JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i3.9984

PENERAPAN *YOU ONLY LOOK ONCE* VERSI 8 (YOLOv8) DALAM MENDETEKSI CITRA JENIS MAKANAN UNTUK MENAMPILKAN INFORMASI NUTRISI BERBASIS WEB

Febri Haerani[*], LM. Bahtiar Aksara[2], LM. Golok Jaya[3]

[1,2,3]Prodi Teknik Informatika, Universitas Halu Oleo, Kampus Hijau Bumi Tridharma, Anduonohu, Kec.Kambu, Kota Kendari, Sulawesi Tenggara, 93232

Keywords:
Nutritional information; Food recognation; YOLOv8; Object detection.

Corespondent Email:
febrihaerani5@gmail.com

Abstrak. Informasi nutrisi makanan memiliki peran penting dalam membantu masyarakat memantau asupan gizi harian. Namun, identifikasi jenis makanan secara manual masih kurang praktis dan berpotensi menimbulkan kesalahan. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem deteksi jenis makanan berbasis citra yang mampu menampilkan informasi nutrisi secara otomatis. Metode yang digunakan adalah *You Only Look Once* versi 8 Nano (YOLOv8n), yang dipilih karena memiliki kecepatan inferensi tinggi dengan kebutuhan komputasi yang relatif ringan. Dataset dikumpulkan melalui dua sumber, yaitu internet dan pengambilan langsung dengan perbandingan 2:1, menghasilkan 2.000 citra untuk 13 kelas makanan. Hasil penelitian menunjukkan bahwa model YOLOv8n mampu mencapai nilai mAP@0.5 sebesar 89,3% pada data uji. Beberapa kelas makanan memperoleh nilai mAP di atas 90%, sedangkan kelas lainnya berada pada rentang 70–80%. Selain itu, sistem berhasil mengintegrasikan API FatSecret untuk menampilkan informasi nutrisi berdasarkan hasil deteksi serta melakukan pemantauan asupan nutrisi harian pengguna. Hasil penelitian menunjukkan bahwa YOLOv8n dapat digunakan secara efektif untuk mendeteksi jenis makanan dan mendukung penyediaan informasi nutrisi secara otomatis.

Abstract. *Food nutrition information plays an important role in helping individuals monitor their daily dietary intake. However, manually identifying food types is often impractical and prone to errors. Therefore, this study aims to develop an image-based food detection system capable of automatically displaying nutritional information. The method employed in this research is You Only Look Once version 8 Nano (YOLOv8n), which was selected due to its high inference speed and relatively low computational requirements. The dataset was collected from two sources, namely the internet and direct image acquisition, with a ratio of 2:1, resulting in a total of 2,000 images across 13 food classes. The experimental results show that the YOLOv8n model achieved an mAP@0.5 score of 89.3% on the test dataset. Several food classes obtained mAP values above 90%, while the remaining classes achieved scores ranging from 70% to 80%. In addition, the system successfully integrated the FatSecret API to retrieve and display nutritional information based on the detected food items, as well as to monitor users' daily nutritional intake. These findings demonstrate that YOLOv8n can be effectively utilized for food type detection and can support the automated provision of nutritional information.*

1. PENDAHULUAN

Kesadaran masyarakat global terhadap pentingnya kesehatan dan gizi seimbang dalam beberapa tahun terakhir terus meningkat. Namun, masalah malnutrisi masih menjadi tantangan besar. Menurut *World Health Organization* (WHO), lebih dari 1,9 miliar orang dewasa mengalami kelebihan berat badan dan 462 juta orang mengalami kekurangan berat badan. Kondisi ini tidak hanya terjadi di negara maju, tetapi juga di negara berkembang seperti Indonesia yang mengalami perubahan gaya hidup dan peningkatan konsumsi makanan cepat saji. Berdasarkan wawancara dengan Ibu Sri Ida Nasaruddin Toby, S.K.M., kesadaran masyarakat Indonesia dalam menjaga gizi seimbang dan memperhatikan kandungan nutrisi makanan masih tergolong rendah. Pola konsumsi makanan yang tinggi gula, garam, dan lemak menjadi salah satu faktor utama meningkatnya penyakit tidak menular seperti diabetes, hipertensi, *stroke,* dan penyakit jantung. Data menunjukkan prevalensi diabetes meningkat dari 1,5 permil pada tahun 2013 menjadi 2 permil pada tahun 2018, gagal ginjal kronis dari 2 permil menjadi 3,8 permil, serta stroke dari 7 permil menjadi 10,9 permil. Selain itu, survei Kementerian Kesehatan RI tahun 2022 menunjukkan bahwa 28,7% masyarakat Indonesia mengonsumsi gula, garam, dan lemak melebihi batas yang direkomendasikan (Kemenkes, 2022). Kondisi ini menunjukkan perlunya solusi yang dapat membantu masyarakat memahami kandungan gizi makanan secara lebih mudah sehingga dapat mendukung pola hidup yang lebih sehat.

Perkembangan teknologi informasi dan kecerdasan buatan membuka peluang untuk menghadirkan sistem yang mampu mendeteksi jenis makanan sekaligus menyajikan informasi gizi secara otomatis. Melalui sistem berbasis pengolahan citra dan deteksi objek, proses identifikasi makanan dapat dilakukan secara cepat dan praktis sehingga memudahkan pengguna dalam memperoleh informasi terkait kandungan nutrisi makanan yang dikonsumsi.

Beberapa penelitian sebelumnya telah menunjukkan keberhasilan penerapan algoritma YOLO pada bidang deteksi makanan. Penelitian yang dilakukan oleh [1]dengan judul *Detection of Indonesian Food to Estimate Nutritional Information Using* YOLOv5 memperoleh rata-rata akurasi sebesar 98,6%, *precision* 95%, *recall* 95,3%, dan F1-*Score* 95%. Sistem yang dikembangkan mampu mengidentifikasi makanan Indonesia menggunakan YOLOv5 dan menampilkan informasi energi, protein, lemak, serta karbohidrat melalui integrasi dengan FatSecret Indonesia. Penelitian lain oleh [2] yang berjudul Evaluasi Kinerja Aplikasi Mobile Penghitung Kalori Makanan Berbasis Algoritma CNN-YOLO menunjukkan hasil yang sangat baik dengan nilai presisi 1,00, *recall* 0,99, *confidence* 0,962, dan F1-*Score* 0,97, serta mampu mencapai akurasi 100% dalam mendeteksi seluruh kelas makanan. Sementara itu, penelitian [3] mengenai aplikasi estimasi kalori makanan berbasis citra menggunakan YOLO memperoleh nilai mAP@0.5 sebesar 0,975 dan berhasil menampilkan estimasi kalori makanan yang terdeteksi melalui integrasi dengan dataset FatSecret. Hasil-hasil tersebut menunjukkan bahwa algoritma YOLO memiliki performa yang baik untuk deteksi makanan dan penyajian informasi gizi.

Setelah makanan berhasil dideteksi, hasil deteksi perlu dicocokkan dengan data gizi yang sesuai. Perbedaan penulisan maupun kemiripan nama makanan sering menjadi kendala dalam proses tersebut. Oleh karena itu, diperlukan *Application Programming Interface* (API) sebagai penghubung antara sistem deteksi makanan dan basis data gizi. Melalui API, sistem dapat mengambil serta menampilkan informasi kandungan protein, lemak, karbohidrat, dan nutrisi lainnya secara otomatis berdasarkan makanan yang terdeteksi.

Berdasarkan hasil kajian literatur, penelitian ini menggunakan YOLOv8 karena mampu memberikan keseimbangan yang baik antara kecepatan dan akurasi deteksi. YOLOv8 dapat melakukan deteksi dan klasifikasi objek dalam satu proses inferensi sehingga sesuai untuk aplikasi real-time. Selain itu, peningkatan pada arsitekturnya membuat YOLOv8 lebih efektif dalam mendeteksi makanan dengan variasi ukuran, bentuk, warna, dan tekstur, termasuk beberapa jenis makanan yang disajikan dalam satu piring. Hasil deteksi kemudian dihubungkan dengan API FatSecret untuk memperoleh informasi nutrisi secara otomatis.

Kebaruan penelitian ini terletak pada fitur pemantauan nutrisi harian. Informasi nutrisi yang diperoleh dari setiap hasil deteksi tidak hanya ditampilkan kepada pengguna, tetapi

juga disimpan dan dikelola sebagai riwayat konsumsi harian. Dengan demikian, pengguna dapat memantau total asupan nutrisi yang telah dikonsumsi dan mengevaluasi tingkat pemenuhan kebutuhan gizinya setiap hari. Sistem yang dikembangkan mencakup deteksi makanan berbasis YOLOv8, integrasi data nutrisi melalui API FatSecret, fitur monitoring dan riwayat nutrisi harian, *backend*, serta antarmuka pengguna yang terintegrasi.

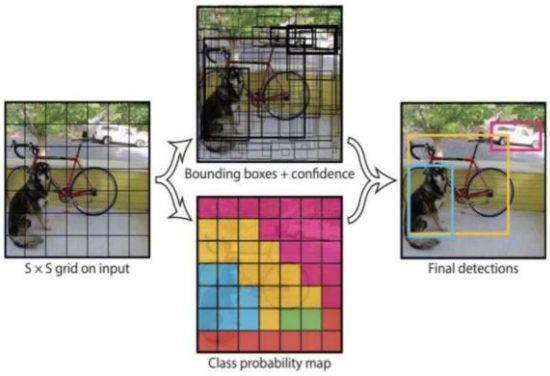
2. TINJAUAN PUSTAKA

2.1 *Deep Learning*

Deep Learning merupakan salah satu cabang dari *machine learning* yang berfokus pada pengembangan algoritma yang terinspirasi dari struktur dan cara kerja otak manusia, yang dikenal sebagai jaringan saraf tiruan[4]**.** Teknologi ini memungkinkan sistem untuk memahami dan menganalisis citra secara lebih mendalam. Dalam penerapannya, *deep learning* memanfaatkan jaringan saraf dengan banyak lapisan untuk mengolah data dalam jumlah besar[5]. Melalui proses tersebut, model dapat mempelajari pola-pola yang kompleks secara otomatis serta menghasilkan keputusan atau prediksi berdasarkan informasi yang diperoleh dari data[6].

2.2 *You Only Look Once* (YOLO)

YOLO merupakan salah satu metode deteksi objek yang menggunakan pendekatan *unified model,* di mana proses deteksi dilakukan oleh satu jaringan saraf (*single neural network*) secara terpadu. Melalui pendekatan ini, model mampu memprediksi *bounding box* dan probabilitas kelas objek secara bersamaan langsung dari sebuah gambar dalam satu kali proses[7]. Keunggulan utama YOLO terletak pada kecepatannya dalam melakukan deteksi, Model YOLO dapat memproses gambar dengan kecepatan hingga 45 FPS sedangkan versi yang lebih ringan, yaitu Fast YOLO, mampu mencapai kecepatan hingga 155 FPS. Performa tersebut menjadikan YOLO sebagai salah satu algoritma deteksi objek tercepat dibandingkan berbagai metode deteksi *real-time* lainnya[8], [9], [10]. Arsitektur YOLO ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur YOLO

2.3 *Application Programming Interface* (API)

Application Programming Interface (API) merupakan antarmuka yang memungkinkan suatu aplikasi berkomunikasi dan bertukar data dengan aplikasi lainnya. API juga dapat dipahami sebagai sekumpulan fungsi, perintah, dan protokol yang disediakan untuk membantu pengembang dalam membangun perangkat lunak pada sistem operasi tertentu[11]. API berfungsi sebagai penghubung antara aplikasi dan basis data sehingga memungkinkan akses serta eksekusi berbagai perintah. Dalam implementasinya, *web service* berperan sebagai perantara antara antarmuka pengguna dan server untuk memproses permintaan yang dikirimkan aplikasi [12]. Salah satu arsitektur API yang sering digunakan adalah *Representational State Transfer* (REST) [13].

REST (*Representational State Transfer*) adalah gaya arsitektur yang digunakan untuk membangun komunikasi antar sistem berbasis *web* dengan memanfaatkan protokol HTTP. Pada penerapannya, setiap sumber daya diidentifikasi melalui URL (*Uniform Resource Identifier*) yang unik dan dapat diakses menggunakan metode HTTP seperti *GET, POST, PUT,* dan *DELETE* melalui *endpoint* yang tersedia [14].

2.5 Flask

Flask merupakan *micro-framework* berbasis Python yang digunakan untuk pengembangan aplikasi *web*. Flask dirancang dengan fitur inti yang sederhana dan tidak menyediakan komponen umum seperti validasi formulir maupun pengelolaan basis data secara bawaan [15]. Namun, Flask mendukung berbagai ekstensi yang memungkinkan pengembang menambahkan fungsionalitas sesuai kebutuhan,

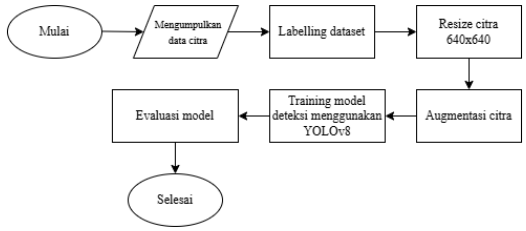
sehingga *framework* ini bersifat fleksibel dan mudah dikembangkan [16].

2.6 Google Colab

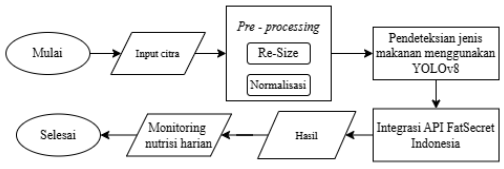
Google Colab merupakan layanan berbasis *cloud computing* yang dikembangkan oleh Google untuk mendukung aktivitas pemrograman menggunakan bahasa Python. *Platform* ini memungkinkan pengguna menjalankan dan mengeksekusi kode secara langsung melalui lingkungan komputasi awan, sehingga tidak diperlukan proses instalasi perangkat lunak maupun konfigurasi lingkungan pengembangan pada laptop atau komputer pribadi [17].

3. METODE PENELITIAN

Penelitian ini mengembangkan sebuah sistem *end-to-end* yang mampu menyajikan informasi nutrisi berdasarkan citra makanan yang diunggah pengguna. Sistem bekerja dengan mengidentifikasi jenis makanan pada gambar, kemudian menampilkan informasi nutrisi yang sesuai melalui antarmuka berbasis *web.* Proses penelitian dilakukan melalui beberapa tahapan, meliputi studi literatur, pengumpulan dataset, *preprocessing* data, perancangan dan pelatihan model YOLOv8, pengembangan antarmuka pengguna menggunakan *framework* Flask, serta integrasi API dengan FatSecret Indonesia sebagai sumber data nutrisi. Alur pemodelan dan mekanisme kerja sistem ditunjukkan pada Gambar 1 dan Gambar 2.



Gambar 2. Alur proses *modelling*



Gambar 3. Alur kerja sistem

3.1. Dataset

Tahap pengumpulan data dilakukan dengan memperoleh citra makanan dari dua sumber, yaitu internet dan hasil pengambilan gambar secara langsung, dengan perbandingan jumlah data 2:1. Total dataset yang digunakan sebanyak 2.000 citra yang terbagi ke dalam 13 kelas makanan, dengan jumlah gambar pada setiap kelas berkisar antara 100 hingga 400 citra. Setelah proses pengumpulan selesai, setiap citra diberi anotasi secara manual menggunakan aplikasi Roboflow. Pemilihan jenis makanan didasarkan pada lauk yang umum dikonsumsi oleh masyarakat Indonesia. Daftar kelas makanan yang digunakan ditampilkan pada Tabel 1.

Tabel 1. Daftar kelas makanan

No	Nama Makanan
1	Ayam Goreng
2	Ayam Goreng Tepung
3	Ikan Goreng Bandeng
4	Nasi Putih
5	Perkedel Jagung
6	Sayur Bening Bayam
7	Singkong Rebus
8	Tahu Goreng
9	Telur Dadar
10	Telur Rebus
11	Tempe Goreng
12	Tumis Buncis
13	Tumis Kangkung

Setelah proses pelabelan selesai, dataset dibagi menjadi tiga kelompok, yaitu 80% data *training*, 10% data *validation*, dan 10% data *testing*. Pembagian data tersebut dilakukan untuk mendukung proses pelatihan dan evaluasi model YOLO sehingga diperoleh kinerja yang optimal. Selanjutnya, dilakukan tahap *preprocessing* dengan menyeragamkan ukuran seluruh citra menjadi 640×640 piksel. Untuk meningkatkan variasi data dan memperkuat kemampuan model dalam mengenali objek pada berbagai kondisi, diterapkan teknik augmentasi data yang meliputi *flip*, *rotation*, dan penyesuaian tingkat kecerahan (*brightness*).

3.2 Pelatihan Model

Proses perancangan dan pelatihan model dilakukan menggunakan platform Google Colab dengan memanfaatkan fasilitas GPU

untuk mendukung komputasi. Arsitektur yang digunakan adalah YOLOv8n karena memiliki ukuran model yang lebih ringan dibandingkan varian YOLOv8 lainnya, namun tetap mampu memberikan kinerja deteksi yang baik. Pelatihan model dilakukan menggunakan citra yang telah diseragamkan ke ukuran 640×640 piksel dan pelatihan awal sebanyak 50 *epoch* dengan *batch size* 16. Pada penelitian ini, parameter *learning rate* dan *optimizer* tidak ditentukan secara manual, sehingga proses pelatihan memanfaatkan fitur *Auto-Optimizer* bawaan YOLOv8 yang dirancang untuk menghasilkan konfigurasi pelatihan yang lebih optimal dan stabil. Setelah pelatihan tahap pertama selesai, proses pelatihan dilanjutkan kembali selama 50 *epoch* dan 10 *epoch* berikutnya untuk meningkatkan kemampuan model dalam mengenali objek.

Evaluasi model dilakukan menggunakan data validasi yang tidak terlibat adalam proses pelatihan maupun *testing.* Pengujian dilakukan dengan konfigurasi ingerensi bawaan YOLOv8 untuk memperoleh performa deteksi yang optimal. Metrik evaluasi yang digunakan meliputi *precision, recall*, mAP50, mAP50-95. Selain itu, kecepatan inferensi juga dianalisis untuk mengetahui kemampuan model dalam memproses citra. Kemudian dilakukan juga analisis pada setiap kelas makanan untuk mengidentifikasi kategori yang memiliki performa deteksi tinggi maupun rendah, sehingga dapat digunakan sebagai dasar dalam mengevaluasi kemampuan model dalam mengenali berbagai jenis makanan.

3.3 Rancangan Sistem

Antarmuka sistem dikembangkan menggunakan *framework* Flask sebagai media interaksi antara pengguna dan sistem deteksi makanan. Fitur utama yang disediakan adalah pengunggahan gambar melalui pemilihan berkas, yang kemudian diproses oleh *backend* untuk mendeteksi jenis makanan yang terdapat pada citra. Setelah proses deteksi selesai, sistem menampilkan daftar makanan yang teridentifikasi beserta informasi nutrisi masing-masing dalam standar penyajian 100 gram. Selain itu, sistem dilengkapi dengan fitur pemantauan nutrisi harian yang berfungsi mencatat akumulasi asupan nutrisi berdasarkan hasil deteksi makanan. Batas kebutuhan nutrisi harian dapat diatur secara mandiri oleh setiap pengguna sesuai kebutuhannya. Sistem juga menyediakan fitur riwayat untuk menyimpan dan menampilkan hasil deteksi makanan yang dilakukan setiap hari. Rancangan antarmuka sistem yang dikembangkan ditunjukkan pada Gambar 4.

Gambar 4. Rancangan antarmuka

3.4 Integrasi *Application Programming Interface* (API)

Integrasi API dilakukan dengan memanfaatkan layanan FatSecret sebagai sumber data nutrisi makanan. Setelah model YOLOv8 berhasil mendeteksi jenis makanan pada citra, nama makanan yang diperoleh dikirimkan ke FatSecret API melalui mekanisme autentikasi OAuth 2.0 menggunakan *client credentials*. Selanjutnya, sistem melakukan pencarian data berdasarkan nama makanan dan mengambil informasi nutrisi yang tersedia, seperti kalori, protein, karbohidrat, dan lemak per 100 gram. Data yang diterima dari API kemudian diproses dan ditampilkan pada antarmuka pengguna. Untuk meningkatkan efisiensi, sistem menerapkan mekanisme *cache* sehingga data nutrisi yang telah diperoleh sebelumnya tidak perlu diminta kembali ke API. Selain itu, disediakan mekanisme *fallback* untuk menangani kondisi ketika data makanan tidak ditemukan atau koneksi API mengalami kendala, sehingga sistem tetap dapat menampilkan informasi nutrisi kepada pengguna.

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Pelatihan Model

Pelatihan model dilakukan melalui tiga tahapan utama. Tahap pertama adalah *training* awal selama 50 *epoch* untuk membangun kemampuan dasar model dalam mengenali objek makanan berdasarkan dataset yang digunakan. Setelah itu, dilakukan *fine-tuning* selama 50 *epoch* dengan memanfaatkan bobot terbaik hasil pelatihan sebelumnya guna meningkatkan akurasi dan kemampuan generalisasi model. Tahap terakhir berupa *hyperparameter tuning* selama 10 *epoch*, yang bertujuan untuk menyempurnakan performa model melalui penyesuaian beberapa parameter pelatihan sehingga diperoleh hasil deteksi yang

lebih optimal. Gambar 1, Gambar 2, dan Gambar 3 memperlihatkan *source code* yang digunakan pada masing-masing tahapan, yaitu *training* awal, *fine-tuning*, dan *hyperparameter tuning.*

```
# Fine-Tuning dengan Penyesuaian Hyperparameter
model.train(
    data='/content/drive/MyDrive/food-dataset/NewDataset/data.yaml',
    epochs=10,          # Diberi 10 epoch karena pergerakan learning ratenya sekarang sangat halus
    imgsz=640,
    batch=16,
    device=0,
    project='/content/drive/MyDrive/food-dataset/runs',
    name='yolov8_food_model_v2_finetune_hyper',
    exist_ok=True,

    # --- HYPERPARAMETER TUNING ---
    lr0=0.0001,         # Learning rate sangat kecil agar tidak merusak model
    lrf=0.01,           # Final learning rate
    warmup_epochs=0.0,  # Langsung train tanpa warmup
    momentum=0.937,     # Mempertahankan arah gradien
    weight_decay=0.0005, # Mencegah model menghafal jawaban (overfitting)
    mosaic=0.0,         # Matikan gambar terpotong agar model fokus ke makanan utuh
    close_mosaic=0,
    optimizer='SGD'     # SGD lebih halus untuk fine-tuning
)
```

Gambar 5. *Training* awal

```
from ultralytics import YOLO
import os

best_weights = os.path.join(yolo_runs_path, 'yolov8_food_model_v2/weights/best.pt')
if os.path.exists(best_weights):
    print("Loading best weights to continue training...")
    fine_tune_model = YOLO(best_weights)

    # Lanjutkan training untuk 50 epoch lagi.
    # Disimpan di folder baru '_finetune' agar tidak menimpa versi sebelumnya.
    fine_tune_model.train(
        data=yaml_path,
        epochs=50,  # Ubah angka ini jika ingin menambah/mengurangi epoch
        imgsz=640,
        batch=16,
        device=0,
        project=yolo_runs_path,
        name='yolov8_food_model_v2_finetune',
        exist_ok=True,
    )
```

Gambar 6. *Fine-Tuning*

```
# Fine-Tuning dengan Penyesuaian Hyperparameter
model.train(
    data='/content/drive/MyDrive/food-dataset/NewDataset/data.yaml',
    epochs=10,          # Diberi 10 epoch karena pergerakan learning ratenya sekarang sangat halus
    imgsz=640,
    batch=16,
    device=0,
    project='/content/drive/MyDrive/food-dataset/runs',
    name='yolov8_food_model_v2_finetune_hyper',
    exist_ok=True,

    # --- HYPERPARAMETER TUNING ---
    lr0=0.0001,         # Learning rate sangat kecil agar tidak merusak model
    lrf=0.01,           # Final learning rate
    warmup_epochs=0.0,  # Langsung train tanpa warmup
    momentum=0.937,     # Mempertahankan arah gradien
    weight_decay=0.0005, # Mencegah model menghafal jawaban (overfitting)
    mosaic=0.0,         # Matikan gambar terpotong agar model fokus ke makanan utuh
    close_mosaic=0,
    optimizer='SGD'     # SGD lebih halus untuk fine-tuning
)
```

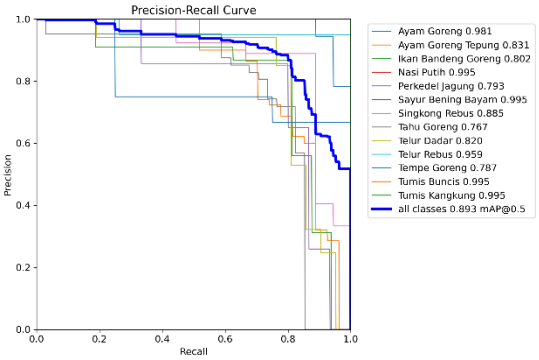
Gambar 7. *Hyperparameter Tuning*

Hasil dari pelatihan model yang dilakukan secara bertahap tersebut secara keseluruhan berhasil mencapai kondisi konvergen, sebagaimana divalidasi melalui tren penurunan *loss* serta kurva evaluasi *PR curve*, *F1-confidence*, dan *confusion matrix*. Performa model terus meningkat secara konsisten, di mana tahap awal menghasilkan *mAP50* sebesar 0,843, kemudian meningkat menjadi 0,883 pada tahap *fine-tuning* (terbaik di *epoch* 45), hingga mencapai titik optimal pada tahap penyesuaian *hyperparameter* (*epoch* 9) dengan nilai *mAP50* 0,893, *precision* 0,858, *recall* 0,870, dan *mAP50-95* 0,658 setelah menerapkan *learning rate* 0,0001, *optimizer* SGD, serta menonaktifkan augmentasi *mosaic*. Berdasarkan evaluasi per kelas, model menunjukkan akurasi sangat tinggi (mAP50 >0,98) pada menu Tumis Buncis, Tumis Kangkung, Sayur Bening Bayam, Nasi Putih, dan Ayam Goreng, meskipun tantangan deteksi akibat kemiripan visual tekstur masih ditemukan pada Ayam Goreng Tepung, Tahu Goreng, dan Perkedel Jagung.

4.2 Hasil Pengujian Model

1. *Precision-Recall Curve*

Evaluasi model pada data uji menunjukkan hasil deteksi yang baik dengan nilai mAP50 sebesar 89,3% untuk seluruh kelas makanan. Nilai tersebut menunjukkan bahwa model mampu mengenali dan membedakan objek makanan dengan tingkat akurasi yang tinggi. Berdasarkan hasil pengujian, beberapa kelas memperoleh performa yang sangat baik, di antaranya Nasi Putih (0,995), Sayur Bening Bayam (0,995), Tumis Buncis (0,995), Tumis Kangkung (0,995), Ayam Goreng (0,981), dan Telur Rebus (0,959). Sementara itu, kelas seperti Ayam Goreng Tepung (0,831), Ikan Bandeng Goreng (0,802), Perkedel Jagung (0,793), Tahu Goreng (0,767), dan Tempe Goreng (0,787) menunjukkan performa yang relatif lebih rendah, meskipun masih berada pada kategori yang cukup baik. Gambar 5 menampilkan kurva *Precision-Recall* (PR *Curve*) untuk seluruh kelas, Kurva pada sebagian besar kelas berada di area atas grafik dan membentuk luas area yang cukup besar, yang menandakan bahwa model memiliki keseimbangan yang baik antara nilai *precision* dan *recall*.

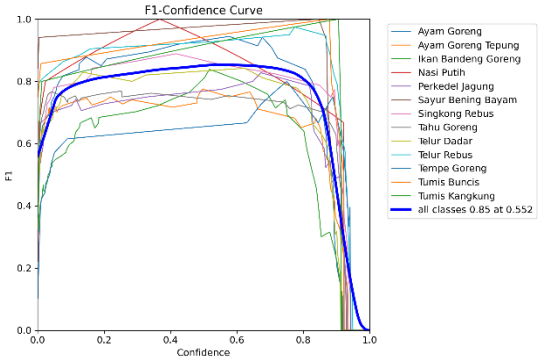


Gambar 8. Grafik *Precision-Recall*

2. F1-*confidence Curve*

Sementara itu, hasil evaluasi berdasarkan metrik F1-*Score* menunjukkan performa yang baik dengan nilai F1-*score* sebesar 0,85 pada *confidence threshold* 0,552 untuk seluruh kelas

makanan. Nilai tersebut menunjukkan bahwa model mampu mencapai keseimbangan yang optimal antara *precision* dan *recall*, sehingga prediksi yang dihasilkan tidak hanya tepat tetapi juga mampu mendeteksi sebagian besar objek yang terdapat pada citra. Hasil ini mengindikasikan bahwa model telah memiliki kemampuan yang baik dalam mengenali berbagai jenis makanan pada dataset yang digunakan. Gambar 9. menampilkan grafik F1-*Confidence* untuk seluruh kelas makanan.

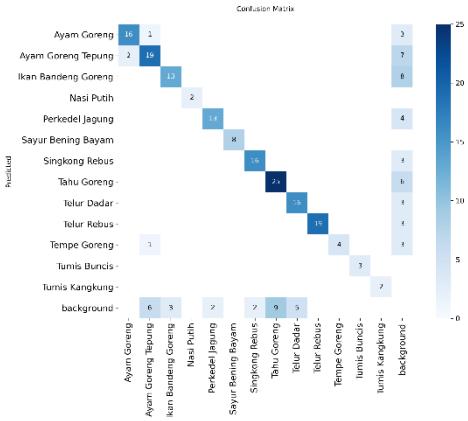


Gambar 9. Grafik F1-*Confidence*

Berdasarkan gambar tersebut, dapat dilihat bahwa Sebagian besar kelas menunjukkan nilai F1 yang relatif tinggi dan stabil pada rentang *confidence* menengah hingga tinggi. Beberapa kelas seperti Nasi Putih, Sayur Bening Bayam, Tumis Buncis, Tumis Kangkung, Ayam Goreng, dan Telur Rebus memiliki kurva yang berada pada area atas grafik, yang menunjukkan performa deteksi yang sangat baik. Sementara itu, kelas seperti Tahu Goreng, Perkedel Jagung, Tempe Goreng, dan Ikan Bandeng Goreng memiliki nilai F1 yang relatif lebih rendah dibandingkan kelas lainnya, meskipun masih menunjukkan kemampuan deteksi yang cukup baik. Selain itu, garis F1 untuk keseluruhan kelas menunjukkan performa yang stabil dengan nilai puncak 0,85 pada *confidence* 0,552. Setelah melewati titik tersebut, nilai F1 cenderung menurun karena model menjadi lebih selektif dalam menerima prediksi sehingga jumlah objek yang berhasil terdeteksi berkurang. Secara keseluruhan, kurva F1-*Confidence* menunjukkan bahwa model telah mampu menghasilkan keseimbangan yang baik antara *precision* dan *recall.*

3. *Confusion Matrix*

Hasil *confusion matrix* menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengenali setiap kelas makanan. Kelas Tahu Goreng memperoleh jumlah prediksi benar tertinggi sebanyak 25 objek, sementara Telur Rebus dan Ayam Goreng Tepung juga menunjukkan hasil deteksi yang baik. Meskipun demikian, masih terdapat beberapa kesalahan klasifikasi, seperti Ikan Bandeng Goreng yang terdeteksi sebagai Ayam Goreng Tepung, yang kemungkinan disebabkan oleh kemiripan bentuk dan tekstur antar objek serta proses pelabelan yang kurang presisi. Selain itu, beberapa objek makanan masih terdeteksi sebagai *background*, hal tersebut dapat dipengaruhi oleh faktor pencahayaan, posisi objek pada citra, maupun kualitas anotasi data. Untuk meningkatkan performa model, dapat dilakukan penambahan variasi dataset, pembuatan anotasi yang lebih akurat, serta penggunaan citra dengan sudut pengambilan yang lebih beragam. Secara keseluruhan, hasil *confusion matrix* menunjukkan bahwa model telah mampu mendeteksi dan membedakan berbagai jenis makanan dengan tingkat ketepatan yang cukup baik. Gambar 10 menampilkan hasil *confusion matrix.*



Gambar 10. *Confusion Matrix*

4.3 Implementasi Sistem

Setelah proses pengembangan *backend* dan integrasi API FatSecret untuk menampilkan informasi nutrisi berhasil dilakukan, tahap selanjutnya adalah membangun antarmuka pengguna sebagai penghubung antara pengguna dan sistem. Antarmuka ini juga digunakan untuk melakukan pengujian terhadap sistem yang telah dikembangkan. Berikut merupakan tampilan antarmuka yang diimplementasikan.

1. Halaman Deteksi

Gambar 11. Tampilan halaman deteksi

Pada halaman deteksi, pengguna dapat mengunggah gambar makanan sebagai masukan (*input*) yang kemudian akan dianalisis oleh sistem untuk mengidentifikasi jenis makanan yang terdeteksi.

2. Halaman Hasil Deteksi

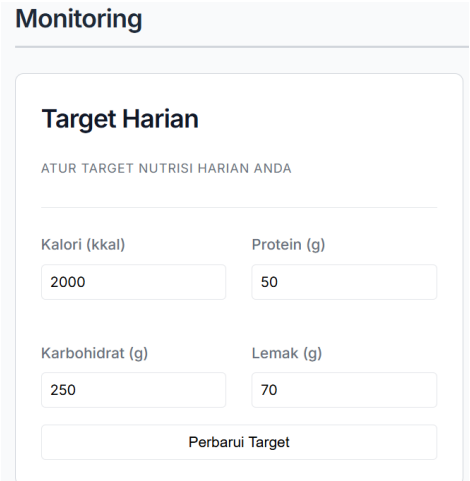
Gambar 12. Tampilan hasil deteksi

Setelah makanan berhasil dideteksi, sistem akan mengakses API FatSecret untuk memperoleh informasi nutrisi berdasarkan jenis makanan yang telah teridentifikasi. Pada Gambar 12, terlihat bahwa model berhasil mendeteksi seluruh objek makanan yang terdapat pada citra dan menampilkan informasi nutrisi dari masing-masing makanan sesuai dengan hasil deteksi yang diperoleh.

3. Halaman Monitoring Harian

Gambar 13. Tampilan monitoring nutrisi

Seluruh informasi nutrisi dari makanan yang berhasil dideteksi akan secara otomatis dicatat ke dalam fitur monitoring nutrisi harian. Fitur ini berfungsi untuk memantau jumlah nutrisi yang telah dikonsumsi dan membandingkannya dengan kebutuhan nutrisi harian pengguna. Gambar 14 menampilkan halaman pengaturan batas atau target nutrisi harian yang dapat disesuaikan oleh setiap pengguna sesuai kebutuhannya dalam satu hari.



Gambar 14. Tampilan set limit nutrisi

4. Riwayat Hasil Deteksi

Gambar 15. Riwayat hasil deteksi

Setiap makanan yang berhasil terdeteksi juga akan otomatis disimpan ke dalam riwayat deteksi beserta informasi nutrisinya. Fitur ini memungkinkan pengguna untuk melihat kembali jenis makanan yang telah dikonsumsi serta memantau asupan nutrisi harian secara lebih mudah dan terorganisir.

5. KESIMPULAN

Berdasarkan hasil yang diperoleh, dapat disimpulkan beberapa poin sebagai berikut:

a. Dengan menggunakan algoritma YOLOv8n, penelitian ini berhasil menghasilkan sistem yang mampu mendeteksi jenis makanan dari citra dan menyajikan informasi nutrisi

secara otomatis. Model dilatih menggunakan 2.000 citra, yang kemudian meningkat menjadi 5.374 citra setelah augmentasi, serta mencapai nilai mAP@0.5 sebesar 89,3% pada data uji.

b. Model mampu mengenali sebagian besar jenis makanan dengan baik. Beberapa kelas yang memperoleh performa tinggi antara lain Ayam Goreng (0,981), Sayur Bening Bayam (0,995), Telur Rebus (0,959), dan Tumis Buncis (0,995). Meskipun demikian, beberapa kelas masih menunjukkan performa yang lebih rendah akibat faktor seperti keterbatasan jumlah data, variasi pencahayaan, kualitas citra, serta kemiripan visual antar objek makanan.
c. Sistem berhasil mengintegrasikan layanan FatSecret API untuk memperoleh dan menampilkan informasi nutrisi berdasarkan jenis makanan yang berhasil dideteksi oleh model.
d. Sistem berhasil melakukan monitoring nutrisi harian dengan mengakumulasi informasi nutrisi dari setiap makanan yang terdeteksi, sehingga pengguna dapat memantau asupan nutrisi yang telah dikonsumsi dalam satu hari.
e. Secara keseluruhan, sistem yang dikembangkan telah memenuhi tujuan penelitian, yaitu mampu mendeteksi jenis makanan, menampilkan informasi nutrisi, serta melakukan pemantauan asupan nutrisi harian pengguna.
f. Penelitian selanjutnya disarankan untuk memperkaya jumlah dan keragaman dataset, terutama citra yang memuat beberapa jenis makanan dalam satu piring, serta menambahkan variasi sudut pengambilan gambar, pencahayaan, dan latar belakang agar model lebih mampu beradaptasi dengan kondisi nyata. Selain itu, pengembangan metode estimasi porsi makanan dan penerapan model deteksi atau segmentasi yang lebih canggih dapat dilakukan untuk meningkatkan akurasi perhitungan nutrisi dan kemampuan deteksi pada objek makanan yang saling berdekatan atau tumpang tindih.

UCAPAN TERIMA KASIH

Penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah memberikan dukungan, bantuan, serta kontribusi selama pelaksanaan penelitian ini. Berbagai bentuk arahan, masukan, dan kerja sama yang diberikan memiliki peran penting dalam mendukung kelancaran setiap tahapan penelitian hingga penyusunan laporan ini dapat diselesaikan dengan baik dan tepat waktu.

DAFTAR PUSTAKA

[1] G. C. Utami, C. R. Widiawati, and P. Subarkah, “Detection of Indonesian Food to Estimate Nutritional Information Using YOLOv5,” *Teknika*, vol. 12, no. 2, pp. 158–165, Jun. 2023, doi: 10.34148/teknika.v12i2.636.

[2] N. K. Hamzidah *et al.*, “Performance Evaluation of Food Calorie Counter Mobile Application Based on CNN-YOLO Algorithms,” *Jambura Journal of Electrical and Electronics Engineering*, vol. 7, no. 2, pp. 253–263, 2025.

[3] R. Supriyadi *et al.*, “PENGEMBANGAN APLIKASI ESTIMASI KALORI MAKANAN BERBASIS CITRA DENGAN PENDEKATAN DETEKSI OBJEK MENGGUNAKAN YOLO,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 14, no. 1, pp. 1393–1404, Jan. 2026, doi: 10.23960/jitet.v14i1.8545.

[4] H. A. Pratiwi, M. Cahyanti, and M. Lamsani, “IMPLEMENTASI DEEP LEARNING FLOWER SCANNER MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK,” *Sebatik*, vol. 25, no. 1, pp. 124–130, Jun. 2021, doi: 10.46984/sebatik.v25i1.1297.

[5] Y. K. Bintang and H. Imaduddin, “PENGEMBANGAN MODEL DEEP LEARNING UNTUK DETEKSI RETINOPATI DIABETIK MENGGUNAKAN METODE TRANSFER LEARNING,” *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 3, pp. 1442–1455, Aug. 2024, doi: 10.29100/jipi.v9i3.5588.

[6] F. M. Syakir and M. Syani, “APLIKASI TAMU WAJIB LAPOR BERBASIS MOBILE (STUDI KASUS KP. PASIR PEUNDEUY CIHAMPELAS BANDUNG BARAT),” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, pp. 3867–3873, Oct. 2024, doi: 10.23960/jitet.v12i3s1.5211.

[7] W. Djurianto, E. Maulana, and C. Brilianto, “Aplikasi Pendeteksi Objek Kendaraan Menggunakan Intel Movidius Neural Compute Stick dengan Algoritma MobileNet-YOLO,” *Jurnal EECCIS2021*, vol. 15, no. 3, pp. 92–97,

2021, [Online]. Available: https://jurnaleeccis.ub.ac.id/

[8] I. M. D. Maleh, R. Teguh, A. S. Sahay, S. Okta, and M. P. Pratama, "Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan Di Taman Nasional Sebangau," *Jurnal Informatika*, vol. 10, no. 1, pp. 19–27, Mar. 2023, doi: 10.31294/inf.v10i1.13922.

[9] I. Inayatul Arifah, F. Nur Fajri, and G. Qorik Oktagalu Pratamasunu, "Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN," *Journal of Applied Informatics and Computing (JAIC)*, vol. 6, no. 2, pp. 2548–6861, 2022, [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[10] Khairunnas, Yuniarno Mulyanto E, and Zaini Ahmad, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *JURNAL TEKNIK ITS*, vol. 10, no. 1, pp. 50–55, 2021.

[11] F. Alfaridzi, J. Dedy Irawan, and M. Orisa, "PERANCANGAN SISTEM MANAJEMEN USER HOTSPOT BERBASIS WEB MENGGUNAKAN APPLICATION PROGRAMMING INTERFACE (API) MIKROTIK," *Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 974–981, 2022.

[12] A. Triawan, A. Ramot, and Y. Siboro, "Penerapan Application Programming Interface (API) Pada Push Notification Untuk Informasi Monitoring Stok Barang Minim," *Jurnal Ilmiah Teknologi-Informasi&Sains*, vol. 11, no. 2, pp. 107–114, 2021, doi: 10.36350/jbs.v11i2.

[13] H. Y. Hermansyah and M. Maryam, "IMPLEMENTASI TEKNOLOGI APPLICATION PROGRAMMING INTERFACE PADA PERANCANGAN APLIKASI ABSENSI PEGAWAI," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 3, pp. 744–754, Aug. 2023, doi: 10.29100/jipi.v8i3.3890.

[14] D. B. Avatara, Y. Sholva, and R. D. Nyoto, "Backend Service untuk Otomatisasi Validasi Format Pas Foto dengan Pengolahan Citra menggunakan Library OpenCV," *Jurnal Ilmiah Sains dan Teknologi*, vol. 4, no. 2, pp. 406–420, 2026.

[15] R. K. Ngantung and M. A. I. Pakereng, "Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, pp. 1052–1062, Jul. 2021, doi: 10.30865/mib.v5i3.3054.

[16] Y. Tamariska Bota and N. Setiyawati, "Pengembangan Sistem Informasi Perantara Bisnis Menggunakan Framework Flask," *Journal of Information Technology Ampera*, vol. 3, no. 2, pp. 2774–2121, 2022, [Online]. Available: https://journal-computing.org/index.php/journal-ita/index

[17] R. Andarsyah and A. Yanuar, "SENTIMEN ANALISIS APLIKASI POSAJA PADA GOOGLE PLAYSTORE UNTUK PENINGKATAN POSPAY SUPERAPP MENGGUNAKAN SUPPORT VECTOR MEACHINE," *Jurnal Teknik Informatika*, vol. 16, no. 2, pp. 1–7, 2024.