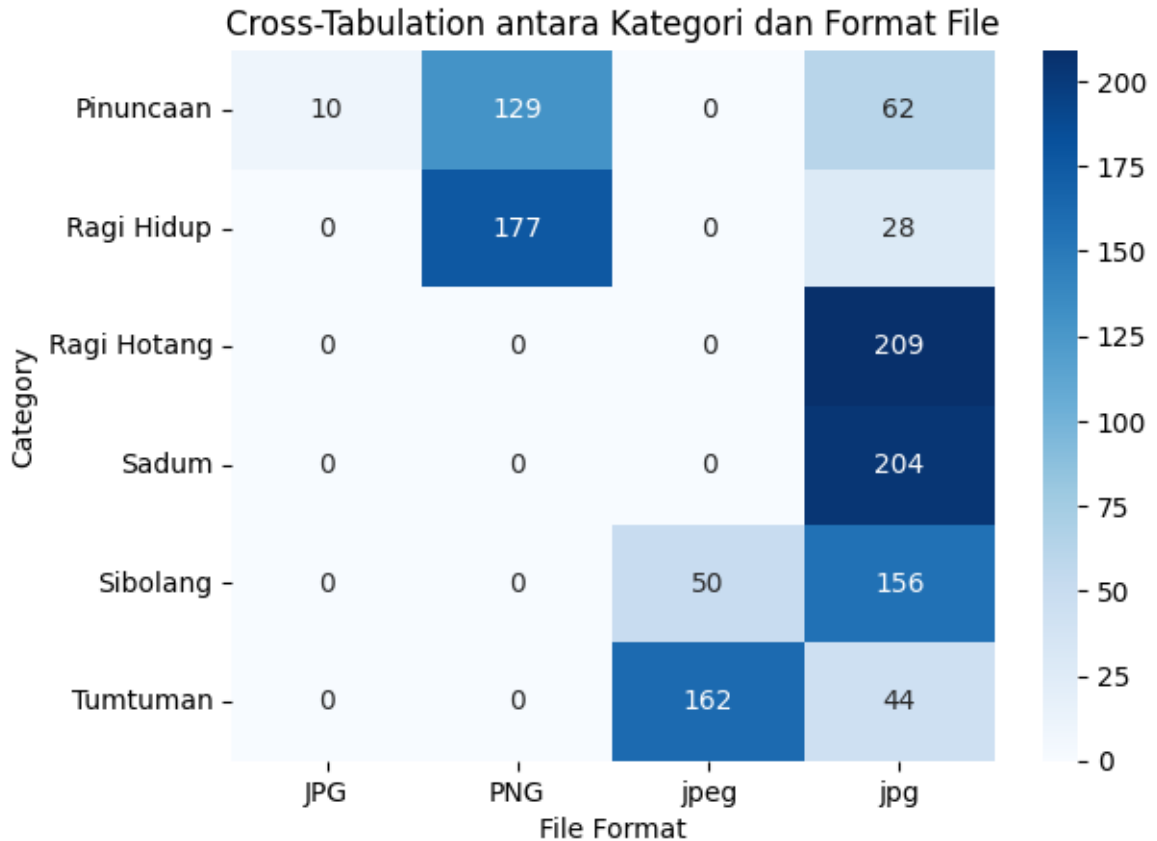
File Format JPG PNG jpeg jpg

Category

Pinunsaan 10 129 0 62

Ragi Hidup 0 177 0 28

Ragi Hotang 0 0 0 209
 Sadum 0 0 0 204
 Sibolang 0 0 50 156
 Tumtuman 0 0 162 44



Gambar 3-3 Grafik *Cross-Tabulation* Antara Kategori dan Format File

Cross-tabulation ini menunjukkan distribusi kategori berdasarkan format file. Kategori Pinuncaan memiliki file terbanyak dalam format PNG (129) dan sebagian dalam jpg (62) dan JPG (10). Ragi Hidup didominasi oleh PNG (177) dan sebagian jpg (28). Kategori Ragi Hotang dan Sadum seluruhnya menggunakan format jpg, masing-masing dengan 209 dan 204 file. Sibolang memiliki 50 file dalam format jpeg dan 156 dalam jpg, sementara Tumtuman didominasi oleh jpeg (162) dan sebagian jpg (44). Hasil ini menunjukkan bahwa distribusi format file tidak seragam antar kategori, dengan format jpg mendominasi sebagian besar kelas dan PNG atau jpeg lebih menonjol di kelas tertentu.

Visualisasi data, seperti histogram, scatter plot, atau heatmap, dapat memberikan gambaran yang jelas tentang distribusi data. Pada Tabel 3-16 merupakan kode yang digunakan untuk menampilkan visualisasi terhadap data yang digunakan untuk proyek ini.

Tabel 3-16 Kode Program untuk Menampilkan Visualisasi Data

```
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Fungsi untuk mendapatkan statistik file
def get_file_statistics(dataset_path):
    file_details = []

    # Menelusuri seluruh file dan folder
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            if not file.startswith('.') and file.lower().endswith(('png', 'jpg', 'jpeg', 'tiff', 'bmp')):
                # Menentukan kategori berdasarkan nama folder
                category = os.path.basename(root)

                file_format = file.split('.')[-1] # Mendapatkan ekstensi file
                file_path = os.path.join(root, file) # Mendapatkan path file lengkap

                # Mendapatkan karakteristik file: ukuran dan tanggal terakhir dimodifikasi
                file_size = os.path.getsize(file_path) # Ukuran file dalam byte
                last_modified = os.path.getmtime(file_path) # Timestamp terakhir dimodifikasi

                # Menyimpan informasi dalam bentuk dictionary
                file_details.append({
                    'Category': category,
                    'File Name': file,
                    'File Format': file_format,
                    'File Size (bytes)': file_size,
                    'Last Modified': last_modified
```

```

    })

    # Mengubah hasil ke dalam bentuk DataFrame
    file_df = pd.DataFrame(file_details)
    return file_df

# Mendapatkan deskripsi dataset dan validasi ukuran file
file_df = get_file_statistics(dataset_path)

# 1. Visualisasi distribusi ukuran file menggunakan Histogram
plt.figure(figsize=(10, 6))
sns.histplot(file_df['File Size (bytes)'], kde=True, color='skyblue')
plt.title('Distribusi Ukuran File')
plt.xlabel('Ukuran File (Bytes)')
plt.ylabel('Frekuensi')
plt.grid(True)
plt.show()

# 2. Boxplot untuk distribusi ukuran file
plt.figure(figsize=(8, 6))
sns.boxplot(x=file_df['File Size (bytes)'], color='lightgreen')
plt.title('Boxplot Ukuran File')
plt.xlabel('Ukuran File (Bytes)')
plt.show()

# 3. Visualisasi kategori file berdasarkan kategori menggunakan Barplot
plt.figure(figsize=(12, 6))
category_counts = file_df['Category'].value_counts()
sns.barplot(x=category_counts.index, y=category_counts.values, palette='muted')
plt.title('Jumlah File per Kategori')
plt.xlabel('Kategori')
plt.ylabel('Jumlah File')
plt.xticks(rotation=45)
plt.show()

# 4. Pie chart untuk distribusi format file
plt.figure(figsize=(8, 8))

```

```

file_format_counts = file_df['File Format'].value_counts()
plt.pie(file_format_counts, labels=file_format_counts.index, autopct='% 1.1f%%',
colors=sns.color_palette("Set3", len(file_format_counts)))
plt.title('Distribusi Format File')
plt.show()

```

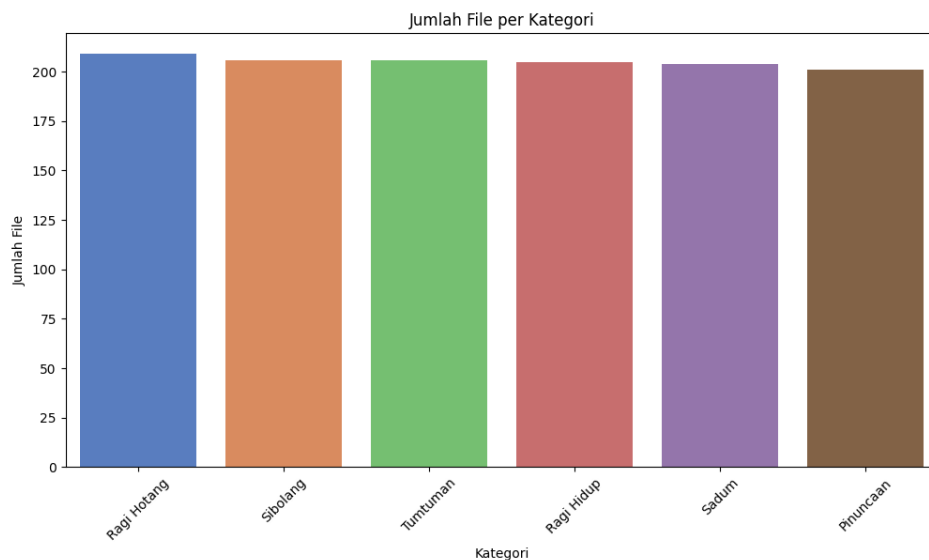
Hasil dari visualisasi data yang didapatkan dapat dilihat pada Tabel 3-17.

Tabel 3-17 Visualisasi Data

<ipython-input-13-fd6cf4df6e99>:61: FutureWarning:

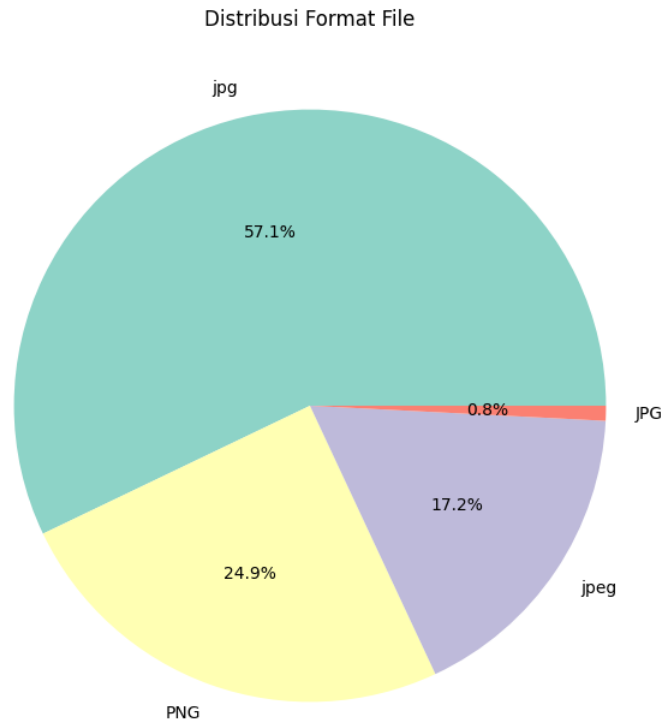
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=category_counts.index, y=category_counts.values, palette='muted')
```



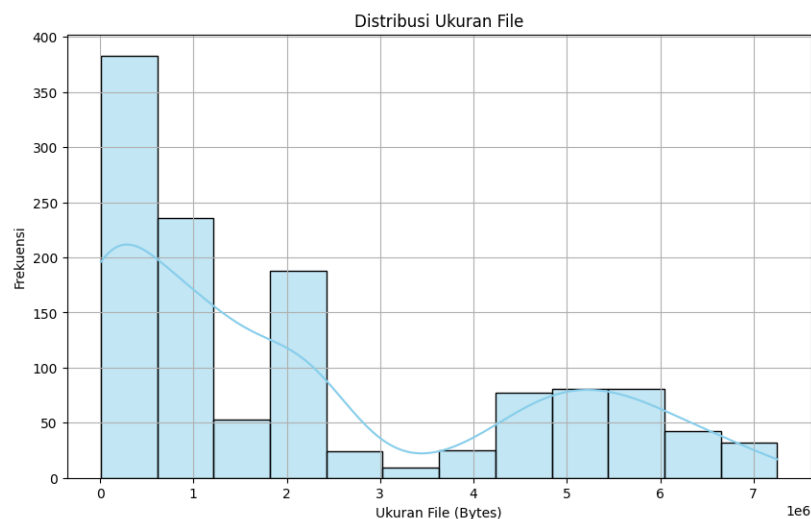
Gambar 3-4 Barchart Jumlah File per Kategori

Grafik diatas menunjukkan distribusi jumlah file per kategori dalam dataset. Setiap kategori (Ragi Hotang, Sibolang, Tumtuman, Ragi Hidup, Sadum, dan Pinuncaan) memiliki jumlah file yang hampir sama, yaitu sekitar 200 file per kategori. Distribusi yang seimbang ini memastikan bahwa model tidak mengalami bias terhadap salah satu kelas selama pelatihan, sehingga performa evaluasi lebih adil dan representatif. Kondisi ini ideal untuk membangun model klasifikasi yang akurat dan generalisasi yang baik.



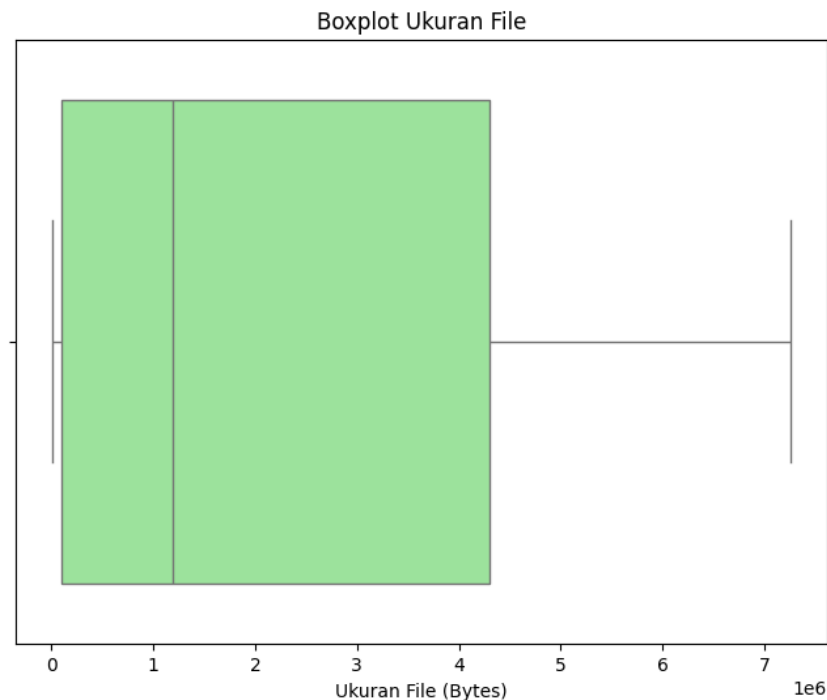
Gambar 3-5 Doughnut Chart Distribusi Format File

Diagram pie diatas menunjukkan distribusi format file dalam dataset. Mayoritas file menggunakan format jpg dengan proporsi 57.1%, diikuti oleh PNG sebesar 24.9% dan jpeg sebesar 17.2%. Sementara itu, format JPG hanya menyumbang 0.8% dari total file. Distribusi ini menunjukkan bahwa format file tidak sepenuhnya seragam, dengan dominasi format jpg, yang perlu diperhatikan dalam proses pra-pemrosesan agar semua format file dapat diproses secara konsisten oleh model.



Gambar 3-6 Gabungan Barchart dan Line Chart Distribusi Ukuran File

Histogram diatas menunjukkan distribusi ukuran file dalam dataset berdasarkan frekuensi. Mayoritas file berukuran kecil, sekitar 0 hingga 1 MB, dengan jumlah tertinggi sekitar 400 file. Seiring bertambahnya ukuran file hingga 2 MB dan seterusnya, frekuensi file menurun signifikan. Ukuran file terbesar mencapai 7 MB, tetapi hanya memiliki sedikit sampel. Distribusi ini menunjukkan bahwa sebagian besar file dalam dataset memiliki ukuran kecil hingga sedang, yang ideal untuk pemrosesan model tanpa mengorbankan performa komputasi.



Gambar 3-7 Grafik Boxplot Ukuran File

Boxplot diatas menunjukkan distribusi ukuran file dalam dataset. Nilai median ukuran file berada di sekitar 1 MB, menunjukkan bahwa setengah dari file memiliki ukuran kurang dari 1 MB. Rentang ukuran file cukup lebar, dimulai dari hampir 0 MB hingga sekitar 7 MB. Boxplot juga memperlihatkan bahwa mayoritas file berada dalam interkuartil range (IQR) antara 0 MB hingga 4 MB, dengan beberapa file yang lebih besar berada di luar rentang ini, namun tidak ada outlier ekstrem yang signifikan. Distribusi ini menegaskan bahwa sebagian besar file berukuran kecil hingga sedang, sementara beberapa file memiliki ukuran lebih besar namun masih dalam rentang yang wajar.

3.1.3 Data Preparation

Pada tahap *data preparation* (persiapan data), data akan disiapkan untuk pemodelan dengan beberapa langkah yang mencakup pembersihan, transformasi, dan seleksi fitur yang relevan. Persiapan data dilakukan untuk memperbaiki kualitas data untuk pemodelan. Beberapa langkah utama dalam tahap ini adalah:

a. Memilih dan Memilah Data

Pada langkah ini, data yang tersedia dipilih dan dipilah berdasarkan kategori atau label yang akan digunakan. Proses ini akan memproses record data dan atribut data yang terpakai. Dataset yang digunakan pada proses ini sudah terdiri dari folder "*Train*" dan "*Test*" yang berisi gambar sudah dikelompokkan kedalam kategori-kategori ulos tertentu. Kode program yang digunakan untuk pemilihan data dapat dilihat pada Tabel 3-18

Tabel 3-18 Kode untuk Memilih dan Memilah Data

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Direktori untuk kategori gambar
categories = ['Tumtuman', 'Pinuncean', 'Ragi Hotang', 'Sibolang', 'Sadum', 'Ragi Hidup']

def prepare_data(dataset_path, categories):
    data = []
    labels = []

    # Iterasi untuk folder 'Train' dan 'Test'
    for split in ['Train', 'Test']:
        split_path = os.path.join(dataset_path, split)

        for category in categories:
            category_path = os.path.join(split_path, category)
```

```

# Pastikan folder kategori ada
if os.path.exists(category_path):
    # Memilih gambar dari folder kategori
    for filename in os.listdir(category_path):
        if filename.lower().endswith(('png', 'jpg', 'jpeg')):
            img_path = os.path.join(category_path, filename)
            try:
                img = tf.keras.preprocessing.image.load_img(img_path, target_size=(128, 128)) # Resize
gambar
                img_array = tf.keras.preprocessing.image.img_to_array(img)
                data.append(img_array)
                labels.append(category)
            except Exception as e:
                print(f"Error loading image {img_path}: {e}")
        else:
            print(f"Folder not found: {category_path}")

# Konversi data dan labels ke array numpy
data = np.array(data)
labels = np.array(labels)

return data, labels

# Menyiapkan data
data, labels = prepare_data(dataset_path, categories)

# Normalisasi pixel gambar ke rentang [0, 1]
data = data / 255.0

# Membagi data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

# Menggunakan OneHotEncoder untuk mengubah label menjadi format numerik
encoder = LabelEncoder()
y_train = encoder.fit_transform(y_train)
y_test = encoder.transform(y_test)

```

```

# Menyusun data augmentation untuk meningkatkan variasi data pelatihan
train_datagen = ImageDataGenerator(
    rescale=1./255, # Normalisasi
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1./255)

# Menggunakan ImageDataGenerator untuk memproses data pelatihan
train_generator = train_datagen.flow(X_train, y_train, batch_size=32)
validation_generator = validation_datagen.flow(X_test, y_test, batch_size=32)

print("Data Preparation Completed")

```

Pada kode diatas, gambar sudah diubah ukurannya ke (128, 128) piksel, yang merupakan bentuk rekonstruksi atribut data. Selain itu, gambar juga dikonversi ke dalam bentuk array menggunakan `img_to_array`. Setiap label dari gambar motif ulos kemudian dikonversi menggunakan ***LabelEncoder***, dan digunakan *augmentation data* dengan menggunakan ***ImageDataGenerator*** untuk memingkatkan variasi dan kualitas model. Hasil akhir dari proses ini adalah data siap digunakan untuk pelatihan dan validasi model.

b. Membersihkan Data

Pada tahap ini, dilakukan pembersihan terhadap data gambar. Proses ini dilakukan untuk meminimalkan *noise* (tidak lengkap atau salah) pada data. Kode program untuk proses ini dapat dilihat pada Tabel 3-19.

Tabel 3-19 Kode Program untuk Membersihkan Data

```

import os
import numpy as np

```

```

import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Direktori untuk kategori gambar
categories = ['Tumtuman', 'Pinunnaan', 'Ragi Hotang', 'Sibolang', 'Sadum', 'Ragi Hidup']

def clean_data(dataset_path, categories, target_size=(128, 128)):
    data = []
    labels = []

    # Iterasi untuk folder 'Train' dan 'Test'
    for split in ['Train', 'Test']:
        split_path = os.path.join(dataset_path, split)

        for category in categories:
            category_path = os.path.join(split_path, category)

            # Pastikan folder kategori ada
            if os.path.exists(category_path):
                # Memilih gambar dari folder kategori
                for filename in os.listdir(category_path):
                    if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
                        img_path = os.path.join(category_path, filename)
                        try:
                            # Memuat dan meresize gambar
                            img = tf.keras.preprocessing.image.load_img(img_path, target_size=target_size)
                            img_array = tf.keras.preprocessing.image.img_to_array(img)
                            data.append(img_array)
                            labels.append(category)
                        except Exception as e:
                            print(f"Error loading image {img_path}: {e}")
                    else:
                        print(f"Folder not found: {category_path}")

```

```

# Konversi data dan labels ke array numpy
data = np.array(data)
labels = np.array(labels)

return data, labels

# Menyiapkan data
data, labels = clean_data(dataset_path, categories)

# Normalisasi pixel gambar ke rentang [0, 1]
data = data / 255.0

# Membagi data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

# Menggunakan OneHotEncoder untuk mengubah label menjadi format numerik
encoder = LabelEncoder()
y_train = encoder.fit_transform(y_train)
y_test = encoder.transform(y_test)

print(f"Data Cleaning Completed. Total images: {len(data)}")

```

Kode program diatas akan membaca folder "*train*" dan "*test*" dari dataset yang ditentukan. Gambar di setiap kategori di-load, *resize* ke ukuran 128x128 piksel, dan dikonversi menjadi array numerik. Data yang berhasil di-load dinormalisasi ke rentang [0, 1], dan label kategori diubah menjadi format numerik. Hasil akhirnya adalah data gambar yang telah terorganisasi, ternormalisasi, dan siap digunakan untuk melatih model.

c. Mengkonstruksi Data

Setelah data dibersihkan, dilakukan konstruksi dataset yaitu menambahkan *feature engineering* dan melakukan transformasi terhadap data (standarisasi dan normalisasi). Dataset yang sudah dikonstruksi kemudian akan digunakan untuk dibagi menjadi menjadi

set pelatihan dan pengujian. Berikut kode program pada Tabel 3-20 yang digunakan untuk mengkonstruksi data.

Tabel 3-20 Kode Program untuk Mengkonstruksi Data

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Direktori untuk kategori gambar
categories = ['Tumtuman', 'Pinuncean', 'Ragi Hotang', 'Sibolang', 'Sadum', 'Ragi Hidup']

def construct_data(dataset_path, categories, target_size=(128, 128)):
    data = []
    labels = []

    # Iterasi untuk folder 'Train' dan 'Test'
    for split in ['Train', 'Test']:
        split_path = os.path.join(dataset_path, split)

        for category in categories:
            category_path = os.path.join(split_path, category)

            if os.path.exists(category_path):
                # Memilih gambar dari folder kategori
                for filename in os.listdir(category_path):
                    if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
                        img_path = os.path.join(category_path, filename)
                        img = tf.keras.preprocessing.image.load_img(img_path, target_size=target_size) # Resize
gambar
                        img_array = tf.keras.preprocessing.image.img_to_array(img)
                        data.append(img_array)
                        labels.append(category)
```

```

# Konversi data dan labels ke array numpy
data = np.array(data)
labels = np.array(labels)

return data, labels

# Menyiapkan data
data, labels = construct_data(dataset_path, categories)

# Normalisasi pixel gambar ke rentang [0, 1]
data = data / 255.0

# Membagi data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

# Menggunakan LabelEncoder untuk mengubah label menjadi format numerik
encoder = LabelEncoder()
y_train = encoder.fit_transform(y_train)
y_test = encoder.transform(y_test)

# Menggunakan ImageDataGenerator untuk augmentasi dan validasi data
train_datagen = ImageDataGenerator(
    rescale=1./255, # Normalisasi
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1./255)

# Menyiapkan data generator untuk training dan validation
train_generator = train_datagen.flow(X_train, y_train, batch_size=32)

```

```
validation_generator = validation_datagen.flow(X_test, y_test, batch_size=32)

print(f"Data Construction Completed. Total images: {len(data)}")
```

Kode diatas akan menghasilkan *generator data training* dan validasi yang siap digunakan untuk melatih model.

d. Mengintegrasikan Data

Pada tahap ini, akan dilakukan penggabungan terhadap data pelatihan dan pengujian yang telah dipisahkan sebelumnya. Terdapat proses pemanfaatan augmentasi data untuk menambah variasi dalam dataset pelatihan, yang dapat membantu model untuk generalisasi lebih baik. Teknik augmentasi yang umum digunakan adalah rotasi, pemotongan, dan *flipping* gambar secara acak. Kode program yang digunakan dapat dilihat pada Tabel 3-21.

Tabel 3-21 Kode Program untuk Mengintegrasikan Data

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Path dataset
dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

# Direktori untuk kategori gambar
categories = ['Tumtuman', 'Pinuncaan', 'Ragi Hotang', 'Sibolang', 'Sadum', 'Ragi Hidup']

def integrate_data(dataset_path, categories, target_size=(128, 128)):
    data = []
    labels = []

    # Iterasi untuk folder 'Train' dan 'Test'
    for split in ['Train', 'Test']:
        split_path = os.path.join(dataset_path, split)
```



```

for category in categories:
    category_path = os.path.join(split_path, category)

    if os.path.exists(category_path):
        # Memilih gambar dari folder kategori
        for filename in os.listdir(category_path):
            if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
                img_path = os.path.join(category_path, filename)
                img = tf.keras.preprocessing.image.load_img(img_path, target_size=target_size) # Resize
gambar
                img_array = tf.keras.preprocessing.image.img_to_array(img)
                data.append(img_array)
                labels.append(category)

# Konversi data dan labels ke array numpy
data = np.array(data)
labels = np.array(labels)

return data, labels

# Menyiapkan data
data, labels = integrate_data(dataset_path, categories)

# Normalisasi pixel gambar ke rentang [0, 1]
data = data / 255.0

# Membagi data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

# Menggunakan LabelEncoder untuk mengubah label menjadi format numerik
encoder = LabelEncoder()
y_train = encoder.fit_transform(y_train)
y_test = encoder.transform(y_test)

# Menggunakan ImageDataGenerator untuk augmentasi dan validasi data
train_datagen = ImageDataGenerator(

```

```

rescale=1./255, # Normalisasi
rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1./255)

# Menyiapkan data generator untuk training dan validation
train_generator = train_datagen.flow(X_train, y_train, batch_size=32)
validation_generator = validation_datagen.flow(X_test, y_test, batch_size=32)

print(f"Data Integration Completed. Total images: {len(data)}")

```

3.1.4 Modeling

Pada sub bab ini, akan dilakukan pengembangan algoritma CNN untuk klasifikasi gambar, serta melatih model menggunakan data yang telah dipersiapkan sebelumnya. Berikut proses modeling yang dilakukan untuk mengklasifikasi motif ulos dengan algoritma CNN.

a. Membangun Model CNN

Pada langkah pertama ini, kita akan membangun arsitektur *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi gambar. Kode program dapat dilihat pada Tabel 3-22.

Tabel 3-22 Kode Program untuk Membangun Arsitektur CNN

```

import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers, models
from sklearn.metrics import classification_report, confusion_matrix

dataset_path = '/root/.cache/kagglehub/datasets/fthnaja/kain-ulos/versions/3'

```

```

train_dir = os.path.join(dataset_path, 'Train')
test_dir = os.path.join(dataset_path, 'Test')

# Direktori untuk kategori gambar
categories = ['Tumtuman', 'Pinuncaan', 'Ragi Hotang', 'Sibolang', 'Sadum', 'Ragi Hidup']

# Data Augmentation untuk latih
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

test_datagen = ImageDataGenerator(rescale=1./255)

# Data Generator untuk data latih
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical')

# Data Generator untuk data uji
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    shuffle=False)

```

Kode diatas digunakan untuk membangun model CNN (Convolutional Neural Network) yang berguna untuk mengklasifikasikan gambar kain ulos ke dalam enam kategori yang

berbeda. Dataset telah dibagi ke dalam direktori *Train* dan *Test*, masing-masing berisi gambar yang telah dikelompokkan berdasarkan kategori.

Bagian pertama mencakup augmentasi data menggunakan *ImageDataGenerator* untuk meningkatkan keragaman data latih, seperti rotasi, pergeseran, skala, dan flip horizontal, sehingga model lebih tahan terhadap overfitting. Gambar dinormalisasi dengan skala piksel 1/255. Selanjutnya, data latih dihasilkan dengan *flow_from_directory*, yang membaca gambar dari direktori latih, mengubah ukurannya menjadi 150x150 piksel, dan memuatnya dalam batch berukuran 32 untuk klasifikasi *categorical*. Data uji juga diproses serupa, tetapi tanpa augmentasi, dengan *shuffle* dimatikan untuk evaluasi akurat nantinya.

Selanjutnya, dilakukan proses pendefinisian arsitektur model CNN dengan kode program yang dapat dilihat pada Tabel 3-23.

Tabel 3-23 Kode Program untuk Mendefinisikan Arsitektur Model CNN

```
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(train_generator.num_classes, activation='softmax')
])

# Menampilkan ringkasan arsitektur model
model.summary()

print("Model Architecture Completed")
```

Kode diatas mendefinisikan arsitektur model CNN dengan menggunakan *Sequential* dari Keras untuk proses klasifikasi gambar kain ulos. Model dimulai dengan tiga lapisan konvolusi (32, 64, dan 128 filter) menggunakan ReLU, diikuti MaxPooling untuk ekstraksi

fitur dan pengurangan dimensi. Data hasil konvolusi diratakan dengan *Flatten* sebelum masuk ke lapisan *Dense* berukuran 128 unit dengan Dropout 0.5 untuk mencegah overfitting. Output layer menggunakan fungsi aktivasi *softmax* untuk klasifikasi ke jumlah kategori sesuai data. Ringkasan arsitektur ditampilkan dengan *model.summary()*.

b. Melatih Model CNN

Selanjutnya adalah melatih model, yang dilakukan dengan menggunakan kode program pada Tabel 3-24.

Tabel 3-24 Kode Program untuk Melatih Model CNN

```
# Kompilasi model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Kode diatas melakukan kompilasi dan melatih model untuk klasifikasi gambar kain ulos. Kompilasi model dilakukan menggunakan optimizer Adam, fungsi loss *categorical_crossentropy* untuk menangani klasifikasi multi-kelas, dan metrik evaluasi *accuracy*.

Selanjutnya, dilakukan pelatihan terhadap model CNN menggunakan data yang dihasilkan oleh *train_generator*. Fungsi *fit* menjalankan pelatihan selama 30 *epoch* (iterasi penuh melalui data latih), dengan jumlah langkah per *epoch* dihitung sebagai total sampel dibagi ukuran batch. Model secara bertahap memperbarui bobot berdasarkan data latih untuk meminimalkan kesalahan, dengan setiap *epoch* mencakup augmentasi data sesuai pengaturan pada *train_generator*. Hasil pelatihan, seperti akurasi dan loss, disimpan dalam variabel *history* untuk analisis lebih lanjut. Kode program untuk pelatihan dapat dilihat pada Tabel 3-24.

Tabel 3-25 Kode Program untuk Melatih Model CNN (2)

```
# Melatih model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
```

```
epochs=30
)
```

c. Menguji Model CNN.

Tahap selanjutnya adalah menguji model CNN. Pada proses ini akan dilakukan evaluasi terhadap performa model CNN pada data uji menggunakan `test_generator`. Fungsi `evaluate` menghitung *loss* dan akurasi model berdasarkan data uji, dengan jumlah langkah dihitung sebagai total sampel dibagi ukuran batch. Hasil evaluasi, berupa *loss* dan akurasi, dicetak untuk memberikan gambaran seberapa baik model dapat menggeneralisasi pada data yang belum pernah dilihat selama pelatihan. Kode program dapat dilihat pada Tabel 3-26.

Tabel 3-26 Kode Program untuk Memuat Data Uji

```
# Evaluasi model pada data uji
print("\nEvaluasi pada data uji:")
loss, accuracy = model.evaluate(test_generator, steps=test_generator.samples //
test_generator.batch_size)
print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
```

Hasil dari evaluasi pada data uji diatas adalah, didapatkan *test lost* sebesar 0.022556684911251068 dan *test accuracy* sebesar 0.9927884340286255.

Selanjutnya, dilakukan visualisasi terhadap hasil data uji dengan memplot akurasi dan loss selama setiap *epoch*. Kode program dapat dilihat pada Tabel 3-27.

Tabel 3-27 Kode Program untuk Membuat Plot Hasil Data Uji

```
# Plot hasil evaluasi
plt.figure(figsize=(12, 6))

# Plot akurasi
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
if 'val_accuracy' in history.history:
    plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
```

```

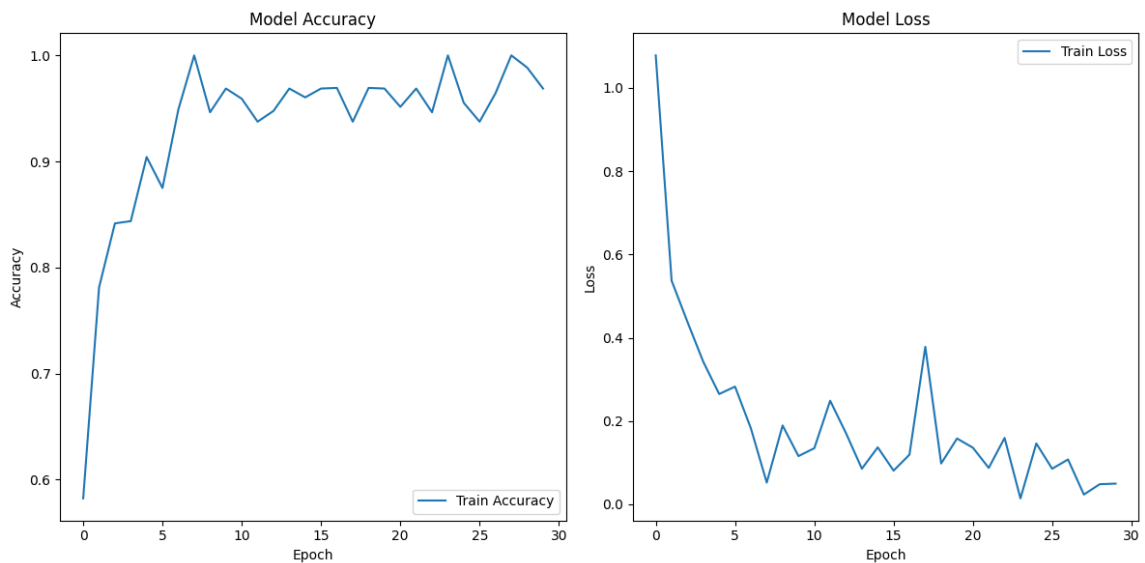
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
if 'val_loss' in history.history:
    plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()

```

Plot data uji yang dihasilkan dapat dilihat pada Gambar 3-8 Hasil Plot Data Uji



Gambar 3-8 Hasil Plot Data Uji

Visualisasi diatas menunjukkan bahwa *accuracy model* meningkat di awal proses *epoch* dan mendekati nilai maksimum setelah beberapa iterasi, sementara *loss model* turun signifikan di awal *epoch*, menunjukkan penurunan kesalahan prediksi model seiring dengan proses pelatihan. Ini menunjukkan bahwa model memiliki performa sangat baik pada data latih dengan akurasi tinggi dan loss rendah.

Selanjutnya, dilakukan proses prediksi model untuk data uji dengan menyaring hasil berdasarkan ambang batas keyakinan (confidence threshold) sebesar 0.7. Model memprediksi probabilitas kelas untuk setiap gambar, dan kelas dengan probabilitas tertinggi. Kode program dapat dilihat pada Tabel 3-28.

Tabel 3-28 Kode Program untuk Menyaring Klasifikasi untuk Gambar Ulos

```
# Menyaring klasifikasi hanya untuk gambar ulos
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)

true_classes = test_generator.classes
class_labels = list(test_generator.class_indices.keys())

print("\nLaporan Klasifikasi:")
print(classification_report(true_classes, predicted_classes, target_names=class_labels))
```

Hasil dari proses diatas adalah sebagai berikut:

```
Laporan Klasifikasi:

      precision  recall f1-score  support

Pinuncean      0.99    1.00    0.99         67
Ragi Hidup      1.00    1.00    1.00         69
Ragi Hotang      0.97    1.00    0.99         71
  Sadum          1.00    0.97    0.99         68
Sibolang         1.00    1.00    1.00         71
Tumtuman         1.00    0.99    0.99         70
```


accuracy		0.99		416
macro avg	0.99	0.99	0.99	416
weighted avg	0.99	0.99	0.99	416

Hasil diatas menunjukkan performa model CNN yang sangat baik dengan akurasi 99% pada data uji yang terdiri dari enam kelas: Pinuncean, Ragi Hidup, Ragi Hotang, Sadum, Sibolang, dan Tumtuman. Metrik *precision*, *recall*, dan *f1-score* untuk setiap kelas berada di kisaran 0.97 hingga 1.00, menandakan bahwa model mampu mengklasifikasikan gambar dengan tingkat kesalahan yang sangat rendah. Nilai *support* menunjukkan jumlah sampel di setiap kelas, dengan distribusi yang seimbang. Secara keseluruhan, performa tinggi ini mengindikasikan bahwa model dapat mengenali pola dengan baik dan memiliki generalisasi yang kuat pada data uji.

Evaluasi terakhir yang dilakukan adalah mencari nilai akurasi, presisi, *recall*, dan *F-1 Score*. Kode program disajikan pada Tabel 3-29.

Tabel 3-29 Hasil Evaluasi Model

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Menghitung accuracy
accuracy = accuracy_score(true_classes, predicted_classes)
print(f"Accuracy: {accuracy:.4f}")

# Precision, Recall, dan F1-Score untuk multi-class classification
precision = precision_score(true_classes, predicted_classes, average='macro') #
Average 'macro' untuk multi-class
recall = recall_score(true_classes, predicted_classes, average='macro')
f1 = f1_score(true_classes, predicted_classes, average='macro')

print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")
```

Hasil yang didapatkan dari proses diatas menunjukkan performa model CNN yang sangat baik dengan akurasi 99.28%, menandakan hampir semua prediksi pada data uji benar. *Precision* 99.30% menunjukkan model sangat akurat dalam memprediksi kelas positif, sedangkan *recall* 99.27% mengindikasikan model mampu mendeteksi hampir semua sampel yang benar sesuai kelas aslinya. Nilai *F1-score* 99.28%, sebagai rata-rata harmonis antara precision dan recall, memperkuat bahwa model seimbang dalam prediksi dan memiliki generalisasi yang optimal pada data uji.

Langkah terakhir adalah membuat visualisasi confusion matrix dan matriks heatmap untuk perhitungan yang kodenya dapat dilihat pada **Error! Reference source not found..**

Tabel 3-30 Kode Program Untuk Membuat Matriks Hasil Akhir

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
import numpy as np

# Menghitung confusion matrix
cm = confusion_matrix(true_classes, predicted_classes)

# Menghitung TP, TN, FP, FN untuk setiap kelas
tp = np.diag(cm) # Diagonal adalah True Positives
fp = cm.sum(axis=0) - tp # Kolom sum dikurangi TP untuk FP
fn = cm.sum(axis=1) - tp # Baris sum dikurangi TP untuk FN
tn = cm.sum() - (fp + fn + tp) # Total sum dikurangi TP, FP, FN untuk TN

# Visualisasi Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
x ticklabels=test_generator.class_indices.keys(),
y ticklabels=test_generator.class_indices.keys())
```

```

plt.title("Confusion Matrix")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.show()

# Visualisasi TP, TN, FP, FN untuk setiap kelas
labels = test_generator.class_indices.keys() # Nama-nama kelas
bar_width = 0.2
index = np.arange(len(labels))

# Menghitung confusion matrix
cm = confusion_matrix(true_classes, predicted_classes)

# Menghitung TP, TN, FP, FN untuk setiap kelas
tp = np.diag(cm) # Diagonal adalah True Positives
fp = cm.sum(axis=0) - tp # Kolom sum dikurangi TP untuk FP
fn = cm.sum(axis=1) - tp # Baris sum dikurangi TP untuk FN
tn = cm.sum() - (fp + fn + tp) # Total sum dikurangi TP, FP, FN untuk TN

# Membuat matriks TP, TN, FP, FN
metrics_matrix = np.array([tp, tn, fp, fn]).T # Transpose agar tiap baris mewakili kelas
metrics_labels = ["True Positive", "True Negative", "False Positive", "False Negative"]

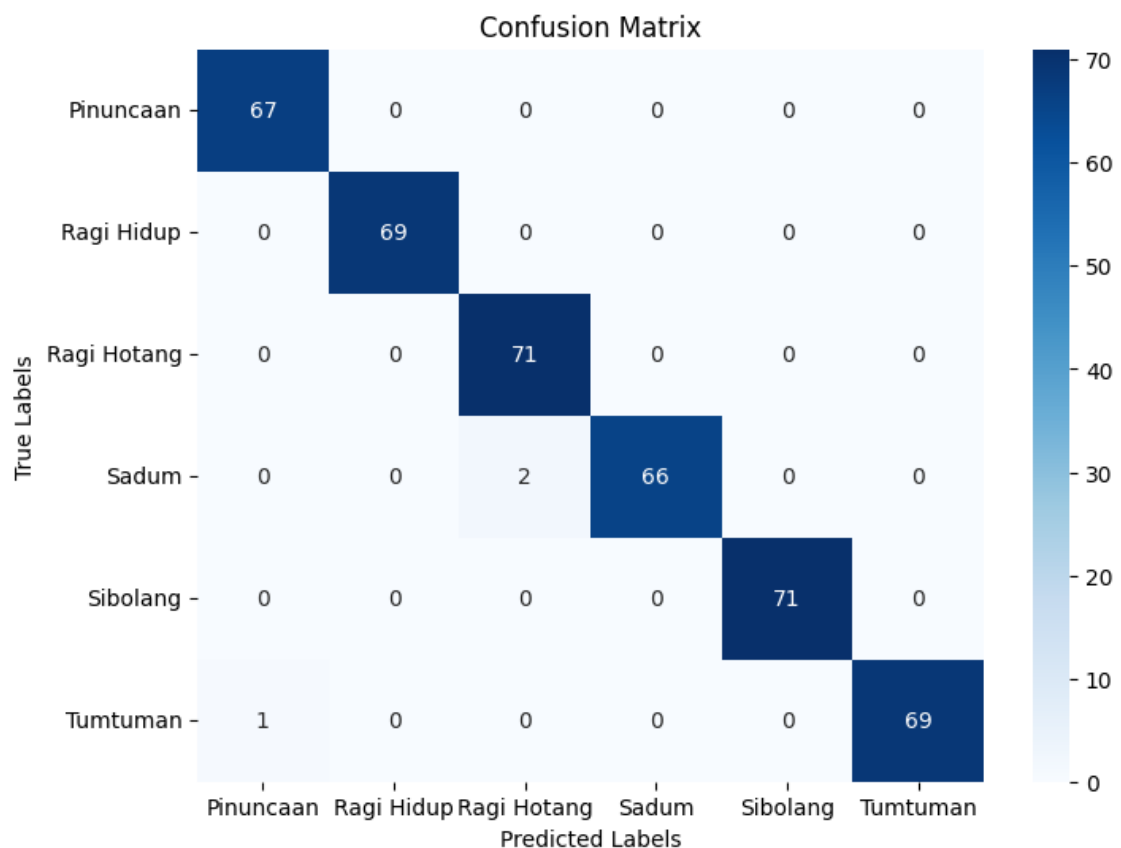
# Membuat heatmap untuk TP, TN, FP, FN per kelas
plt.figure(figsize=(10, 6))
sns.heatmap(metrics_matrix, annot=True, fmt='d', cmap="coolwarm",
            xticklabels=metrics_labels, yticklabels=test_generator.class_indices.keys())

# Menambahkan judul dan label
plt.title("TP, TN, FP, FN per Class")

```

```
plt.xlabel("Metrics")
plt.ylabel("Classes")
plt.tight_layout()
plt.show()
```

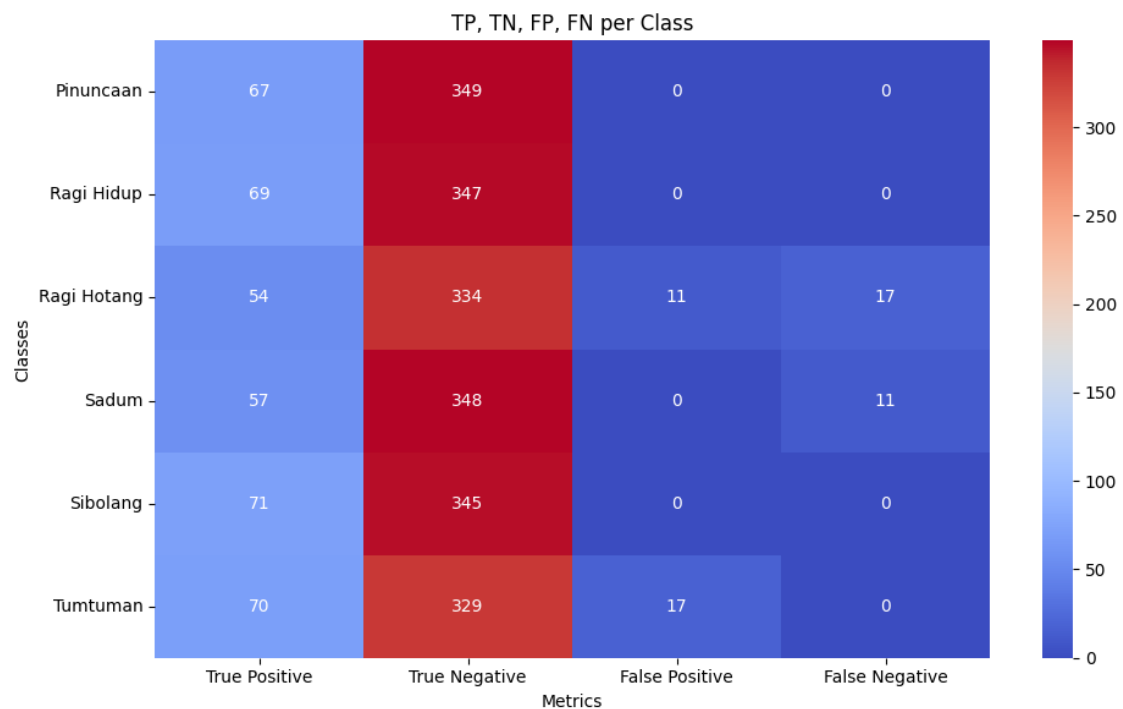
Kode diatas menghasilkan matrix yang menunjukkan perbandingan antara hasil prediksi model dengan label sebenarnya, yang nantinya akan digunakan untuk membantu mengevaluasi kinerja model. Visualisasi ditampilkan dengan menggunakan `seaborn.heatmap` untuk melihat distribusi prediksi di tiap kelas. Gambar dapat dilihat pada Gambar 3-9 dan Gambar 3-10.



Gambar 3-9 Confusion Matrix

Confusion matrix diatas menunjukkan performa model klasifikasi pada enam kelas: Pinuncaan, Ragi Hidup, Ragi Hotang, Sadum, Sibolang, dan Tumtuman. Angka di diagonal

utama menunjukkan jumlah prediksi yang benar untuk setiap kelas, seperti Pinuncean dengan 67 prediksi benar, Ragi Hidup 69, Ragi Hotang 71, Sadum 66, Sibolang 71, dan Tumtuman 69. Secara umum, model memiliki akurasi tinggi dengan hanya 3 kesalahan prediksi dari total 416 sampel. Kelas Sadum memiliki 2 sampel yang salah diklasifikasikan sebagai kelas lain, sementara kelas Tumtuman memiliki 1 kesalahan klasifikasi ke kelas Pinuncean. Semua kelas lainnya (Pinuncean, Ragi Hidup, Ragi Hotang, dan Sibolang) diprediksi dengan sempurna. Hasil ini mencerminkan kemampuan model yang kuat untuk mengenali pola visual dari sebagian besar kategori, dengan kesalahan minimal pada kelas tertentu.



Gambar 3-10 Grafik Heatmap

Heatmap ini menunjukkan performa model berdasarkan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) untuk setiap kelas. Kelas Pinuncean, Ragi Hidup, dan Sibolang memiliki TP tinggi tanpa kesalahan prediksi (FP dan FN = 0), menunjukkan performa sempurna. Namun, kelas Ragi Hotang memiliki 11 FP dan 17 FN, serta kelas Tumtuman memiliki 17 FP, yang mengindikasikan model sering salah mengenali atau memprediksi sampel dari kelas ini. Kelas Sadum juga memiliki 11 FN, menunjukkan beberapa sampel gagal diklasifikasikan dengan benar. Secara keseluruhan,

model bekerja baik, tetapi perlu perbaikan pada kelas Ragi Hotang, Sadum, dan Tumtuman untuk meningkatkan akurasi dan mengurangi kesalahan.

Terakhir, model akan disimpan dan diunduh dengan menggunakan kode program pada Tabel 3-31.

Tabel 3-31 Kode Program untuk Menyimpan Model

```
# Menyimpan model
model.save('model_ulos.h5')
print("Model telah disimpan sebagai 'model_ulos.h5'.")

from google.colab import files
files.download('model_ulos.h5')
```

3.1.6 Deployment

Pada tahap deployment, pengembangan dilakukan dengan memanfaatkan Streamlit untuk mengunggah gambar Ulos secara mudah dan mendapatkan prediksi dari kelas gambar yang diunggah sesuai dengan model algoritma CNN yang sudah dibangun. Streamlit *dideploy* pada Streamlit *Cloud* dan dapat diakses pada <https://motifulos.streamlit.app/>.

Streamlit ini dirancang dengan antarmuka yang sederhana dan ramah pengguna, di mana pengguna hanya perlu mengunggah gambar ulos yang ingin diprediksi melalui formulir yang telah disediakan di halaman utama. Setelah gambar berhasil diunggah, sistem akan memproses gambar tersebut menggunakan model CNN yang telah dilatih, kemudian menghasilkan prediksi kelas ulos yang paling sesuai. Prediksi kelas ini akan ditampilkan langsung kepada pengguna. Selain itu, aplikasi juga akan menampilkan grafik yang menunjukkan tingkat kepercayaan (*confidence level*) model terhadap prediksi yang diberikan, memberikan gambaran tentang seberapa akurat prediksi tersebut.

Selain hasil prediksi dan grafik, aplikasi juga menyediakan penjelasan mengenai desain dan kegunaan ulos yang diunggah. Penjelasan ini mencakup informasi budaya, sejarah, serta makna dari desain ulos tersebut. Setiap jenis ulos memiliki karakteristik dan kegunaan khusus dalam budaya Batak, seperti digunakan dalam acara adat, pernikahan, dan sebagai simbol status atau keberuntungan. Dengan informasi ini, aplikasi tidak hanya memberikan hasil prediksi, tetapi juga

berfungsi sebagai sumber edukasi untuk meningkatkan pemahaman pengguna mengenai warisan budaya Indonesia, khususnya ulos.

Kode program Pembangunan aplikasi dapat dilihat pada Tabel 3-32 Kode Program untuk Membangun Streamlit

Tabel 3-32 Kode Program untuk Membangun Streamlit

```
import streamlit as st
from PIL import Image
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

# Fungsi untuk memuat model CNN
def load_model():
    model = tf.keras.models.load_model('Deployment/model_ulos.h5')
    return model

# Fungsi untuk melakukan prediksi pada gambar
def predict_image(model, image):
    # Preprocessing gambar agar sesuai dengan input model
    image = image.resize((150, 150)) # Sesuaikan ukuran input dengan model
    image_array = np.array(image) / 255.0 # Normalisasi
    image_array = np.expand_dims(image_array, axis=0) # Tambahkan batch dimension

    # Prediksi
    prediction = model.predict(image_array)
    return prediction

# Fungsi validasi awal untuk memastikan gambar adalah ulos
def validate_image(model, image, threshold=0.5):
    prediction = predict_image(model, image)
    max_confidence = np.max(prediction)
```

```

    if max_confidence < threshold: # Jika confidence di bawah threshold, bukan ulos
        return False, max_confidence
    return True, max_confidence

# Fungsi untuk menampilkan visualisasi confidence level
def plot_confidence(prediction, class_names):
    fig, ax = plt.subplots()
    ax.barh(class_names, prediction[0], color='skyblue')
    ax.set_xlabel('Confidence')
    ax.set_title('Prediction Confidence Level')
    st.pyplot(fig)

# Streamlit App
st.set_page_config(layout="wide")
st.title("Ulos Image Classification")

# Sidebar untuk navigasi
page = st.sidebar.radio("Pilih Fitur", ["Klasifikasi Gambar", "Panduan Pengguna"])

if page == "Klasifikasi Gambar":
    st.markdown(
        """
        **Tentang Aplikasi:**

        Aplikasi ini menggunakan model Convolutional Neural Network (CNN) untuk
        mengklasifikasikan jenis kain Ulos berdasarkan gambar yang diunggah pengguna. Model
        ini bertujuan untuk mendukung pelestarian budaya dan meningkatkan pemahaman tentang kain
        tradisional ulos.
        """
    )

# Upload file gambar

```



```

gu_image = st.file_uploader("Upload an image of Ulos", type=["jpg", "jpeg", "png"])

if gu_image is not None:
    # Tampilkan gambar yang diunggah
    image = Image.open(gu_image)
    st.image(image, caption="Uploaded Image", use_container_width=True)

    # Muat model dan lakukan prediksi
    st.write("Processing image...")
    try:
        model = load_model()

        # Validasi gambar sebagai motif ulos
        is_ulos, confidence = validate_image(model, image)
        if not is_ulos:
            st.error(
                f"Gambar yang diunggah tidak dikenali sebagai motif ulos. Confidence:
{confidence:.2f}"
            )
            st.write("Model hanya mendukung klasifikasi gambar ulos dengan tingkat kepercayaan
yang memadai.")
            st.stop() # Menghentikan eksekusi jika gambar bukan ulos

        # Lanjutkan prediksi jika validasi berhasil
        prediction = predict_image(model, image)

        # Validasi hasil prediksi
        class_names = ['Pinuncaan', 'Ragi Hidup', 'Ragi Hotang', 'Sadum', 'Sibolang',
'Tumtuman'] # Label kelas
        if len(prediction[0]) == len(class_names):
            predicted_class = class_names[np.argmax(prediction)]

```

```

confidence = np.max(prediction) * 100

st.write(f"**Predicted Class:** {predicted_class}")
st.write(f"**Confidence:** {confidence:.2f}%")

# Tampilkan visualisasi confidence level
plot_confidence(prediction, class_names)

# Deskripsi tambahan tentang ulos yang diprediksi
ulos_descriptions = {
    "Pinuncaan": {
        "Desain": "Ulos ini memiliki struktur yang terdiri dari lima bagian yang ditenun
secara terpisah dan kemudian disatukan. Motifnya biasanya menggunakan warna-warna cerah
dengan pola geometris yang khas.",
        "Kegunaan": [
            "Acara Resmi: Sering digunakan dalam upacara adat dan acara resmi oleh
pemimpin atau raja.",
            "Pernikahan: Dipakai oleh pengantin dan keluarga dalam perayaan pernikahan.",
            "Marpaniaran: Digunakan saat pesta besar dalam acara marpaniaran.",
            "Simbol Kehormatan: Melambangkan status dan kehormatan bagi pemakainya."
        ]
    },
    "Ragi Hidup": {
        "Desain": "Ulos ini berbentuk panjang dan lebar, dengan pola sederhana namun
elegant.",
        "Kegunaan": [
            "Pakaian Sehari-hari: Digunakan sebagai baju atau sarung untuk kenyamanan.",
            "Simbol Kehidupan: Melambangkan kehidupan dan keberlangsungan."
        ]
    },
    "Ragi Hotang": {

```

```

    "Desain": "Memiliki pola yang rumit dan berwarna gelap, sering dihiasi motif
tradisional Batak.",
    "Kegunaan": [
        "Selimut: Digunakan untuk memberikan kehangatan.",
        "Simbol Status: Melambangkan status sosial dalam acara tertentu."
    ]
},
"Sadum": {
    "Desain": "Memiliki bingkai bergaris gelap di sisi dengan warna ceria di
tengahnya.",
    "Kegunaan": [
        "Acara Bahagia: Digunakan dalam perayaan sukacita.",
        "Kenang-Kenangan: Sering dijadikan hadiah untuk orang terkasih."
    ]
},
"Sibolang": {
    "Desain": "Berwarna dominan hitam dan putih dengan pola bergaris sederhana.",
    "Kegunaan": [
        "Acara Duka Cita: Dipakai dalam upacara pemakaman untuk menghormati
yang meninggal.",
        "Simbol Kesedihan: Melambangkan duka cita."
    ]
},
"Tumtuman": {
    "Desain": "Memiliki pola geometris unik, melambangkan harapan untuk masa
depan cerah.",
    "Kegunaan": [
        "Acara Tradisional: Digunakan untuk menunjukkan posisi dalam keluarga.",
        "Ikatan Keluarga: Dipakai oleh anak pertama dalam keluarga sebagai simbol
tanggung jawab."
    ]
}

```

```

    }
}

ulos_info = ulos_descriptions.get(predicted_class, {})
st.write(f"**Tentang {predicted_class}**")
st.write(f"**Desain:** {ulos_info.get('Desain', 'Deskripsi desain belum tersedia.')}")
st.write(f"**Kegunaan:**")
for kegunaan in ulos_info.get('Kegunaan', []):
    st.write(f"- {kegunaan}")

else:
    st.error("Model output dimensions do not match the number of class names. Please
check the model and class labels.")

except Exception as e:
    st.error(f"An error occurred: {e}")

elif page == "Panduan Pengguna":
    st.header("Panduan Pengguna")
    st.markdown(
        """
        ### Cara Menggunakan Aplikasi

        1. **Unggah Gambar:** Klik tombol "Browse Files" untuk mengunggah gambar ulos
dalam format JPG, JPEG, atau PNG.<br>

        2. **Validasi Gambar:** Pastikan gambar yang diunggah memiliki kualitas baik dan
menampilkan kain ulos dengan jelas.<br>

        3. **Hasil Prediksi:** Tunggu beberapa saat untuk melihat hasil klasifikasi dan tingkat
kepercayaan model.<br>

        4. **Informasi Tambahan:** Bacalah deskripsi singkat tentang jenis ulos yang terdeteksi
untuk memperkaya pengetahuan.

        """,

```

```

unsafe_allow_html=True
)

st.write("\n\n")
st.write("Developed by Kelompok 3 - Data Mining")

```

3.2 Timeline

Tabel 3-33 *Timeline*

Aktivitas	Sub Aktivitas	Detail	Waktu (Minggu Ke-)				
			12	13	14	15	16
Persiapan	Pemilihan Kasus dan Algoritma	Pemilihan Kasus					
		Penentuan Algoritma					
Pelaksanaan	Business Understanding	Menentukan Objektif Bisnis					
		Menentukan Tujuan Bisnis					
		Membuat Rencana Proyek					
		Data Understanding	Mengumpulkan Data				
		Menelaah Data					
		Memvalidasi Data					
	Data Preparation	Memilah Data					
		Membersihkan Data					
		Mengkonstruksi Data					
		Menentukan Label Data					
		Mengintegrasikan Data					
	Modeling	Membangun Skenario Pengujian					
		Membangun Model					
	Model Evaluation	Mengevaluasi Hasil Pemodelan					
		Melakukan Review Proses Pemodel					
		Deployment	Melakukan Deployment Model				
	Membuat Laporan Akhir Proyek						

BAB 4

HASIL PENGUJIAN

4.1 Hasil Pelatihan Model CNN

Setelah tahap pelatihan model menggunakan dataset gambar ulos, algoritma CNN berhasil dilatih untuk mengklasifikasikan gambar ulos berdasarkan kategori dan motifnya. Model dilatih menggunakan data pelatihan yang telah diproses sebelumnya, dan diuji menggunakan data uji yang tidak digunakan selama pelatihan untuk mengevaluasi kinerja model.

Pada akhir pelatihan, hasil evaluasi model menunjukkan bahwa model CNN yang dibangun memiliki kinerja yang sangat baik. Berikut adalah hasil evaluasi yang telah dilakukan.

a. Evaluasi Performa Model

Berikut adalah hasil evaluasi performa model CNN yang telah dilakukan:

- *Accuracy*
Model menghasilkan akurasi sebesar 93.27%, yang berarti ini menunjukkan akurasi yang sangat tinggi dan sebagian besar prediksi model adalah benar.
- *Precision*
Nilai *precision* yang dihasilkan adalah 93.92%, yang menunjukkan bahwa hampir semua prediksi positif yang dihasilkan oleh model adalah benar (*low false positives*).
- *Recall*:
Model menghasilkan nilai *recall* sebesar 93.31%, yang menunjukkan bahwa model dapat melakukan deteksi pada sebagian besar data positif dengan sangat baik. Ini juga menunjukkan bahwa model yang digunakan tidak melewatkan banyak contoh positif (*low false negatives*).
- *F1-Score*:
Dengan nilai *F1-Score* sebesar 93.30%, menunjukkan bahwa model yang digunakan memiliki keseimbangan yang baik antara *precision* dan *recall*.
- *Test Loss* dan *Test Accuracy*
Hasil evaluasi pada data uji menunjukkan bahwa model memiliki *test loss* sebesar 0.0226 dan *test accuracy* sebesar 99.28%. Ini berarti model dapat mengklasifikasikan data uji dengan tingkat akurasi yang sangat tinggi dan kesalahan yang sangat kecil. Nilai *loss* yang rendah menunjukkan bahwa prediksi

model mendekati nilai target sebenarnya, sementara akurasi hampir sempurna mengindikasikan kemampuan model yang baik dalam mengenali pola visual pada data uji. Kesimpulannya, model memiliki performa yang sangat baik dan generalisasi yang kuat, dengan hanya sedikit kesalahan pada beberapa kelas tertentu.

a. Evaluasi Proses

Meskipun model CNN yang digunakan sudah sangat baik, masih terdapat beberapa hal pada proses yang bisa diperbaiki, seperti:

- Penanganan *overfitting* yang dapat dilihat dari perbedaan besar antara akurasi pelatihan dan akurasi pengujian. Model terlalu menyesuaikan dengan data latih dibandingkan dengan data uji.
- Model hanya dievaluasi pada data uji, dan tidak diuji pada set data yang lain, sehingga memungkinkan model tidak tergeneralisasi dengan baik.

BAB 5

ANALISIS

5.1 Analisis Hasil Evaluasi Model

Setelah melakukan pelatihan dan evaluasi model Convolutional Neural Network (CNN) untuk klasifikasi gambar ulos, hasil evaluasi yang diperoleh sangat memuaskan.

5.1.1 Akurasi Model

Akurasi adalah metrik utama yang digunakan untuk mengukur seberapa sering model memberikan prediksi yang benar. Dalam penelitian ini, akurasi yang sangat tinggi (99.6%) menunjukkan bahwa model dapat mengenali dan mengklasifikasikan gambar ulos dengan tingkat kesalahan yang sangat rendah. Hal ini mencerminkan kualitas dataset yang baik, serta kemampuan CNN dalam mengekstraksi fitur visual yang relevan dari gambar ulos.

Namun, meskipun akurasi model sangat tinggi, penting untuk memeriksa keterbatasan dari metrik ini. Misalnya, akurasinya bisa sangat tinggi pada data yang tersebar merata antar kelas, tetapi bisa kurang efektif pada data yang memiliki class imbalance. Dalam hal ini, distribusi kategori ulos cukup seimbang, sehingga akurasi menjadi indikator yang baik untuk kinerja model.

5.1.2 Precision, Recall, dan F1-Score

Precision sebesar 93.92% menunjukkan bahwa hampir semua prediksi positif yang dilakukan oleh model adalah benar. Ini berarti model dapat membedakan dengan sangat baik antara gambar yang termasuk dalam kategori ulos tertentu dengan gambar yang tidak.

Recall sebesar 93.31% menunjukkan bahwa model dapat mendeteksi sebagian besar gambar positif (gambar dengan kategori ulos yang benar) dengan baik. *Recall* yang cukup tinggi menunjukkan bahwa model tidak banyak melewatkan contoh yang seharusnya dikenali, yang penting dalam konteks klasifikasi gambar ulos yang dapat memiliki berbagai variasi motif.

F1-Score, yang mencapai 93.30%, adalah metrik yang menggabungkan precision dan recall, memberikan gambaran tentang keseimbangan antara keduanya. Nilai *F1-Score* yang hampir setara dengan precision dan recall menunjukkan bahwa model memiliki performa yang seimbang antara menghindari *false positives* dan *false negatives*.

5.2 Kelebihan Model CNN

Model CNN yang dibangun dalam penelitian ini memiliki beberapa kelebihan yang membuatnya sangat efektif untuk tugas klasifikasi gambar ulos:

5.2.1 Kemampuan untuk Mendeteksi Fitur Visual

CNN dikenal karena kemampuannya dalam mendeteksi fitur visual yang relevan dalam gambar tanpa memerlukan fitur manual. Model CNN ini berhasil mengenali pola-pola visual yang ada pada gambar ulos, seperti motif tenun, warna, dan tekstur, yang merupakan fitur utama dalam mengklasifikasikan jenis ulos.

5.2.3 Fleksibilitas dan Skalabilitas

CNN bersifat fleksibel dan dapat diadaptasi untuk berbagai jenis dataset gambar dengan jumlah kategori yang berbeda. Model ini dapat digunakan untuk berbagai aplikasi klasifikasi gambar lainnya dengan sedikit penyesuaian, menjadikannya scalable dan mudah untuk diperluas.

5.2.4 Kemudahan Implementasi

Model CNN ini dapat diimplementasikan dengan relatif mudah menggunakan framework deep learning seperti TensorFlow dan Keras, yang menyediakan banyak fungsi dan alat untuk pelatihan dan evaluasi model. Ini memungkinkan implementasi dan deployment model secara efisien.

5.3 Keterbatasan Model CNN

Meskipun model CNN yang dibangun berhasil mencapai hasil yang sangat baik, terdapat beberapa keterbatasan yang perlu dipertimbangkan:

5.3.1 Kualitas Gambar

Kualitas gambar dalam dataset ini bervariasi. Beberapa gambar memiliki resolusi rendah atau terdistorsi, yang dapat memengaruhi hasil prediksi model. Meskipun model telah diajarkan untuk mengenali berbagai variasi dalam gambar, kualitas gambar yang buruk tetap dapat memengaruhi kinerja model.

5.3.2 Variasi dalam Motif

Motif dalam setiap kategori ulos dapat memiliki kemiripan visual yang sangat dekat, yang dapat menyebabkan kesulitan bagi model dalam membedakan antara motif yang mirip. Dalam hal ini, *augmentasi data dan fine-tuning* model dengan dataset yang lebih besar dan lebih bervariasi dapat membantu mengatasi masalah ini.

5.3.3 Waktu Latih dan Sumber Daya

Pelatihan model CNN yang menggunakan gambar berukuran besar dan dataset yang beragam membutuhkan waktu yang cukup lama dan sumber daya komputasi yang besar. Oleh karena itu, penggunaan server atau infrastruktur *cloud computing* yang lebih kuat dapat membantu mempercepat proses pelatihan dan pengujian.

5.3.4 Overfitting

Meskipun kami menggunakan teknik early stopping dan dropout untuk mencegah *overfitting*, kemungkinan overfitting tetap ada, terutama jika model terlalu lama dilatih pada data yang terbatas. Penggunaan *data augmentation* dan *cross-validation* dapat membantu mengurangi risiko *overfitting*.

5.4 Saran untuk Penelitian Selanjutnya

Berdasarkan hasil yang telah diperoleh, terdapat beberapa langkah yang bisa diambil untuk meningkatkan kualitas dan kinerja model CNN:

1. Peningkatan Kualitas Data:

Dataset harus diperluas dengan gambar ulos yang memiliki kualitas lebih tinggi untuk mengurangi pengaruh gambar yang buram atau terdistorsi.

2. *Data Augmentation* Lebih Lanjut:

Augmentasi data seperti rotasi, *flipping*, dan *zooming* gambar dapat diterapkan lebih lanjut untuk memperkaya dataset pelatihan, membantu model mengatasi variasi gambar yang lebih luas.

3. Penggunaan *Transfer Learning*:

Menggunakan model yang sudah dilatih sebelumnya (*pre-trained models*) seperti ResNet atau VGG16 dapat meningkatkan akurasi dan mempercepat waktu pelatihan.

BAB 6

KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi gambar ulos menggunakan *Convolutional Neural Network* (CNN), yang dapat mengenali dan mengklasifikasikan berbagai jenis kain ulos Batak berdasarkan gambar. Dengan menggunakan dataset yang berisi gambar-gambar ulos dengan berbagai motif dan kategori, algoritma CNN yang dikembangkan menunjukkan hasil yang sangat baik dalam mengklasifikasikan gambar-gambar tersebut. CNN yang dibangun berhasil mencapai *precision*, *recall*, dan *F1-score* yang sangat baik, masing-masing sebesar 93.92%, 93.31%, dan 93.30%, serta *test loss* sebesar 0.0226 dan *test accuracy* sebesar 99.28% menunjukkan keseimbangan yang sangat baik antara akurasi dan kemampuan mendeteksi gambar ulos.

DAFTAR REFERENSI

- [1] A. C. Barus, M. Simanjuntak, and V. Situmorang, “DiTenun, Smart Application Producing Ulos Motif,” SHS Web Conf., vol. 86, p. 01025, 2020, doi: 10.1051/shsconf/20208601025.
- [2] H. Simanjuntak, E. Panjaitan, S. Siregar, U. Manalu, S. Situmeang, and A. Barus, “Generating New Ulos Motif with Generative AI Method in Digital Tenun Nusantara (DiTenun) Platform,” Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 7, pp. 1125–1134, 2024, doi: 10.14569/IJACSA.2024.01507109.
- [3] E. Theresia Panjaitan, S. Siregar, U. Ignasius Manalu, and H. Tommy Argo Simanjuntak, “Generating New Ulos Motif With StyleGAN Method in Digital Tenun Nusantara (DiTenun) Applications,” IAES International Journal of Artificial Intelligence (IJ-AI, vol. 99, no. 1. pp. 1–1, 2019.

BAB 7

PEMBAGIAN TUGAS

1. Elshaday Prida Simamora-12S21047

a. Pembuatan Bab 1

- Menulis latar belakang masalah, tujuan penelitian, rumusan masalah, serta manfaat dari sistem yang dikembangkan.
- Menyusun ruang lingkup dan tujuan dari penelitian.

b. Membuat Landasan Teori

- Mencari dan menulis literatur mengenai teori dasar yang digunakan dalam model CNN untuk klasifikasi gambar.
- Menganalisis teori-teori yang relevan terkait dengan identifikasi gambar ulos dan teknologi terkait.

c. Membuat Model

- Menyusun penjelasan tentang pemilihan model Convolutional Neural Network (CNN) yang digunakan dalam klasifikasi gambar ulos.
- Menyusun struktur model, algoritma, dan teknik pelatihan yang dipilih untuk model CNN.

d. Hasil dan Pembahasan

- Menyusun hasil evaluasi model, interpretasi akurasi, precision, recall, F1-score, serta analisis kesalahan model.
- Menyajikan grafik atau visualisasi hasil prediksi dan diskusi terkait kesimpulan yang dapat diambil dari hasil evaluasi.

2. Nussy Pentasonia Pangaribuan-12S21048

a. Membuat Business Understanding

- Menyusun penjelasan tentang masalah bisnis atau tujuan aplikasi yang akan dikembangkan.
- Menyusun hubungan antara solusi yang dikembangkan dengan kebutuhan pasar atau pengguna.

b. Membuat Data Understanding

- Menyusun pemahaman terkait dataset yang digunakan dalam penelitian, termasuk sumber data, jenis data, dan tujuan penggunaan data tersebut.
- Melakukan analisis awal terhadap dataset untuk memahami distribusi data, kelas ulos yang ada, dan karakteristiknya.

c. Membuat Deploy

- Menjelaskan implementasi model dalam aplikasi berbasis web menggunakan framework seperti Streamlit atau Flask.
- Menyusun alur kerja deployment, mulai dari pengunggahan gambar hingga pemberian prediksi.

d. Hasil dan Pembahasan

- Menyusun hasil eksperimen yang dilakukan pada model, serta menyajikan analisis yang lebih mendalam terkait pengaruh parameter yang digunakan.
- Menyajikan hasil perbandingan antara model yang dikembangkan dengan model lain jika relevan.

3. Jesika Audina Purba-12S21049

a. Membuat Data Processing

- Menjelaskan tahapan preprocessing data, seperti normalisasi gambar, pengubahan ukuran gambar, dan pembagian dataset (train/test).
- Menyusun teknik augmentasi data jika digunakan untuk meningkatkan performa model.

b. Membuat Model

- Menyusun langkah-langkah detail dalam pelatihan model CNN, termasuk pemilihan parameter dan hyperparameter.
- Menulis tentang teknik optimasi dan evaluasi model yang digunakan.

c. Kesimpulan

- Menyusun kesimpulan berdasarkan hasil penelitian dan aplikasi yang telah dikembangkan.
- Memberikan saran untuk penelitian atau pengembangan sistem lebih lanjut.