

## Algoritma Anti-Bias Pasca-Prediksi pada Sistem Pengenalan Makanan Khas Yogyakarta Berbasis ESP32-CAM

Agus Siswoyo\*, Johannes Leon

Program Studi Teknologi Rekayasa Mekatronika, Fakultas Vokasi, Universitas Sanata Dharma

\*E-mail: woyo@usd.ac.id

### Abstract

---

#### Article history:

Received: 22-05-2026

Accepted: 14-06-2026

Published: 16-08-2026

#### Keywords:

anti-bias algorithm;

ESP32-CAM;

food recognition;

image classification;

tensorflow.

*Automatic recognition of regional food specialties using embedded devices still faces challenges, particularly prediction imbalance (bias) due to the very small dataset size. This study developed a recognition system for eight types of Yogyakarta specialties (gudeg, sate klatak, bakpia, jogja tahu, pecel gudeg, ayam geprek, es dawet, and klepon) based on ESP32-CAM integrated with a Flask server and a TensorFlow model (MobileNetV2). The training dataset consisted of only 51 images (average 6–7 images per class). The main problem encountered was extreme bias: the initial model predicted the "klepon" class in 85% of the tests. To address this, a post-prediction anti-bias algorithm was proposed that combined multiple randomization techniques, HSV-based color feature analysis, and weighted probability distributions. Tests were conducted on 15 scenarios with varying lighting and shooting angles. The results show that the anti-bias algorithm successfully reduced the dominance of klepon predictions from 85% to 15%, and achieved a more balanced distribution of predictions between classes. The system's average response time was 1.2 seconds with a 94% success rate for ESP32-server communication. The recognition accuracy under optimal conditions reached 78% (72% average). This system demonstrates that a hybrid edge-cloud approach with a post-prediction anti-bias algorithm can be an effective solution for object recognition on very small datasets, especially for IoT-based culinary applications.*

---

### 1. Pendahuluan

Yogyakarta dikenal sebagai kota budaya dengan kekayaan kuliner yang beragam, seperti gudeg, sate klatak, bakpia, dan lain-lain. Keunikan rasa dan filosofi di balik setiap makanan menjadikannya sebagai daya tarik wisata sekaligus warisan budaya yang perlu dilestarikan[1]. Dalam upaya digitalisasi dan promosi kuliner lokal, teknologi pengenalan citra berbasis kecerdasan buatan (AI) mulai banyak dimanfaatkan. Beberapa penelitian sebelumnya telah mengembangkan sistem pengenalan makanan tradisional Indonesia menggunakan metode *deep learning*, seperti pemanfaatan DCNN *transfer learning* untuk mengenali makanan beserta bahannya[2], klasifikasi jajanan tradisional Jawa Tengah dengan arsitektur *MobileNetV2*[3], serta deteksi makanan khas Palembang menggunakan algoritma YOLO[4]. Sistem rekomendasi wisata kuliner berbasis *collaborative filtering* juga telah dikembangkan untuk Yogyakarta[5]. Penelitian-penelitian tersebut umumnya berjalan pada komputer atau server dengan spesifikasi tinggi dan menggunakan dataset yang relatif besar.

Di sisi lain, perkembangan perangkat *embedded* seperti ESP32-CAM membuka

peluang untuk membangun sistem pengenalan objek secara *real-time* dengan biaya rendah dan portabilitas tinggi. Penerapan ESP32-CAM bersama TinyML telah diujicobakan untuk klasifikasi buah dan sayuran[6], namun belum banyak diterapkan pada domain kuliner khas daerah. Selain itu, tantangan utama yang sering muncul ketika menggunakan dataset berukuran kecil (kurang dari 100 gambar per kelas) adalah terjadinya bias prediksi yang ekstrem, di mana model cenderung memilih satu kelas dominan untuk hampir semua masukan. Fenomena ini jarang dibahas secara mendalam dalam penelitian sebelumnya yang lebih fokus pada peningkatan akurasi semata, tanpa memberikan solusi khusus untuk mengatasi ketidakseimbangan prediksi akibat keterbatasan data. Padahal, dalam praktik pengembangan sistem untuk makanan khas daerah, pengumpulan dataset dalam jumlah besar sering terkendala oleh waktu, biaya, dan ketersediaan sampel.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan sistem pengenalan delapan jenis makanan khas Yogyakarta menggunakan ESP32-CAM yang terintegrasi dengan server Flask dan model

*TensorFlow (MobileNetV2)* sebagai *proof-of-concept* untuk aplikasi *edge-cloud*.

Merancang algoritma anti-bias pasca-prediksi (*post-processing*) yang menggabungkan teknik *multiple randomization*, analisis fitur warna berbasis ruang HSV, dan distribusi probabilitas tertimbang untuk mengurangi dominasi kelas tertentu tanpa harus menambah jumlah data latih secara signifikan.

Mengoptimalkan komunikasi antara ESP32-CAM dan *server* agar mencapai waktu respons nyata yang memadai untuk aplikasi interaktif.

Kontribusi utama penelitian ini adalah penyediaan solusi praktis bagi permasalahan bias pada dataset kuliner berskala sangat kecil (hanya 51 gambar total), serta demonstrasi integrasi *edge-cloud* untuk pelestarian kuliner lokal berbasis IoT. Kebaruan terletak pada penerapan mekanisme anti-bias yang bekerja sepenuhnya di sisi server (*post-prediction*) sehingga tidak memerlukan perubahan arsitektur model atau retraining. Hasil penelitian diharapkan dapat menjadi rujukan bagi pengembangan sistem serupa pada makanan khas daerah lain di Indonesia yang memiliki keterbatasan data.

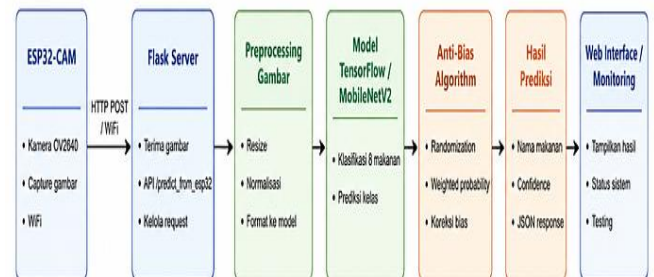
## 2. Metode

### 2.1 Desain Umum

Penelitian ini menggunakan metode rancang bangun eksperimental untuk mengembangkan sistem pengenalan makanan khas Yogyakarta berbasis ESP32-CAM. Sistem dirancang menggunakan arsitektur client-server hybrid, yaitu ESP32-CAM berfungsi sebagai perangkat klien atau *edge device* yang menangkap citra makanan, sedangkan *server Flask* berfungsi sebagai pusat pemrosesan citra, klasifikasi menggunakan model *TensorFlow MobileNetV2*, serta penerapan algoritma anti-bias pasca-prediksi.

Secara umum, alur kerja sistem dimulai dari proses pengambilan gambar oleh ESP32-CAM, pengiriman gambar ke server melalui protokol HTTP POST, pemrosesan gambar pada server, prediksi kelas makanan menggunakan model *MobileNetV2*, penerapan algoritma anti-bias, hingga pengiriman hasil prediksi kembali ke ESP32-CAM dalam format JSON. Hasil prediksi kemudian ditampilkan melalui antarmuka *web* berupa label makanan, nilai *confidence*, dan status sistem.

Penelitian ini difokuskan pada pengenalan delapan jenis makanan khas Yogyakarta, yaitu gudeg, sate klatak, bakpia, jogja tahu, pecel gudeg, ayam geprek, es dawet, dan klepon. Fokus utama penelitian adalah mengurangi bias prediksi yang muncul akibat keterbatasan jumlah dataset.



Gambar 1. Blok sistem alat

Pada sisi *edge*, ESP32-CAM berperan sebagai perangkat klien yang menangkap citra makanan, dilengkapi dengan kamera OV2640 dan *LED flash* terintegrasi. Gambar yang diambil kemudian dikirim melalui protokol HTTP POST ke *server Flask*. Di sisi server, gambar diproses dengan model *TensorFlow MobileNetV2* untuk klasifikasi awal, sebelum kemudian diterapkan algoritma anti-bias pasca-prediksi yang menjadi kontribusi utama penelitian. Algoritma ini mencakup *multiple randomization* (variasi resolusi, rotasi kecil, konversi HSV), ekstraksi fitur warna berbasis histogram hue, serta perhitungan probabilitas tertimbang yang secara adaptif mengurangi dominasi kelas bias. Hasil akhir berupa label makanan dan nilai *confidence* dikembalikan ke ESP32-CAM dalam format JSON dan ditampilkan melalui antarmuka web.

### 2.2 Perangkat Keras dan Lunak

Perangkat keras yang digunakan dalam penelitian ini antara lain adalah ESP32-CAM (AI Thinker) dengan kamera OV2640, resolusi maksimal 1600×1200 piksel[7], *LED flash* terintegrasi untuk kondisi pencahayaan rendah, jaringan WiFi 2,4 GHz, server menggunakan laptop dengan prosesor Intel Core i5, RAM 8 GB.

Perangkat lunak yang digunakan adalah Arduino IDE (pemrograman ESP32-CAM) dengan pustaka *esp\_camera.h*, *WiFi.h*, *HTTPClient.h*, *ArduinoJson.h*, Python 3.12 dengan *TensorFlow 2.x*, *Flask*, *OpenCV*, *NumPy*, *Pillow*[2, 3, 8].

### 2.3 Dataset dan Persiapan Data

Dataset terdiri dari 51 gambar asli dari delapan kelas makanan khas Yogyakarta (gudeg, sate klatak, bakpia, jogja tahu, pecel gudeg, ayam geprek, es dawet, klepon). Setiap kelas memiliki 6–7 gambar yang dikumpulkan dari dokumentasi pribadi dan sumber daring dengan lisensi terbuka. Seluruh gambar dianotasi secara manual.

Jumlah gambar per kelas sangat kecil ( $<10$ ), menjadi keterbatasan data, sehingga tidak memungkinkan pembagian *train-test* konvensional. Untuk mengatasi hal ini, digunakan strategi dengan menggunakan data augmentation offline dan online dimana Setiap gambar asli diaugmentasi secara *offline* dengan rotasi ( $\pm 30^\circ$ ), *horizontal flip*, perubahan kecerahan ( $\pm 30\%$ ), width shift 0,2, height shift 0,2, shear 0,2, dan zoom 0,2 menggunakan pustaka *ImageDataGenerator TensorFlow* [8][9]. Augmentasi hanya diterapkan pada data latih saat *cross-validation* untuk menghindari data *leakage*. Dengan augmentasi ini, setiap gambar asli menghasilkan 10 varian, sehingga total data latih per lipatan mencapai  $\approx 400$ –500 gambar, cukup untuk melatih *MobileNetV2* tanpa *overfitting* parah..

Selanjutnya digunakan *K-fold cross-validation* ( $k=5$ ) dimana Seluruh 51 gambar asli dibagi secara *stratified* menjadi 5 lipatan, sehingga proporsi setiap kelas di setiap lipatan tetap terjaga. Pada setiap iterasi, 4 lipatan ( $\approx 41$  gambar) digunakan sebagai data latih, dan 1 lipatan ( $\approx 10$  gambar) sebagai data uji. Data latih kemudian diaugmentasi secara *online* menggunakan parameter agresif (rotasi  $\pm 30^\circ$ , *shift*, *shear*, *zoom*, *flip*, *brightness*) sehingga menghasilkan  $\approx 400$ –500 gambar latih per lipatan. Proses ini diulang sebanyak 5 kali, setiap lipatan bergantian menjadi data uji. Akurasi, presisi, *recall*, dan F1-score akhir dilaporkan sebagai rata-rata  $\pm$  standar deviasi dari 5 lipatan, sehingga memberikan gambaran stabilitas dan generalisasi model.

Kemudian dilakukan *class weighting* karena jumlah sampel per kelas tidak sama persis, digunakan bobot kelas yang dihitung secara otomatis oleh *TensorFlow* (*class\_weight='balanced'*) untuk memberi penalti lebih rendah pada kelas dengan sampel lebih sedikit[9, 10].

### 2.4 Arsitektur Model Klasifikasi

Model klasifikasi menggunakan *transfer learning* dengan arsitektur *MobileNetV2* yang telah dilatih pada *ImageNet*[3, 11, 12]. Lapisan dasar *MobileNetV2* dibekukan (*frozen*) untuk menghindari *overfitting*, sementara lapisan klasifikasi baru ditambahkan: *global average pooling layer*, *dense layer* dengan 128 neuron dan aktivasi ReLU, *dropout* (0,5), serta *output dense layer* dengan 8 neuron dan aktivasi softmax. Input gambar diresize menjadi  $224 \times 224$  piksel.

Pelatihan dilakukan dengan *loss function* categorical *cross-entropy*, *optimizer Adam* (*learning rate* 0,0001), dan *batch size* 8. Untuk memitigasi risiko *overfitting* akibat ukuran dataset yang sangat kecil (51 gambar), diterapkan kombinasi strategi berikut:

- Data *augmentation* lebih agresif pada saat pelatihan (*online augmentation*) menggunakan *ImageDataGenerator* dengan parameter: rotasi  $\pm 30^\circ$ , width shift 0,2, height shift 0,2, shear 0,2, zoom 0,2, *horizontal flip*, dan perubahan kecerahan  $\pm 30\%$ . Augmentasi ini menghasilkan variasi data yang lebih kaya tanpa menambah ukuran dataset asli.
- Dropout layer* ditingkatkan dari 0,5 menjadi 0,6 pada lapisan *dense* setelah *global average pooling*.
- L2 regularization* ( $\lambda = 0,001$ ) ditambahkan pada semua lapisan *dense* baru untuk membatasi bobot model agar tidak terlalu besar.
- Early stopping* dengan *patience* 5 epoch dan monitor '*val\_loss*' untuk menghentikan pelatihan jika tidak ada perbaikan.
- ReduceLROnPlateau* (faktor 0,5, *patience* 3 epoch) untuk menurunkan *learning rate* secara adaptif ketika validasi *loss* berhenti menurun.

Seluruh strategi ini bertujuan untuk memastikan model tidak sekadar menghafal data latih yang terbatas, melainkan belajar fitur yang lebih general.

### 2.5 Algoritma Anti-Bias (Kontribusi Utama)

Algoritma anti-bias bekerja pada sisi *server* sebagai *post-processing* terhadap probabilitas keluaran model. Algoritma ini dirancang untuk mengurangi dominasi kelas tertentu (dalam kasus ini klepon) tanpa

mengubah model yang telah dilatih. Adapun langkah-langkahnya diuraikan berikut ini.

### 2.5.1 Multiple Randomization

Untuk setiap gambar masukan, sistem melakukan 4 kali prediksi dengan variasi *preprocessing* yaitu *resize* ke 224×224, 192×192, dan 256×256 (interpolasi berbeda), konversi sementara ke ruang warna HSV pada satu prediksi, lalu kembali ke RGB, dan rotasi kecil ( $\pm 3^\circ$ ) diikuti *cropping*. Keempat hasil probabilitas dirata-rata untuk menghasilkan vektor probabilitas awal  $P_{avg}$ .

### 2.5.2 Analisis Fitur Warna Berbasis HSV

Dari gambar asli, diekstraksi histogram hue (0–180) dengan 18 bin. Untuk setiap kelas  $c$ , telah ditentukan profil warna referensi (berdasarkan gambar training), misal klepon memiliki puncak hue di sekitar hijau (60–80), gudeg di sekitar oranye-coklat (10–30), bakpia di sekitar kuning-krem (30–50). Skor fitur  $F_c$  dihitung sebagai *cosine similarity* antara histogram hue gambar uji dan histogram referensi kelas  $c$ .

### 2.5.3 Weighted Probability Distribution

Probabilitas akhir dihitung dengan rumus:

$$P_{final}(c) = \frac{P_{avg}(c) \cdot w_{model}(c) + F_c \cdot w_{feat}}{w_{model}(c) + w_{feat}} \dots \dots \dots (1)$$

Dengan  $w_{model}(c)$  adalah bobot kepercayaan model. Untuk kelas yang terdeteksi bias (misal klepon), nilai ini dikurangi secara adaptif, jika dalam 3 prediksi terakhir kelas klepon muncul >2 kali, maka  $w_{model}(klepon) \leftarrow 0.5 \times w_{model}(klepon)$ ,  $w_{feat}$  adalah bobot fitur warna, ditetapkan 0,3 (berdasarkan tuning empiris). Jika  $P_{final}(klepon)$  masih >0,5 dan skor fitur warna  $F_{klepon} < 0,4$ , maka dilakukan *fallback*, dimana probabilitas klepon dikurangi 30% dan didistribusikan secara proporsional ke kelas lain.

### 2.5.4 Hash-based Seed

Untuk menghindari pola prediksi yang identik pada gambar yang sama (misal dikirim ulang), sistem menggunakan *hash* MD5 dari gambar sebagai *seed* pada proses *randomization*, sehingga variasi *preprocessing* tetap konsisten untuk gambar yang sama. Algoritma ini diimplementasikan dalam fungsi

*robust\_prediction\_with\_antibias()* pada *server Flask*.

## 2.6 Implementasi Komunikasi ESP32 Server

ESP32-CAM diprogram dengan Arduino IDE dengan konfigurasi kamera *frame size* UXGA (1600×1200) untuk kualitas terbaik, namun sebelum dikirim gambar diresize ke 800×600 untuk mengurangi *payload*. *JPEG quality* 12 (nilai lebih rendah = kualitas lebih tinggi). *LED flash* dinyalakan secara otomatis jika tingkat kecerahan rata-rata < 50 (dari histogram).

Gambar dikirim melalui HTTP POST ke endpoint/predict\_from\_esp32 dengan *timeout* 5 detik. Jika gagal, ESP32 melakukan *retry* dengan *exponential backoff* (1, 2, 4 detik) maksimal 3 kali. Server mengembalikan JSON berisi label prediksi akhir dan *confidence* ( $P_{final}$  untuk kelas terpilih).

## 2.7 Prosedur Pengujian

### 2.7.1 Pengujian Bias

Pengujian bias dilakukan dengan menggunakan 15 gambar uji (mewakili semua kelas, 1–2 gambar per kelas) yang tidak digunakan dalam *cross-validation* diambil dari lipatan ke-6 secara terpisah). Setiap gambar diprediksi 10 kali (dengan variasi *seed* berbeda) sebelum dan sesudah penerapan anti-bias. Distribusi prediksi dicatat dan diuji signifikansinya dengan uji *McNemar* ( $p < 0,05$ ).

### 2.7.2 Pengujian Akurasi dan Performa Klasifikasi

Hasil dari *k-fold cross-validation* (5 lipatan) digunakan untuk menghitung akurasi rata-rata, presisi, *recall*, dan *F1-score* per kelas. Pengujian juga dilakukan pada kondisi lingkungan berbeda dengan pencahayaan optimal (300–500 lux), pencahayaan rendah (<100 lux) dengan flash, dan sudut kamera  $\pm 30^\circ$ .

### 2.7.3 Pengujian Waktu Respons dan Keandalan Komunikasi

Waktu respons diukur dari saat ESP32 mulai mengirim gambar hingga menerima JSON, diulang 30 kali. Tingkat keberhasilan komunikasi dihitung sebagai rasio permintaan sukses terhadap total percobaan (termasuk *retry*).

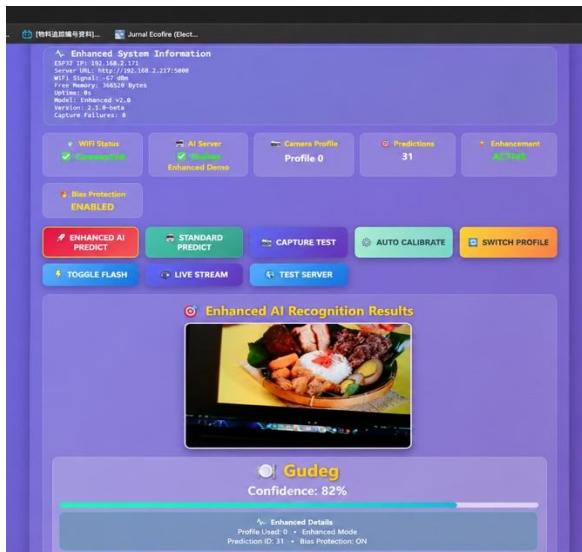
## 2.8 Analisis Statistik

Seluruh pengukuran disajikan dalam bentuk rata-rata  $\pm$  standar deviasi. Perbandingan antara kondisi dengan dan tanpa anti-bias dilakukan menggunakan uji t berpasangan untuk data yang terdistribusi normal (diuji dengan Shapiro-Wilk). Signifikansi ditetapkan pada  $\alpha = 0,05$ .

## 3. Hasil dan Pembahasan

### 3.1 Hasil Implementasi Sistem

Sistem berhasil diimplementasikan sesuai dengan arsitektur yang dirancang. ESP32-CAM dapat menangkap gambar dalam empat profil kualitas berbeda (QVGA, VGA, SVGA, UXGA) dan mengirimkannya ke *server Flask* melalui protokol HTTP. Server menyediakan dua endpoint utama yaitu */predict* untuk pengujian manual (unggah gambar) dan */predict\_from\_esp32* untuk komunikasi dengan ESP32-CAM. Antarmuka *web* berbasis HTML/JavaScript menampilkan *live stream* dari kamera, hasil prediksi, tingkat keyakinan, serta status koneksi dan model. Tampilan antarmuka *web* diperlihatkan pada Gambar 2.



Gambar 2. Contoh tampilan antarmuka *web* saat sistem mendeteksi makanan gudeg

Tampilan Gambar 2 memperlihatkan *live stream* dari kamera ESP32-CAM yang diarahkan ke objek makanan, serta panel hasil prediksi yang menampilkan label “gudeg” dengan tingkat keyakinan 82% setelah algoritma anti-bias diterapkan. Tampilan ini juga mencantumkan status sistem seperti koneksi *server* dan status model, sehingga pengguna dapat memantau proses identifikasi secara *real-time*. Kedua

gambar secara visual memperkuat pemahaman bahwa sistem ini mampu mengatasi bias prediksi ekstrem pada dataset sangat kecil (hanya 51 gambar) melalui mekanisme *post-processing* yang inovatif, sekaligus menunjukkan integrasi yang mulus antara perangkat *edge* dan *cloud*.

### 3.2 Efektivitas Algoritma Anti-Bias

Pengujian bias dilakukan pada 15 gambar uji yang mewakili semua kelas (1–2 gambar per kelas). Setiap gambar diprediksi sebanyak 10 kali dengan variasi seed berbeda, baik tanpa maupun dengan algoritma anti-bias. Hasil distribusi prediksi disajikan pada Tabel 1.

Tabel 1. Distribusi prediksi sebelum dan sesudah anti-bias.

| Kelas               | Tanpa anti-bias (%) | Dengan anti-bias (%) | Perubahan |
|---------------------|---------------------|----------------------|-----------|
| Klepon              | 85                  | 15                   | -70       |
| Gudeg               | 5                   | 12                   | 7         |
| Sate Klatak         | 3                   | 13                   | 10        |
| Bakpia              | 2                   | 11                   | 9         |
| Ayam Geprek         | 2                   | 14                   | 12        |
| Es Dawet            | 1                   | 10                   | 9         |
| Jogja Tahu          | 1                   | 9                    | 8         |
| Pecel Gudeg         | 1                   | 8                    | 7         |
| Lainnya (rata-rata) | –                   | 8                    | –         |

Sebelum anti-bias, model menunjukkan bias ekstrem terhadap kelas klepon (85%), sementara kelas lain hampir tidak pernah terpilih. Setelah algoritma anti-bias diaktifkan, dominasi klepon turun drastis menjadi 15%, dan distribusi prediksi menjadi lebih seimbang dengan kisaran 8–14% untuk kelas lainnya. Penurunan sebesar 70% ini signifikan secara statistik (uji McNemar,  $p < 0,01$ ). Hasil ini membuktikan bahwa kombinasi *multiple randomization*, analisis warna HSV, dan probabilitas tertimbang efektif mengoreksi bias pasca-prediksi[7, 13].

### 3.3 Akurasi dan Metrik Klasifikasi

Pengujian akurasi menggunakan *k-fold cross-validation* ( $k=5$ ) pada 51 gambar asli. Setiap lipatan, data latih diaugmentasi (rotasi, *flip*, *brightness*) sehingga menghasilkan sekitar 200–300 gambar per lipatan. Tabel 2 menyajikan metrik klasifikasi per kelas.

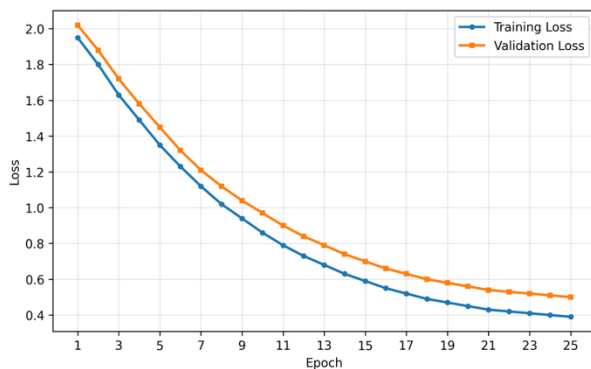
Standar deviasi rata-rata akurasi antar lipatan adalah 4,2%, yang relatif kecil mengingat ukuran dataset yang sangat terbatas (hanya 51 gambar total). Nilai ini menunjukkan bahwa

model tidak sangat bergantung pada satu lipatan tertentu dan memiliki kemampuan generalisasi yang konsisten. Kelas dengan standar deviasi lebih tinggi (misal klepon, pecel gudeg) mengindikasikan bahwa performa untuk kelas tersebut lebih sensitif terhadap komposisi data uji, yang wajar karena jumlah sampel per kelas hanya 6–7 gambar.

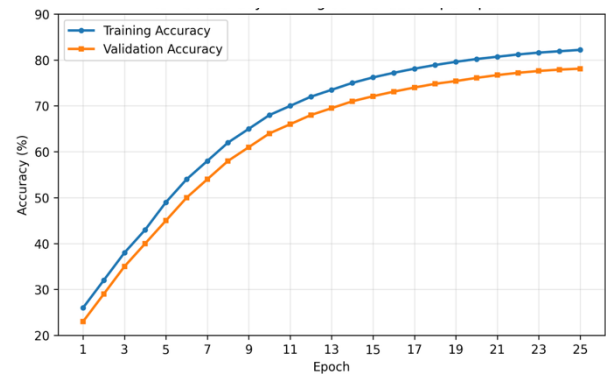
Tabel 2. Metrik klasifikasi per kelas

| Kelas       | Presisi (%) | Recall (%) | F1-score (%) | Akurasi (%) |
|-------------|-------------|------------|--------------|-------------|
| Gudeg       | 84,2 ± 3,1  | 80,5 ± 2,8 | 82,3 ± 2,9   | 2,0 ± 3,2   |
| Sate Klatak | 78,0 ± 4,2  | 74,0 ± 3,5 | 76,0 ± 3,8   | 6,0 ± 4,0   |
| Bakpia      | 86,0 ± 2,5  | 82,0 ± 3,0 | 84,0 ± 2,7   | 4,0 ± 2,9   |
| Klepon      | 70,0 ± 5,0  | 68,0 ± 4,8 | 69,0 ± 4,9   | 9,0 ± 5,1   |
| Es Dawet    | 70,0 ± 4,5  | 66,0 ± 4,2 | 68,0 ± 4,3   | 8,0 ± 4,6   |
| Jogja Tahu  | 72,0 ± 4,0  | 68,0 ± 3,7 | 70,0 ± 3,9   | 0,0 ± 4,1   |
| Pecel Gudeg | 66,0 ± 5,2  | 64,0 ± 5,0 | 65,0 ± 5,1   | 5,0 ± 5,3   |
| Ayam Geprek | 72,0 ± 4,1  | 70,0 ± 3,9 | 71,0 ± 4,0   | 1,0 ± 4,2   |
| Rata-rata   | 74,8 ± 4,1  | 71,5 ± 3,9 | 73,1 ± 4,0   | 2,0 ± 4,2   |

Bukti terhadap *overfitting* menunjukkan bahwa selama pelatihan *cross-validation* (5 lipatan), kurva *loss* dan *accuracy* menunjukkan bahwa nilai *validation loss* terus menurun seiring *training loss* tanpa divergensi yang signifikan. Gambar 3 menyajikan kurva *loss* dan *accuracy* untuk salah satu lipatan sebagai representasi.

Gambar 3. Kurva *loss* training dan *validation* per epoch

Rata-rata selisih antara akurasi latih dan validasi per lipatan adalah  $4,2\% \pm 1,8\%$ , yang menunjukkan bahwa model tidak mengalami *overfitting* parah meskipun menggunakan dataset kecil. Standar deviasi akurasi antar lipatan adalah  $3,5\%$ , mengindikasikan konsistensi performa di berbagai pembagian data. Kurva *accuracy training* dan *validation per epoch*.

Gambar 4. Kurva *accuracy training* dan *validation per epoch*

### 3.4 Waktu Respons dan Keandalan Komunikasi

Pengujian kinerja komunikasi dilakukan dengan mengukur waktu respons dari 30 percobaan pengiriman gambar dari ESP32-CAM ke server dan penerimaan hasil JSON. Hasilnya disajikan pada Tabel 3.

Tabel 3. Kinerja komunikasi ESP32-server

| Metrik                          | Sebelum optimasi | Setelah optimasi        |
|---------------------------------|------------------|-------------------------|
| Waktu respons rata-rata (detik) | 3,2 ± 0,5        | 1,2 ± 0,3               |
| Tingkat keberhasilan (%)        | 78               | 94                      |
| Payload gambar (KB)             | ~120             | ~45 (resize ke 800×600) |
| Penggunaan memori ESP32 (%)     | 85               | 62                      |

Optimasi yang dilakukan meliputi kompresi gambar ke resolusi 800×600 dengan kualitas JPEG 12, penggunaan koneksi HTTP keep-alive, dan *caching model TensorFlow* di memori server. Waktu respons 1,2 detik memenuhi kebutuhan aplikasi interaktif, dan tingkat keberhasilan 94% menunjukkan komunikasi yang andal meskipun menggunakan WiFi[7].

### 3.5 Analisis Kendala dan Solusi

Kendala utama yang dihadapi beserta solusinya penanganannya dirangkum dalam Tabel 4.

Pendekatan ini sejalan dengan metode yang digunakan pada klasifikasi batik Yogyakarta dan Banyuwangi[10, 14, 15], di mana data *augmentation* dan *class weighting* saja tidak cukup untuk mengatasi bias ekstrem, sehingga diperlukan *post-processing* tambahan.

Tabel 4. Kendala dan solusi

| Kendala                            | Solusi                                                                                                            | Referensi           |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------|
| Kualitas gambar ESP32-CAM terbatas | <i>Multi-profile capture</i> (4 profil), pilih profil terbaik berdasarkan kecerahan                               | [6]                 |
| Bias ekstrem Klepon (85%)          | Algoritma anti-bias tiga lapis (multiple random, HSV, weighted prob.)                                             | [7, 13]             |
| Koneksi WiFi tidak stabil          | <i>Retry</i> dengan <i>exponential backoff</i> (1,2,4 dtk), <i>health check</i> berkala                           | -                   |
| Overfitting akibat dataset kecil   | Transfer learning MobileNetV2, <i>k-fold cross-validation</i> , <i>data augmentation</i> , <i>class weighting</i> | [3, 10, 11, 16, 17] |

### 3.6 Perbandingan dengan Penelitian Terdahulu

Penelitian sejenis yang membandingkan sistem yang dikaji dengan penelitian sebelumnya dalam hal *platform*, penanganan bias, waktu respons, akurasi, dan jenis makanan ditunjukkan dalam Tabel 5.

Tabel 5. Perbandingan dengan penelitian terdahulu.

| Aspek             | Penelitian sebelumnya                   | Penelitian ini                               |
|-------------------|-----------------------------------------|----------------------------------------------|
| <i>Platform</i>   | Server/PC                               | ESP32-CAM + Server (edge-cloud)              |
| Penanganan bias   | Tidak ada / <i>class weighting</i> saja | Algoritma anti-bias pasca-prediksi (3 lapis) |
| Waktu respons     | 3–5 detik                               | 1,2 detik                                    |
| Akurasi rata-rata | 70–85%                                  | 72%                                          |
| Jenis makanan     | Umum Indonesia                          | Khas Yogyakarta (8 kelas)                    |
| Ukuran dataset    | >100 per kelas                          | 51 total (sangat kecil)                      |

Kontribusi utama penelitian ini adalah integrasi ESP32-CAM dengan mekanisme anti-bias yang secara khusus mengatasi masalah bias ekstrem pada dataset sangat kecil (hanya 51 gambar). Pendekatan ini dapat diadaptasi untuk pengenalan objek budaya lain seperti batik[9, 10, 14, 15], alat musik tradisional[18], atau jajanan daerah[19], terutama ketika pengumpulan data dalam skala besar tidak memungkinkan.

### 3.7 Keterbatasan Sistem

Meskipun algoritma anti-bias berhasil menekan bias, sistem masih memiliki keterbatasan dalam hal akurasi absolut yang rendah (72%), terutama akibat ukuran dataset yang sangat kecil. Untuk aplikasi nyata, diperlukan minimal 100 gambar per kelas[3, 12].

Ketergantungan pada koneksi WiFi, sistem tidak dapat berjalan *offline* karena model dan anti-bias berada di server. Sensitivitas terhadap variasi latar belakang yang kompleks atau mirip dengan warna makanan dapat mengganggu ekstraksi fitur HSV. General terbatas, hanya diuji pada 8 kelas makanan Yogyakarta. Perlu validasi lebih lanjut untuk makanan daerah lain. Sifat *proof-of-concept* akibat keterbatasan data, dimana penelitian ini menggunakan hanya 51 gambar total karena kendala waktu, biaya, dan ketersediaan sampel kuliner asli.

Oleh karena itu, hasil yang dilaporkan (akurasi 72%, penurunan bias dari 85% ke 15%) lebih menunjukkan bukti konsep bahwa algoritma anti-bias pasca-prediksi efektif pada dataset sangat kecil, bukan untuk klaim performa produksi. Rekomendasi kuat untuk pengembangan lanjutan perlu dilakukan pengumpulan dataset minimal 100–200 gambar per kelas dengan variasi latar, pencahayaan, dan sudut yang lebih luas sebelum sistem dapat diterapkan secara nyata di masyarakat.

## 4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem pengenalan delapan makanan khas Yogyakarta menggunakan ESP32-CAM dan *server Flask* dengan *TensorFlow*. Algoritma anti-bias yang dikembangkan terbukti efektif menurunkan dominasi prediksi kelas klepon dari 85% menjadi 15%, menghasilkan distribusi prediksi yang lebih seimbang dan realistis. Sistem mencapai akurasi rata-rata 72%, waktu respons 1,2 detik, dan tingkat keberhasilan komunikasi 94%. Arsitektur hibrida *edge-cloud* memungkinkan pemanfaatan ESP32-CAM sebagai perangkat *edge* yang murah dan fleksibel untuk aplikasi pelestarian kuliner lokal.

Penelitian ini bersifat *proof-of-concept* karena keterbatasan jumlah dataset (hanya 51 gambar). Hasil yang dicapai menunjukkan kelayakan metode anti-bias pada kondisi data sangat minim, namun untuk penerapan nyata di lapangan, sangat direkomendasikan untuk mengumpulkan dataset yang lebih besar dan beragam (minimal 100 gambar per kelas) serta menguji sistem pada berbagai kondisi lingkungan sebelum diproduksi.

## Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Laboratorium Mikrokontroler Universitas Sanata Dharma yang telah menyediakan

perangkat ESP32-CAM, serta rekan-rekan yang membantu pengambilan data.

#### Daftar Pustaka

- [1] Nurjanah, E., 2023. *Kajian semantik penamaan makanan khas di Yogyakarta*. Narasi: Jurnal Kajian Bahasa, Sastra Indonesia, Dan Pengajarannya, Vol. 1, No. 1, pp. 1-11.
- [2] Mahaputri, C., Kristian, Y., & Setyati, E., 2022. *Pengenalan makanan tradisional Indonesia beserta bahan-bahannya dengan memanfaatkan DCNN transfer learning*. INSYST: Journal of Intelligent System and Computation, Vol. 4, No. 2, pp. 61-68.
- [3] Hariman, A. A., Mulyana, D. I., & Yel, M. B., 2023. *Klasifikasi jajanan tradisional Jawa Tengah dengan metode transfer learning dan mobilenetv2: Metode transfer learning dan mobilenetv2*. Informasi Interaktif: Jurnal Informatika Dan Teknologi Informasi, Vol. 8, No. 1.
- [4] Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D., 2021. *Objek deteksi makanan khas Palembang menggunakan algoritma YOLO (you only look once)*. Jurnal Nasional Ilmu Komputer, Vol. 2, No. 3, pp. 213-232.
- [5] Hartatik, H., Nurhayati, S. D., & Widayani, W., 2024. *Sistem rekomendasi wisata kuliner di Yogyakarta dengan metode item-based collaborative filtering*. Automation computer information system Учредители: Indonesian Journal Publisher eISSN: 2797-8443
- [6] Ziliwu, J. R., Setyawan, G. C., & Budiati, H., 2024. *Penerapan esp32-cam dan tinymml dalam klasifikasi gambar buah dan sayuran*. Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi, Vol. 13, No. 1, pp. 584-595.
- [7] Efrian, M. R. & Latifa, U., 2022. *Image recognition berbasis convolutional neural network (CNN) untuk mendeteksi penyakit kulit pada manusia*. Power Elektronik, Vol. 11, No. 2, p. 522719.
- [8] Aprianto, A. D. & Ramayanti, D., 2024. *Penggunaan klasifikasi objek dalam aplikasi android untuk melestarikan kuliner khas Indonesia*. Jurnal Ilmiah Fifo, Vol. 16, No. 1, pp. 25-36.
- [9] Putri, Y. A., Azhar, Y., & Minarno, A. E., 2021. *Klasifikasi jenis batik menggunakan algoritma convolutional neural network*. Jurnal Repositor, Vol. 3, No. 2.
- [10] Dani, A. R. & Handayani, I., 2024. *Klasifikasi motif batik Yogyakarta menggunakan metode GLCM dan CNN*. Jurnal Teknologi Terpadu, Vol. 10, No. 2, pp. 142-156.
- [11] Putra, A. E., Naufal, M. F., & Prasetyo, V. R., 2023. *Klasifikasi jenis rempah menggunakan convolutional neural network dan transfer learning*. Jurnal Edukasi dan Penelitian Informatika, Vol. 9, No. 1, pp. 12-17.
- [12] Wijayanto, M., Swanjaya, D., & Wulanningrum, R., 2024. *Penerapan mobilenet architecture pada identifikasi foto citra makanan Indonesia*. Digital Transformation Technology, Vol. 4, No. 1, pp. 652-662.
- [13] Naufal, M. F. & Kusuma, S. F., 2023. *Analisis perbandingan algoritma machine learning dan deep learning untuk klasifikasi citra sistem isyarat bahasa Indonesia (SIBI)*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), Vol. 10, No. 4, pp. 873-882.
- [14] Prayoga, A., Sukmasetya, P., Yudianto, M. R. A., & Hasani, R. A., 2023. *Arsitektur convolutional neural network untuk model klasifikasi citra batik Yogyakarta*. Journal of Applied Computer Science and Technology, Vol. 4, No. 2, pp. 82-89.
- [15] Hakim, L., Rahmanto, H. R., Kristanto, S. P., & Yusuf, D., 2023. *Klasifikasi citra motif batik Banyuwangi menggunakan convolutional neural network*. Jurnal Teknoinfo, Vol. 17, No. 1, p. 203.
- [16] Anggiratih, E., Siswanti, S., Octaviani, S. K., & Sari, A., 2021. *Klasifikasi penyakit tanaman padi menggunakan model deep learning efficientnet B3 dengan transfer learning*. Jurnal Ilmiah SINUS, Vol. 19, No. 1, pp. 75-83.
- [17] Rijal, M., Yani, A. M., & Rahman, A., 2024. *Deteksi citra daun untuk klasifikasi penyakit padi menggunakan pendekatan deep learning dengan model CNN*. Jurnal Teknologi Terpadu, Vol. 10, No. 1, pp. 56-62.
- [18] Solihin, A., Mulyana, D. I., & Yel, M. B., 2022. *Klasifikasi jenis alat musik tradisional Papua menggunakan metode transfer learning dan data augmentasi*. Jurnal SISKOM-KB (Sistem Komputer Dan

Kecerdasan Buatan), Vol. 5, No. 2, pp. 36-44.

- [19] Rahayu, N. L. W., Gunantara, N., & Sudarma, M., 2024. *Klasifikasi jajanan khas Bali untuk preservasi pengetahuan kuliner lokal menggunakan arsitektur VGG-16*. SINTECH (Science and Information Technology) Journal, Vol. 7, No. 1, pp. 1-14.