

Klasifikasi Kunyit dan Temulawak dengan VGG16 dan Fuzzy Tsukamoto Berbasis Android

Muhammad Rizki Setyawan, Fajar Rahardika Bahari Putra*, Ahmad Ilham, Dimas Adi Suseno

¹ Fakultas Teknik, Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong, Sorong, Indonesia
Email: ¹Rizki@um-sorong.ac.id, ^{2*}fajar_rbp@um-sorong.ac.id, ³ahmadilham0517@gmail.com, ⁴dimasadisuseno02@gmail.com

Email Penulis Korespondensi: fajar_rbp@um-sorong.ac.id

Submitted 02-06-2025; Accepted 30-06-2025; Published 30-06-2025

Abstrak

Indonesia memiliki keanekaragaman hayati yang sangat kaya, termasuk berbagai tanaman obat yang sangat menguntungkan secara finansial dan bermanfaat bagi kesehatan. Di antara tanaman obat ini, temulawak dan kunyit adalah dua rimpang yang paling populer dan banyak digunakan dalam pengobatan tradisional maupun industri herbal. Namun, karena bentuk dan warna kedua tanaman ini sangat mirip, seringkali sulit untuk membedakannya, terutama bagi orang awam dan pekerja industri baru. Penelitian ini mengembangkan sebuah aplikasi berbasis Android yang dapat membedakan temulawak dan kunyit secara efektif dan akurat untuk mengatasi masalah ini. Untuk aplikasi ini, arsitektur *Convolutional Neural Network* (CNN) model VGG-16 digunakan bersama dengan metode *fuzzy Tsukamoto* sebagai lapisan tambahan. Uji coba yang dilakukan pada model yang dikembangkan menggunakan data uji menunjukkan tingkat ketepatan sebesar 0.97, nilai recall sebesar 0.98, dan skor F1 sebesar 0.97. Sedangkan untuk pengujian *blackbox* menunjukkan bahwa aplikasi ini berfungsi dengan stabil tanpa kendala teknis, sehingga siap untuk digunakan. Selain itu, pengujian *blackbox* menunjukkan bahwa sistem dapat berfungsi dengan stabil tanpa kendala, sehingga layak untuk digunakan dalam penggunaan nyata.

Kata Kunci: Kunyit; Temulawak; VGG-16; Fuzzy Tsukamoto; Android

Abstract

Indonesia has a very rich biodiversity, including various medicinal plants that are highly financially beneficial and health-promoting. Among these medicinal plants, temulawak and turmeric are the two most popular rhizomes widely used in traditional medicine as well as the herbal industry. However, because the shape and color of these two plants are very similar, it is often difficult to distinguish between them, especially for laypeople and new industry workers. This research developed an Android-based application that can effectively and accurately distinguish between temulawak and turmeric to address this issue. For this application, the Convolutional Neural Network (CNN) architecture of the VGG-16 model is used along with the Tsukamoto fuzzy method as an additional layer. The trials conducted on the developed model using test data showed an accuracy rate of 0.97, a recall value of 0.98, and an F1 score of 0.97. Meanwhile, the blackbox testing shows that this application functions stably without technical issues, making it ready for use. Additionally, blackbox testing shows that the system can function stably without any issues, making it suitable for real-world use.

Keywords: Turmeric; Temulawak; VGG-16; Fuzzy Tsukamoto; Android

1. PENDAHULUAN

Indonesia dikenal sebagai salah satu negara dengan tingkat keanekaragaman hayati tertinggi di dunia, termasuk dalam hal kekayaan tanaman obat tradisional [1], [2]. Dua di antara tanaman herbal yang banyak dimanfaatkan adalah temulawak (*Curcuma xanthorrhiza*) dan kunyit (*Curcuma longa*) [3], [4]. Kedua tanaman ini memiliki kemiripan bentuk fisik, khususnya pada bagian rimpangnya, sehingga sering menimbulkan kesulitan dalam proses identifikasi secara visual, terutama bagi masyarakat awam maupun petani pemula. Kesalahan dalam mengklasifikasikan kedua jenis tanaman tersebut dapat berdampak pada menurunnya kualitas produk, ketidaktepatan dalam pengolahan, serta berkurangnya efektivitas penggunaan tanaman untuk pengobatan atau konsumsi [5], [6]. Perkembangan teknologi di bidang kecerdasan buatan (*Artificial Intelligence*), khususnya *Convolutional Neural Network* (CNN), telah menghadirkan solusi potensial dalam proses klasifikasi citra tanaman secara otomatis [7], [8]. CNN memiliki keunggulan dalam mengenali fitur-fitur visual kompleks seperti tekstur, warna, dan bentuk, yang sangat relevan untuk membedakan objek-objek dengan kemiripan tinggi [9], [10]. Meskipun demikian, hasil klasifikasi dari CNN sering kali bersifat deterministik dan kurang fleksibel dalam menghadapi ketidakpastian data, seperti kondisi pencahayaan yang buruk, sudut pengambilan gambar yang beragam, atau kualitas citra yang rendah [11]. Dalam kondisi semacam itu, dibutuhkan metode yang mampu mengakomodasi ketidakpastian dan variabilitas data input. Salah satu pendekatan yang relevan adalah penerapan logika *fuzzy*, khususnya *Fuzzy Tsukamoto* [12]. Metode ini mampu menangani input numerik secara halus dan menghasilkan keluaran yang lebih bersifat gradual, berdasarkan aturan-aturan yang merepresentasikan pengetahuan manusia [13]. Dalam konteks klasifikasi temulawak dan kunyit, algoritma ini dapat memperkuat keputusan klasifikasi dengan menilai derajat keanggotaan suatu fitur terhadap kelas tertentu, sehingga sistem menjadi lebih adaptif terhadap ambiguitas yang ada dalam citra dan juga terdapat beberapa kekurangan pemahaman dalam membedakan antar temulawak dan kunyit di mata masyarakat kota sorong papua barat daya.

Beberapa penelitian terdahulu telah mengimplementasikan *Convolutional Neural Network* (CNN) dan *Fuzzy Tsukamoto* dalam berbagai bidang. Penelitian yang dilakukan oleh [14] membahas tentang klasifikasi kualitas daging marmer menggunakan *Convolutional Neural Network* (CNN) dengan dataset 300 gambar. Model dilatih selama 50 *epoch*, dengan akurasi pelatihan 96,67% dan akurasi pengujian 88,15%, Grafik *loss* dan *accuracy* menunjukkan model mengalami konvergensi yang baik, dengan peningkatan akurasi yang stabil. Model ini diterapkan dalam aplikasi berbasis

Android untuk klasifikasi otomatis, membuktikan bahwa pendekatan CNN efektif untuk identifikasi kualitas daging marmer, meskipun perlu evaluasi lebih lanjut untuk meningkatkan klasifikasi pada kategori kualitas rendah. Penelitian selanjutnya [15] membahas klasifikasi kualitas beras dengan menggunakan *Convolutional Neural Network* (CNN) dengan model VGG16 dan pendekatan pembelajaran transfer pada dataset yang terdiri dari 340 gambar beras berkualitas baik dan buruk. Data dibagi menjadi sepuluh persen validasi, dua puluh persen pengujian, dan tujuh puluh persen pelatihan. Model dilatih selama sepuluh epoch menggunakan bobot *pre-trained ImageNet*. Untuk aplikasi Android, itu disimpan dalam format TFLite. Hasilnya menunjukkan tingkat akurasi sebesar 97%, dengan nilai *precision*, *recall*, dan *F1-score* yang tinggi pada kedua kelas. Studi [16] menggunakan metode *Fuzzy Tsukamoto* dalam menentukan jumlah produksi roti didasarkan pada nilai permintaan dan stok, yang diklasifikasikan ke dalam himpunan fuzzy seperti “Sedikit” dan “Banyak”. Prosesnya mencakup *fuzzifikasi*, inferensi dengan aturan *IF-THEN*, dan *defuzzifikasi* menggunakan metode centroid. Hasilnya menunjukkan bahwa metode ini memiliki tingkat akurasi tinggi dengan error rendah, menghasilkan prediksi produksi yang mendekati nilai aktual. Pendekatan *fuzzy* terbukti mengakomodasi ketidakpastian dan fluktuasi stok, memberikan prediksi adaptif dan fleksibel terhadap perubahan pasar, sehingga efektif sebagai sistem pendukung keputusan untuk menghindari overproduksi maupun kekurangan produksi.

Berdasarkan hasil peneliti terdahulu, meskipun *Convolutional Neural Network* (CNN) telah banyak digunakan untuk klasifikasi citra dengan akurasi tinggi, metode ini cenderung bersifat deterministik dan kurang fleksibel dalam menghadapi ketidakpastian data visual seperti variasi pencahayaan, sudut pengambilan gambar, dan kualitas citra rendah. Penelitian terdahulu oleh [16] dan [17] menunjukkan efektivitas CNN dalam klasifikasi objek seperti daging marmer dan beras, namun tidak mempertimbangkan ketidakpastian atau ambiguitas visual, serta tidak menggabungkan pendekatan fuzzy untuk memperkuat proses pengambilan keputusan. Di sisi lain, penelitian [18] membuktikan bahwa metode *Fuzzy Tsukamoto* efektif dalam menangani data numerik yang mengandung ketidakpastian, tetapi belum diterapkan pada domain visual atau dalam kombinasi dengan CNN. Hingga saat ini, belum terdapat penelitian yang secara khusus menggabungkan CNN dan *Fuzzy Tsukamoto* untuk klasifikasi citra tanaman herbal yang memiliki kemiripan visual tinggi seperti temulawak dan kunyit, terutama dalam konteks lokal masyarakat Papua Barat Daya yang masih mengalami kebingungan dalam membedakan kedua jenis tanaman ini. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi berbasis CNN-Fuzzy *Tsukamoto* yang lebih adaptif dan akurat dalam mengatasi ambiguitas visual, serta relevan dalam membantu identifikasi temulawak dan kunyit di lapangan.

Dengan demikian aspek krusial lain dalam pengembangan sistem klasifikasi ini adalah kemudahan akses teknologi bagi masyarakat. Pemilihan platform Android sebagai basis sistem dinilai strategis, mengingat dominansi sistem operasi ini di pasar Indonesia. Solusi berbasis Android memungkinkan pemanfaatan kamera smartphone sebagai perangkat akuisisi citra langsung di lapangan, sehingga proses klasifikasi dapat dilakukan secara *real-time*. Pendekatan ini tidak hanya mempercepat identifikasi tetapi juga meningkatkan efisiensi praktis, khususnya di lokasi budidaya atau fasilitas pengolahan tanaman [17], [18]. Berdasarkan masalah di atas, penulis ingin merancang dan mengembangkan sistem klasifikasi kunyit dan temulawak berbasis citra digital dengan menggabungkan arsitektur *Convolutional Neural Network* (CNN) dengan model VGG16 dan metode *Fuzzy Tsukamoto* yang di implementasikan dalam aplikasi berbasis Android. Diharapkan sistem ini dapat melakukan identifikasi tanaman secara otomatis, cepat, dan akurat sekaligus menyediakan solusi teknologi yang bermanfaat untuk mendorong masyarakat pertanian lokal.

2. METODOLOGI PENELITIAN

2.1 Kajian Pustaka

2.1.1. Metode Fuzzy Tsukamoto

Metode *Tsukamoto* merupakan salah satu bagian dari *Fuzzy Inference System* (FIS), di mana setiap aturan yang berbentuk *if-then* harus memiliki konsekuensi yang direpresentasikan oleh himpunan fuzzy dengan fungsi keanggotaan monoton. FIS sendiri merupakan kerangka kerja komputasional yang menggabungkan sejumlah aturan fuzzy berdasarkan input yang tersedia, dengan landasan teori himpunan fuzzy, aturan *if-then*, dan penalaran berbasis logika fuzzy. Proses dalam sistem ini diawali dengan *fuzzifikasi*, yaitu mengubah variabel crisp menjadi variabel fuzzy berdasarkan derajat keanggotaannya dalam suatu himpunan. Setelah dilakukan proses inferensi untuk menentukan nilai output fuzzy dari aturan-aturan yang ada, dilakukan tahap akhir berupa *defuzzifikasi*, yaitu mengubah kembali hasil fuzzy menjadi nilai crisp agar dapat digunakan dalam pengambilan keputusan [19].

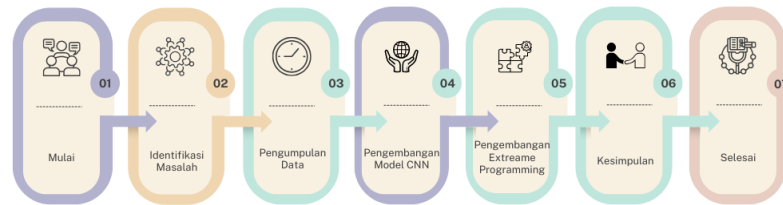
2.1.2. VGG16

VGG16 adalah salah satu arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh Visual Geometry Group (VGG) dari University of Oxford. Model ini memiliki total 16 lapisan dengan bobot yang dapat dilatih, terdiri dari 13 lapisan konvolusi dan 3 lapisan *fully connected*. VGG16 dirancang untuk pengenalan citra berskala besar dan dikenal karena arsitekturnya yang konsisten menggunakan filter 3x3 serta struktur berulang, yang memudahkan implementasi dan adaptasi untuk berbagai tugas klasifikasi visual [20], [21].

2.2 Metodologi Penelitian

2.2.1. Tahapan Penelitian

Dalam penelitian ini, penulis mengusulkan pengembangan aplikasi yang memanfaatkan arsitektur Convolutional Neural Network (CNN) model VGG-16 yang dikombinasikan dengan metode Fuzzy Tsukamoto untuk klasifikasi tanaman kunyit dan temulawak. Pengembangan dilakukan secara bertahap, dimulai dengan identifikasi masalah dan berakhir pada tahap pengujian aplikasi yang dapat dilihat pada Gambar 1 [22].



Gambar 1. Alur Penelitian

Gambar 1 menunjukkan alur tahapan dalam sebuah proyek penelitian atau pengembangan sistem berbasis CNN. Proses dimulai dari tahap Mulai, dilanjutkan dengan Identifikasi Masalah untuk memahami isu yang ingin diselesaikan. Setelah itu, dilakukan Pengumpulan Data yang relevan sebagai dasar pelatihan model. Tahap berikutnya adalah Pengembangan Model CNN, di mana dilakukan pelatihan dan pengujian model deep learning. Model ini kemudian diterapkan dalam sistem menggunakan pendekatan *Extreme Programming* yang menekankan iterasi cepat dan kolaborasi. Selanjutnya, dibuat Kesimpulan dari hasil pengujian dan evaluasi, lalu diakhiri dengan tahap Selesai sebagai penutup dari keseluruhan proses.

2.2.2. Identifikasi Masalah

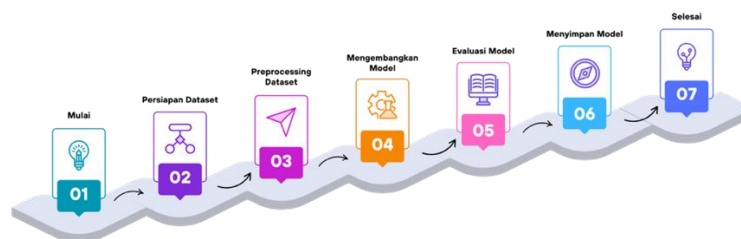
Langkah pertama adalah mengidentifikasi masalah utama, yaitu klasifikasi temulawak dan kunyit. Klasifikasi ini penting untuk membedakan keduanya, karena tugas ini cukup kompleks. Identifikasi masalah ini menjadi dasar pengembangan aplikasi berbasis Android dengan model *Deep Learning* dan algoritma *Fuzzy Tsukamoto* untuk membedakan temulawak dan kunyit.

2.2.3. Pengumpulan Data

Setelah masalah diidentifikasi, peneliti mencari informasi tentang metode klasifikasi yang akan digunakan dan mengumpulkan data. Proses ini mencakup pencarian literatur terkait penelitian sebelumnya, identifikasi sumber data, serta pengumpulan dataset gambar temulawak dan kunyit untuk pelatihan model.

2.2.4. Pengembangan Model CNN

Pada tahap ini, peneliti mengembangkan arsitektur CNN menggunakan mode VGG16 yang dilengkapi dengan layer Fuzzy Tsukamoto. Model ini dikembangkan menggunakan Kaggle notebook dan bahasa pemrograman Python. Proses pengembangan model CNN dapat dilihat pada Gambar 2 [23].



Gambar 2. Pengembangan Model CNN

Adapun penjelasan dari pengembangan CNN pada Gambar 2 sebagai berikut:

a. Persiapan Dataset

Pada tahap ini, dilakukan pengumpulan dataset yang terdiri dari gambar kunyit dan temulawak. Dataset tersebut kemudian diorganisasi menjadi tiga subset utama: *training*, *validation*, dan *testing*, yang masing-masing digunakan untuk melatih, memvalidasi, dan menguji model.

b. Praproses Dataset

Tahapan praproses dilakukan untuk meningkatkan kualitas dan keragaman data pelatihan melalui teknik augmentasi citra. Pendekatan ini bertujuan untuk meningkatkan kemampuan generalisasi model terhadap variasi data dunia nyata.

c. Mengembangkan Model

Pada tahap ini, arsitektur *Convolutional Neural Network* (CNN) dengan model VGG16 dikombinasikan dengan metode *Fuzzy Tsukamoto* untuk meningkatkan performa klasifikasi, terutama pada data yang mengandung ketidakpastian visual.

d. Evaluasi Model

Pada tahap ini, model dievaluasi dengan menggunakan data tes untuk menghasilkan confusion matrix dan menghitung metrik akurasi. Tujuan dari evaluasi ini adalah untuk mengetahui seberapa baik model dapat membedakan antara kelas temulawak dan kunyit.

e. Menyimpan Model

Pada tahap ini, peneliti akan menyimpan model dalam format .tflite untuk digunakan dalam Aplikasi Android.

2.2.5. Pengembangan Extreme Programming

Pada tahap ini, penulis mengembangkan aplikasi Android saat ini menggunakan Extreme Programming (XP), tahapan yang digambarkan pada Gambar 3 [24].



Gambar 3. Pengembangan *Extreme Programming*

Dengan menggunakan metode Extreme Programming, setiap tahapan pengembangan aplikasi dijelaskan sebagai berikut:

a. *Planning*

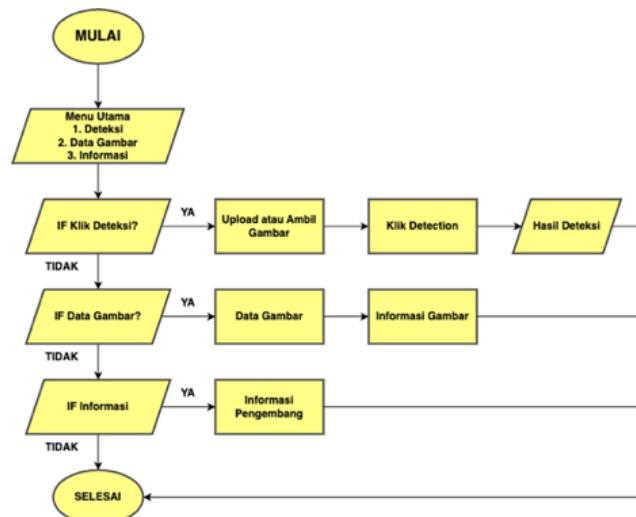
Tahap ini menganalisis kebutuhan sistem, yang mencakup kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional termasuk kemampuan pengguna untuk mengakses aplikasi, menampilkan hasil klasifikasi, dan mendapatkan informasi tentang tingkat akurasi. Tabel 1 menunjukkan kebutuhan non-fungsional saat ini.

Tabel 1. Kebutuhan Non-fungsional

No	Perangkat Lunak	Perangkat Keras
1.	Windows 11 64 bit	Laptop ASUS X550Z
2.	Kaggle sebagai Workshop pelatihan CNN	Smartphone Redmi Note 10S
3.	Google Chrome sebagai Browser	
4.	Android Studio sebagai Developer	

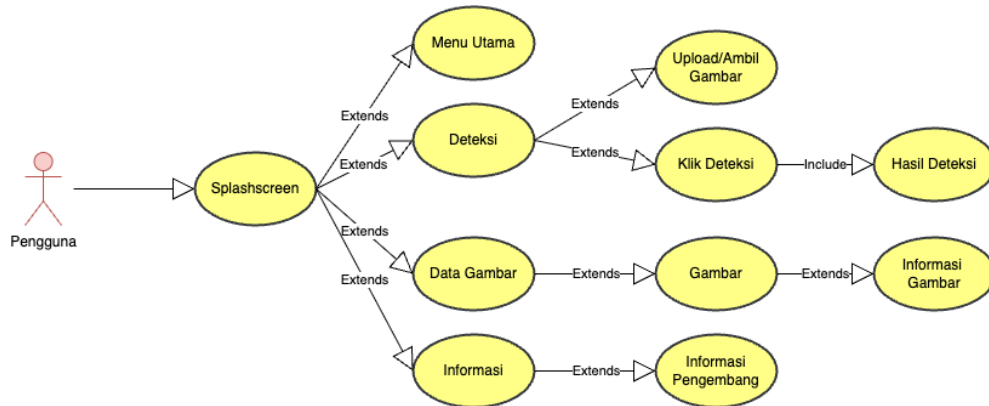
b. *Design*

Tahap ini mencakup pembuatan *flowchart* dan *use case* untuk menggambarkan alur kerja serta fungsi utama sistem. Desain *flowchart* dari aplikasi yang akan dibuat dapat dilihat pada Gambar 4.



Gambar 4. *Flowchart*

Sedangkan untuk desain *use case* dari aplikasi yang akan dibuat dapat dilihat pada gambar 5.



Gambar 5. Use case

a. Coding

Pada tahap ini, aplikasi Android dikembangkan menggunakan Android Studio untuk merealisasikan desain yang telah dibuat.

b. Testing

Pada tahap selanjutnya, pengujian sistem dilakukan untuk memastikan bahwa aplikasi beroperasi sesuai dengan spesifikasi yang telah dirancang. Metode pengujian *blackbox* digunakan untuk mengevaluasi kesesuaian output terhadap input. Metode ini tidak mempertimbangkan struktur internal program.

2.2.6. Kesimpulan

Bagian kesimpulan berisi ringkasan dari hasil yang telah diperoleh dalam penelitian atau pengembangan sistem.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Model CNN

a. Persiapan Dataset

Pada tahap ini, data dikumpulkan secara manual dengan memotret gambar kunyit dan temulawak. Contoh dataset yang telah berhasil dikumpulkan ditampilkan pada Gambar 7.



Gambar 7. Contoh Dataset

Gambar 7 menampilkan contoh dataset yang terdiri atas 800 citra, masing-masing sebanyak 400 citra untuk kelas kunyit dan temulawak. Dataset ini kemudian dibagi ke dalam tiga subset, yaitu data *training*, *validation*, dan *testing* dengan rasio 70:20:10. Hasil pembagian dataset tersebut ditampilkan pada Gambar 8.

```

# === 2. SPLIT DATASET ===

train_split=.7
test_split=.1
dummy_split=test_split/(1-train_split)
train_df, dummy_df=train_test_split(df, train_size=train_split, shuffle=True, random_state=123)
test_df, valid_df=train_test_split(dummy_df, train_size=dummy_split, shuffle=True, random_state=123)
print('train_df length: ', len(train_df), ' valid_df length: ', len(valid_df), ' test_df length: ', len(test_df))

train_df length: 560  valid_df length: 160  test_df length: 80
  
```

Gambar 8. Hasil Pembagian Dataset

Gambar 8 menunjukkan hasil pembagian dataset berdasarkan rasio yang telah ditentukan yaitu data *training* sebesar 560 gambar, data *validation* sebesar 160 gambar dan data *testing* sebesar 80 gambar.

b. Praproses Dataset

Selanjutnya dataset yang telah dikumpulkan kemudian dilakukan preprocessing, sebagaimana ditampilkan pada Gambar 9.

```
# == 3. PRE-PROCESSING DATASET ==
height = 224
width = 224
channels = 3
img_shape = (height, width, channels)
img_size = (height, width)

# Tentukan batch size yang diperbolehkan
allowed_batch_sizes = [64]

# Panjang dataset
length = len(test_df)

# Cari batch size yang bisa digunakan
valid_batch_sizes = [b for b in allowed_batch_sizes if length % b == 0]

# Pilih batch size terbesar yang masih sesuai, jika tidak ada ambil yang terkecil
batch_size = max(valid_batch_sizes) if valid_batch_sizes else min(allowed_batch_sizes)

test_batch_size = batch_size

# Hitung jumlah langkah (steps)
test_steps = length // test_batch_size

print('Test batch size:', test_batch_size, 'Test steps:', test_steps)

gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40, #
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.3, # Lebih agresif
    horizontal_flip=True,
    fill_mode='nearest' # Mengisi area kosong dengan piksel terdekat
)

train_gen = gen.flow_from_dataframe(
    train_df, x_col='filepath', y_col='labels', target_size=img_size,
    class_mode='categorical', color_mode='rgb', shuffle=True, batch_size=batch_size
)

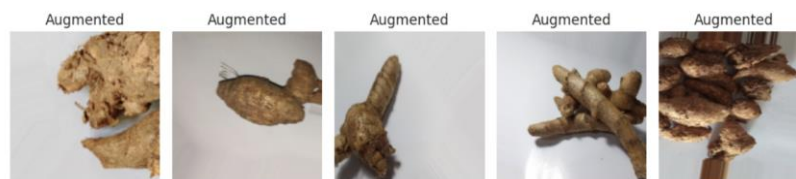
valid_gen = ImageDataGenerator(rescale=1./255)
valid_gen = valid_gen.flow_from_dataframe(
    valid_df, x_col='filepath', y_col='labels', target_size=img_size,
    class_mode='categorical', color_mode='rgb', shuffle=False, batch_size=batch_size
)

test_gen = ImageDataGenerator(rescale=1./255)
test_gen = test_gen.flow_from_dataframe(
    test_df, x_col='filepath', y_col='labels', target_size=img_size,
    class_mode='categorical', color_mode='rgb', shuffle=False, batch_size=test_batch_size
)

classes = list(train_gen.class_indices.keys())
print(classes)
class_count = len(classes)
```

Gambar 9. Preprocessing Dataset

Gambar 8 menampilkan kode praproses dataset yang diterapkan meliputi pertama, ditetapkan ukuran gambar input sebesar 224x224 piksel dengan 3 kanal warna (RGB). Kemudian ditentukan ukuran batch yang valid, yaitu 64, dengan memastikan bahwa jumlah data dalam test dapat dibagi habis oleh batch tersebut. Selanjutnya data pelatihan diterapkan augmentasi citra seperti rotasi, translasi, *zoom*, *flipping horizontal*, dan pengisian piksel kosong dengan metode *nearest neighbor* guna meningkatkan keragaman data. Data validasi dan pengujian hanya dinormalisasi tanpa augmentasi dan tidak diacak untuk menjaga konsistensi evaluasi. Ketiga generator (*training*, *validation*, dan *testing*) dihasilkan dari dataframe gambar dengan label masing-masing, menggunakan ukuran target yang telah ditentukan. Pada tahap akhir, nama-nama kelas diekstraksi dari training generator untuk mengidentifikasi jumlah serta jenis kelas yang digunakan dalam proses klasifikasi. Contoh hasil augmentasi dari dataset ditampilkan pada Gambar 10.



Gambar 10. Contoh Hasil Augmentasi Gambar

c. Mengembangkan Model CNN+Fuzzy Tsukamoto

Pada tahap ini, pengembangan model CNN dan *Fuzzy Tsukamoto* dimulai dengan mendefinisikan terlebih dahulu lapisan (*layer*) *Fuzzy Tsukamoto* yang akan digunakan dalam model. Proses pendefinisian ini bertujuan untuk mengintegrasikan logika fuzzy ke dalam jaringan, guna meningkatkan kemampuan klasifikasi model terhadap data yang memiliki ketidakpastian. Gambar proses pendefinisian *layer Fuzzy Tsukamoto* ditampilkan pada Gambar 10.

```
def fuzzy_tsukamoto(x):
    kunyit = tf.clip_by_value((x - 0.4) / (1.0 - 0.4), 0.0, 1.0)
    temulawak = tf.clip_by_value((0.6 - x) / (0.6 - 0.2), 0.0, 1.0)
    rule_kunyit = kunyit * 0.8
    rule_temulawak = temulawak * 0.2
    numerator = rule_kunyit + rule_temulawak
    denominator = kunyit + temulawak + 1e-6
    fuzzy_score = numerator / denominator
    fuzzy_score = tf.reduce_mean(fuzzy_score, axis=1, keepdims=True)
    return fuzzy_score
```

Gambar 10. Pendefinisian Fungsi *Fuzzy Tsukamoto*

Gambar 10 merupakan pendefinisian fungsi *fuzzy_tsukamoto* yang digunakan untuk mengimplementasikan *fuzzy Tsukamoto* sebagai lapisan tambahan setelah fitur diekstraksi oleh CNN. Fungsi ini menerima input berupa tensor berdimensi (*batch_size*, *feature_dim*) dan menghitung derajat keanggotaan terhadap dua kategori, yaitu "kunyit" dan "temulawak", dengan menggunakan fungsi keanggotaan linear yang dibatasi nilainya antara 0 dan 1. Aturan fuzzy diterapkan dengan memberikan bobot (*konsekuensi*) sebesar 0.8 untuk kunyit dan 0.2 untuk temulawak. Hasil dari kedua aturan tersebut kemudian dikombinasikan melalui proses *defuzzifikasi Tsukamoto*, yaitu dengan membagi jumlah tertimbang oleh total keanggotaan, dan nilai akhir *diformalisasi* sebagai skor fuzzy yang berada di antara 0 dan 1. Skor akhir kemudian dirata-rata di seluruh fitur untuk menghasilkan satu nilai fuzzy representatif per sampel. Setelah melakukan pendefinisian fungsi *fuzzy_tsukamoto*, selanjutnya adalah membangun arsitektur CNN menggunakan model

VGG-16 yang dimodifikasi dengan mengintegrasikan *layer Fuzzy Tsukamoto* yang telah didefinisikan sebelumnya. Integrasi ini bertujuan untuk memperkuat kemampuan model dalam menangani ambiguitas data visual, seperti perbedaan warna dan tekstur antara kunyit dan temulawak. Pengembangan arsitektur model VGG-16 dan *Fuzzy Tsukamoto* dapat dilihat pada Gambar 11.

```
base_models = {
    "VGG16": VGG16
}
models = {}

for name, BaseModel in base_models.items():
    base_model = BaseModel(weights='imagenet', include_top=False,
                             input_shape=img_shape)

    # Hitung 80% dari total layer untuk di-freeze
    freeze_until = int(len(base_model.layers) * 0.8)
    for layer in base_model.layers[:freeze_until]:
        layer.trainable = False
    for layer in base_model.layers[freeze_until:]:
        layer.trainable = True
    x = Flatten()(base_model.output)
    x_fuzzy = Lambda(fuzzy_tsukamoto, output_shape=lambda input_shape:
                     (input_shape[0], 1), name=f'fuzzy_tsukamoto_{name}')(x)
    x_combined = tf.keras.layers.Concatenate(axis=1)([x, x_fuzzy])
    x = Dense(512, activation='relu')(x_combined)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    x = Dense(128, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    out = Dense(2, activation='softmax')(x)
```

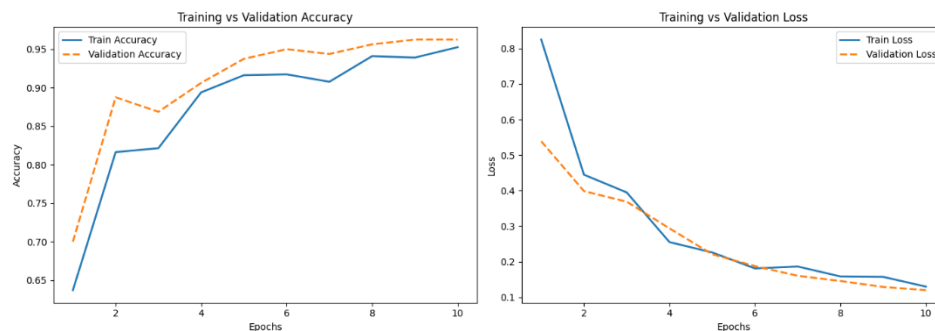
Gambar 11. Arsitektur Model VGG16 + Layer Fuzzy Tsukamoto

Gambar 11 menunjukkan arsitektur gabungan antara CNN berbasis VGG16 dan logika *fuzzy Tsukamoto* untuk meningkatkan akurasi klasifikasi citra. Model VGG16 digunakan dengan bobot *pre-trained* dari *ImageNet*, di mana 80% lapisan awal dibekukan untuk menjaga fitur dasar, sementara sisanya disesuaikan dengan data baru. Output dari VGG16 di-flatten menjadi vektor satu dimensi dan diproses melalui fungsi *fuzzy_tsukamoto*, yang menggunakan dua *membership function* dan aturan *fuzzy* untuk menghasilkan satu nilai keputusan. Nilai ini kemudian digabungkan dengan fitur CNN menggunakan layer *Concatenate*. Gabungan fitur ini diproses melalui tiga *layer Dense* berturut-turut, dilengkapi dengan *BatchNormalization* dan *Dropout* untuk stabilitas dan mencegah *overfitting*. Terakhir, *layer output Dense* dengan aktivasi *softmax* digunakan untuk klasifikasi dua kelas. Selanjutnya model dilakukan pelatihan yang dapat dilihat pada Gambar 12.

```
model = Model(inputs=base_model.input, outputs=out)
model.compile(optimizer=tf.keras.optimizers.AdamW(learning_rate=5e-5),
              loss='categorical_crossentropy', metrics=['accuracy'])
models[name] = model
```

Gambar 12. Compile dan Latih Model

Gambar 12 menunjukkan kode untuk menyusun dan mengkompilasi model dengan menggunakan *optimizer Adam* dengan *learning rate* sebesar $5e-5$, yang dipilih untuk menjaga kestabilan *fine-tuning* pada model yang telah dilatih sebelumnya (*pretrained model*) dan mencegah perubahan besar pada bobot yang telah ada. Fungsi *loss* yang digunakan adalah *categorical_crossentropy* untuk tugas klasifikasi multi-kelas antara kunyit dan temulawak. Metrik yang digunakan untuk evaluasi model adalah *accuracy* untuk memantau kinerja klasifikasi. Setelah kompilasi, model disiapkan untuk dilatih dengan tujuan memaksimalkan akurasi dan mengurangi kesalahan klasifikasi. Grafik pelatihan dan validasi akurasi dan *loss* dari model yang telah dilatih ditampilkan pada Gambar 13.



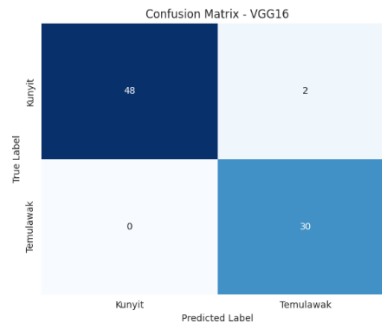
Gambar 13. Hasil Pelatihan Model

Gambar 13 menunjukkan perkembangan akurasi dan loss selama 10 epoch pelatihan. Akurasi pelatihan meningkat stabil dari sekitar 65% pada epoch pertama hingga mencapai lebih dari 95% pada epoch ke-10, sementara akurasi validasi juga menunjukkan peningkatan yang signifikan meskipun dengan sedikit fluktuasi, menandakan model mampu belajar

dengan baik namun sedikit mengalami ketidakstabilan pada data validasi. Grafik *loss* memperlihatkan penurunan yang konsisten pada kedua data pelatihan dan validasi, dengan *loss* validasi mengalami fluktuasi kecil di beberapa titik, yang mengindikasikan kemungkinan *overfitting*. Secara keseluruhan, model menunjukkan kinerja pelatihan yang luar biasa dengan penurunan kehilangan yang signifikan..

d. Evaluasi Model

Setelah model diuji dengan data pengujian, evaluasi kinerja dilakukan menggunakan matriks kekacauan, seperti yang ditunjukkan pada Gambar 14.



Gambar 14. Confusion Matrix

Gambar 14 menggambarkan kinerja model dalam mengklasifikasikan dua kelas, yaitu kunyit dan temulawak. Model berhasil mengklasifikasikan seluruh sampel kunyit dengan akurasi 48 dari 50 sampel di mana sampel kunyit teridentifikasi sebagai temulawak, sedangkan pada kelas temulawak tidak terdapat kesalahan dalam klasifikasi. Hasil ini menunjukkan kemampuan model yang luar biasa untuk membedakan kedua kelas dengan tingkat kesalahan yang sangat rendah. Hasil metrik evaluasi model untuk masing-masing kelas, termasuk ketepatan, recall, skor F1, dan dukungan, seperti yang ditunjukkan dalam laporan klasifikasi pada Gambar 15.

	precision	recall	f1-score	support
Kunyit	1.00	0.96	0.98	50
Temulawak	0.94	1.00	0.97	30
accuracy			0.97	80
macro avg	0.97	0.98	0.97	80
weighted avg	0.98	0.97	0.98	80

Gambar 15. Classification Report

Gambar 15 menyajikan hasil evaluasi kinerja model ditunjukkan pada Gambar dengan Akurasi total model adalah 97%. Untuk kunyit, diperoleh nilai *precision* sebesar 1,00, *recall* sebesar 0.96, dan *F1-score* sebesar 0.98 pada lima puluh data test; untuk temulawak, diperoleh nilai *precision* sebesar 0.94, *recall* sebesar 1.00, dan *F1-score* sebesar 0.97 pada 30 data test.

e. Menyimpan Model

Model yang telah dikembangkan disimpan dalam format .tflite dan diintegrasikan ke dalam aplikasi berbasis Android, sehingga memungkinkan pengguna untuk melakukan prediksi secara langsung melalui antarmuka aplikasi. Proses dan hasil penyimpanan model tersebut ditampilkan pada Gambar 16.

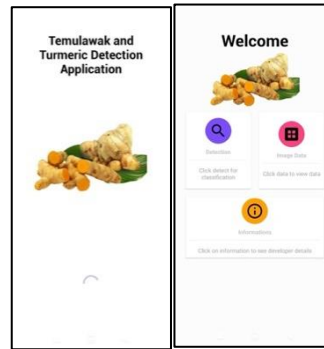
```
TFLite model saved as best_model_VGG16.tflite
Model berhasil disimpan di Kaggle output: /kaggle/working/best_model_VGG16.tflite
```

Gambar 16. Saving Model

Gambar 16 menunjukkan model yang berhasil disimpan dengan nama `best_model_VGG16.tflite`.

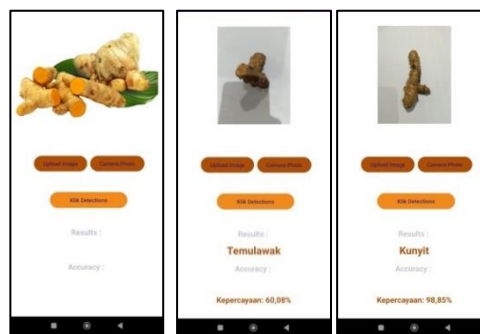
3.2 Implementasi Android

Setelah model disimpan, langkah selanjutnya adalah mengimplementasikan model tersebut ke dalam aplikasi Android yang dikembangkan menggunakan perangkat lunak Android Studio. Tampilan *splash screen* dan menu utama dari aplikasi yang telah dirancang ditampilkan pada Gambar 17.



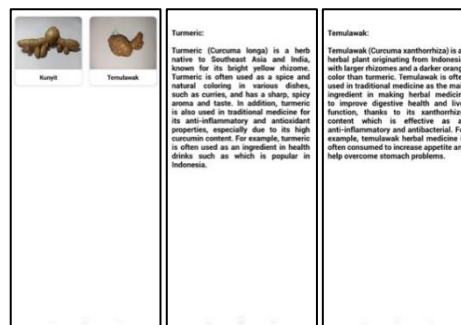
Gambar 17. Tampilan *Splashscreen* dan Menu Utama

Gambar 17 menunjukkan tampilan *splash screen* saat aplikasi dijalankan, yang kemudian dilanjutkan dengan tampilan menu utama. Adapun tampilan menu deteksi serta hasil klasifikasi dari aplikasi dapat dilihat pada Gambar 18.



Gambar 18. Tampilan Menu Deteksi dan Hasil Deteksi

Gambar 18 memperlihatkan tampilan menu deteksi beserta hasil deteksi pada aplikasi. Setelah pengguna mengunggah gambar atau mengambil foto dan memilih opsi deteksi, aplikasi akan menampilkan hasil klasifikasi beserta tingkat akurasi. Selanjutnya, tampilan menu data gambar dan informasi terkait dari aplikasi yang dikembangkan disajikan pada Gambar 19.



Gambar 19. Tampilan Menu Data Gambar dan Informasi gambar

Gambar 19 menunjukkan tampilan menu data gambar yang menampilkan contoh citra kunyit dan temulawak. Setelah pengguna memilih salah satu gambar, aplikasi akan menampilkan informasi terkait tanaman kunyit atau temulawak tersebut. Tampilan informasi lengkap dari aplikasi dapat dilihat pada Gambar 20.



Gambar 20. Tampilan Menu Informasi

Gambar 20 menyajikan informasi mengenai pengembang aplikasi, yang mencakup institusi, alamat, nomor telepon, serta alamat email.

3.3 Blackbox Testing

Setelah aplikasi berhasil dikembangkan, dilakukan pengujian untuk memastikan bahwa aplikasi berfungsi sesuai dengan persyaratan dan desain yang telah ditetapkan. Pengujian ini dilakukan menggunakan metode blackbox testing. Hasil pengujian tersebut disajikan pada Tabel 2.

Tabel 2. Blackbox Testing

No.	Fitur	Skenario Pengujian	Output yang diharapkan	Output Aktual	Hasil
1.	Tampilan <i>Splashscreen</i>	Pengguna membuka aplikasi	Menampilkan <i>splashscreen</i>	Menampilkan <i>splashscreen</i>	Sesuai
2.	Menu Utama	Pengguna membuka aplikasi	Menampilkan menu utama	Menampilkan menu utama	Sesuai
3.	Menu Deteksi	Pengguna memilih menu deteksi	Menampilkan menu deteksi	Menampilkan menu deteksi	Sesuai
4.	Upload/Ambil Gambar	Pengguna melakukan upload gambar atau mengambil gambar melalui kamera	Gambar berhasil diupload/diambil	Gambar berhasil diupload / diambil	Sesuai
5.	Melakukan Deteksi	Pengguna melakukan klik deteksi	Menampilkan hasil deteksi	Menampilkan hasil deteksi	Sesuai
6.	Menu Data Gambar	Pengguna memilih menu gambar dan memilih gambar	Menampilkan data gambar dan keterangannya	Menampilkan gambar beserta keterangannya	Sesuai
7.	Menu Informasi	Pengguna memilih menu informasi	Menampilkan Informasi	Menampilkan Informasi	Sesuai

Tabel 2 menunjukkan setelah dilakukan pengujian blackbox aplikasi yang dikembangkan dapat berjalan tanpa adanya kendala fungsional.

4. KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan sistem deteksi berbasis Android yang dapat membedakan kunyit dan temulawak dengan menggunakan arsitektur *Convolutional Neural Network* (CNN) dengan model VGG-16 yang dipadukan dengan *Fuzzy Tsukamoto*. Dengan mempertimbangkan tingkat keanggotaan fuzzy dalam kedua kelas (temulawak dan kunyit), lapisan tambahan ini dirancang untuk meningkatkan akurasi klasifikasi. Sistem ini dimaksudkan untuk membuat pengguna, terutama masyarakat umum dan industri herbal, lebih mudah membedakan kedua rimpang yang memiliki kemiripan visual. Uji coba yang dilakukan pada model VGG-16 yang dibuat menggunakan data uji menunjukkan bahwa model memiliki kemampuan klasifikasi yang seimbang dan dapat diandalkan untuk kedua kelas tersebut. Hasil uji menunjukkan bahwa penambahan lapisan fuzzy Tsukamoto menghasilkan nilai akurasi 97%, dengan nilai rata-rata untuk *precision* sebesar 0.97, *recall* sebesar 0.98, dan *F1-score* sebesar 0.97. Aplikasi ini juga terbukti berfungsi dengan stabil tanpa kendala teknis selama pengujian *blackbox*, sehingga siap untuk digunakan.

UCAPAN TERIMA KASIH

Penulis menyampaikan rasa terima kasih yang mendalam kepada Majelis Pendidikan Tinggi, Penelitian, dan Pengembangan (Diktilitbang) Pimpinan Pusat Muhammadiyah atas dukungan dan kepercayaan yang diberikan melalui Program Hibah RisetMu Batch 8. Bantuan pendanaan ini sangat berarti dalam menunjang kelancaran pelaksanaan penelitian serta pengembangan hasil yang bermanfaat bagi kemajuan ilmu pengetahuan dan teknologi.

REFERENCES

- [1] D. A. Widiyastuti, A. Ajizah, L. Nurtamara, N. Huda, and M. Afdal, "Pemberdayaan Ekonomi Skala Rumah Tangga Melalui Pembuatan Jamu Bubuk Rempah Temulawak, Jahe, Kunyit, dan Sereh," *J. Pengabd. ILUNG (Inovasi Lahan Basah Unggul)*, vol. 3, no. 2, p. 369, 2023, doi: 10.20527/ilung.v3i2.10298.
- [2] A. Z. Pramesthi, F. Purnamasari, and I. Setiawan, "Eksplorasi Keanekaragaman Hayati Tanaman Konsumsi di Pekarangan Desa Sumber RW 07 Surakarta Integrasi Fungsi Pangan dan Pengobatan," vol. 4, pp. 7865–7880, 2024, doi: <https://doi.org/10.31004/innovative.v4i6>.
- [3] A. Wilapangga, D. Rahmat, and R. Rachmaniar, "Formulasi dan Evaluasi Sediaan Gel Nanopartikel Ekstrak Temulawak (*Curcuma xanthorrhiza*) sebagai Tabir Surya," *Indones. J. Pharm. Educ.*, vol. 3, no. 1, pp. 26–32, 2023, doi: 10.37311/ijpe.v3i1.18854.
- [4] M. Y. Zamzam, Yuniarti Fayla, and Yossi Onirika Anggraeni, "UJI AKTIVITAS ANTIOKSIDAN KOMBINASI EKSTRAK

- KULIT BUAH NAGA MERAH (*Hylocereus polyrhizus*) DAN EKSTRAK KUNYIT (*Curcuma longa* L.) DENGAN METODE DPPH,” *Med. Sains J. Ilm. Kefarmasian*, vol. 8, no. 1, pp. 85–96, 2023, doi: <https://doi.org/10.37874/ms.v8i1.546>.
- [5] M. Mayasari, D. Iskandar Mulyana, M. Betty Yel, and S. Tinggi Ilmu Komputer Cipta Karya Informatika Jl Raden, “Komparasi Klasifikasi Jenis Tanaman Rimpang Menggunakan Principal Component Analisis, Support Vector Machine, K-Nearest Neighbor Dan Decision Tree,” *J. Tek. Inform. Kaputama*, vol. 6, no. 2, pp. 644–655, 2022.
 - [6] A. Arifin, J. Hendyli, and D. E. Herwindiati, “Klasifikasi Tanaman Obat Herbal Menggunakan Metode Support Vector Machine,” *Comput. J. Comput. Sci. Inf. Syst.*, vol. 5, no. 1, p. 25, 2021, doi: [10.24912/computatio.v1i1.12811](https://doi.org/10.24912/computatio.v1i1.12811).
 - [7] A. Effendi and Yantri Komala Dewi, “Analisis Bibliometrik Perancangan Arsitektur Dengan Kecerdasan Buatan,” *SARGA J. Archit. Urban.*, vol. 17, no. 1, pp. 48–63, 2023, doi: <https://doi.org/10.56444/sarga.v17i1.219>.
 - [8] G. A. Sandag and J. Waworundeng, “Identifikasi Foto Fashion Dengan Menggunakan Convolutional Neural Network (CNN),” *Cogito Smart J.*, vol. 7, no. 2, pp. 305–314, 2021, doi: <https://doi.org/10.31154/cogito.v7i2.340.305-314>.
 - [9] S. Salsabila, “Implementasi Algoritma Convolutional Neural Network Untuk Mendeteksi Pola Gerak Tangan Dalam Permainan Pong Ball Dengan Ai,” *Method. J. Tek. Inform. dan Sist. Inf.*, vol. 9, no. 2, pp. 8–11, 2023, doi: <https://doi.org/10.46880/mtk.v9i2.1995>.
 - [10] M. H. V. Sinaga, M. Albirra, and M. F. Sidiq, “Klasifikasi Gambar Pemandangan dengan Kecerdasan Buatan Berbasis CNN,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 412–417, 2024, doi: <https://doi.org/10.35870/jtik.v8i2.1424>.
 - [11] C. Hu, B. B. Sapkota, J. A. Thomasson, and M. V. Bagavathiannan, “Influence of image quality and light consistency on the performance of convolutional neural networks for weed mapping,” *Remote Sens.*, vol. 13, no. 11, 2021, doi: [10.3390/rs13112140](https://doi.org/10.3390/rs13112140).
 - [12] A. S. Irohoto Nozomi, “PREDIKSI PRODUKSI DAN PENJUALAN MENGGUNAKAN METODE FUZZY TSUKAMOTO,” vol. 1, no. 1, pp. 469–477, 2023, doi: <https://doi.org/10.62833/embistek.v3i2>.
 - [13] J. M. Jonson Manurung, B. S. Bosker Sinaga, P. M. H. Paska Marto Hasugian, L. Logaraj, and S. R. Sethu Ramen, “Analisis Algoritma C4.5 Dan Fuzzy Sugeno Untuk Optimasi Rule Base Fuzzy,” *J. Sist. Inf. dan Ilmu Komput. Prima (JUSIKOM PRIMA)*, vol. 5, no. 2, pp. 166–171, 2022, doi: <https://doi.org/10.34012/jurnalsisteminformasidanilmukomputer.v5i2.2488>.
 - [14] S. Bagas Valentino, “Klasifikasi Kualitas Daging Marmer Berdasarkan Citra Warna Daging Menggunakan Metode Convolutional Neural Network,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 1, pp. 125–129, 2023, doi: [10.36040/jati.v7i1.6128](https://doi.org/10.36040/jati.v7i1.6128).
 - [15] M. Yusuf, R. Ruimassa, A. I. Tawainella, and D. Maharani, “J-Icon : Jurnal Informatika dan Komputer KLASIFIKASI KUALITAS BERAS MENGGUNAKAN CONVOLUTIONAL NEURAL The importance of rice as a food commodity in Asia , especially in Indonesia , lies not only in its role as the main source of carbohydrates , but also in,” vol. 12, no. 2, pp. 186–192, 2024, doi: [10.35508/jicon.v12i2.18004](https://doi.org/10.35508/jicon.v12i2.18004).
 - [16] A. F. R. Muh. Fajrin Bakri, Muhammad Fajar B, Gebby Indriani, “Penentuan Jumlah Produksi Roti Pada Toko Roti Kayla Menggunakan Fuzzy Logic Metode Tsukamoto,” *J. Deep Learn. Comput. Vis. Digit. Image Process.*, vol. 2, no. 1, pp. 29–43, 2024, doi: <https://doi.org/10.61255/decoding.v2i1.300>.
 - [17] S. Arif Dwi Saputra, Ragil Wijianto Adhi, “Pengembangan Aplikasi Tamu Wajib Lapor Di Desa Karangsalam Baturaden Berbasis Android,” *Informatics Comput. Eng. J.*, vol. 3, no. 1, pp. 79–87, 2023, doi: <https://doi.org/10.31294/icej.v3i1.1783>.
 - [18] I. Salamah, S. Humairoh, and S. Soim, “Implementasi Convolutional Neural Network Pada Alat Klasifikasi Kematangan dan Ukuran Buah Nanas Berbasis Android,” *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 2, p. 243, 2023, doi: [10.35314/isi.v8i2.3413](https://doi.org/10.35314/isi.v8i2.3413).
 - [19] A. Safitri, A. Azzahra, and S. T. Kurnia, “Pengendalian Untuk Mengoptimalkan Produksi Mie Pada Warung Mie Pedas Dengan Menggunakan Logika Fuzzy Berbasis Metode Tsukamoto,” *J. Deep Learn. Comput. Vis. Digit. Image Process.*, vol. 2, no. 1, pp. 21–28, 2024, doi: <https://doi.org/10.61255/decoding.v2i1.316>.
 - [20] G. W. Intyanto, “Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network),” *J. Arus Elektro Indones.*, vol. 7, no. 3, p. 80, 2021, doi: <https://doi.org/10.19184/jaei.v7i3.28141>.
 - [21] A. Mufidatuzzaini and M. Faisal, “Penggunaan Teknik Transfer Learning pada Metode CNN untuk Pengenalan Tanaman Bunga,” vol. 10, no. 2, pp. 195–206, 2025.
 - [22] R. Soekarta, N. Nurdjan, and A. Syah, “Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN),” *Insect (Informatics Secur. J. Tek. Inform.)*, vol. 8, no. 2, pp. 143–151, 2023, doi: <https://doi.org/10.33506/insect.v8i2.2356>.
 - [23] F. R. B. Putra, M. R. Setyawan, and L. J. F. Rendra Soekarta, Nabila, “Implementasi Deep Learning Menggunakan Cnn Untuk Klasifikasi Tingkat Kematangan Buah Jeruk Berbasis Android,” vol. 8, no. 1, pp. 36–43, 2023, doi: [10.51544/jurnalmi.v9i2.5462](https://doi.org/10.51544/jurnalmi.v9i2.5462).
 - [24] M. Rizki Setyawan, “Sistem Pakar Deteksi Penyakit Kambing Menggunakan Certainty Factor Berbasis Android,” *J. MAHAJANA Inf.*, vol. 8, no. 1, pp. 36–43, Jun. 2023, doi: [10.51544/jurnalmi.v8i1.4008](https://doi.org/10.51544/jurnalmi.v8i1.4008).