Perancangan Dan Analisis Model *Object Recognition* Untuk Mengenali Makanan Pada Aplikasi Foodit Dengan Menggunakan Yolov8

1st Sheerlyn Eliza Febriyana
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
Sheerlynnn@student.telkomuniversity.ac.id

2nd Suryo Adhi Wibowo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
suryoadhiwibowo@telkomuniversity.ac.id

3rd Korediantо Usman
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
korediantousman@telkomuniversity.ac.id

***Abstrak* — Kesehatan tubuh sangat penting bagi kehidupan seseorang. Menjaga pola makan yang seimbang dan terpenuhinya nutrisi gizi adalah faktor utama dalam menjaga kesehatan tubuh. Maka dari itu untuk mencegah resiko kurangnya gizi yang dikonsumsi, pemerintah menetapkan Angka Kecukupan Gizi (AKG) bagi masyarakat di Indonesia melalui Peraturan Menteri Kesehatan (PMK) No.28 Tahun 2019 tentang Angka Kecukupan Gizi yang Dianjurkan bagi Masyarakat Indonesia. Untuk mengatasi masalah tersebut, pada *Capstone Design* ini tim penulis membuat sebuah aplikasi Foodit yang berbasis Android. Aplikasi ini dirancang untuk membantu pengguna dalam menghitung dan mengkontrol kebutuhan zat gizi makro berdasarkan ketetapan AMDR perharinya. Kelebihan dari aplikasi Foodit adalah dirancang dengan menggunakan metode *deep learning,* yaitu *object recognition.* Dimana aplikasi ini dapat mengenali objek makanan apa saja yang terdapat pada gambar, saat meng-*input* makanan secara otomatis dengan melakukan pengambilan gambar melalui kamera *smartphone.* Penulis merancang model *object recognition* dengan algoritma YOLOv8. Penulis melakukan konfigurasi nilai *hyperparameter* untuk mendapatkan nilai mAP yang terbaik. Dari hasil konfigurasi tersebut didapatkan nilai mAP sebesar 0.995 dengan *learning rate* sebesar 0.001 dan *batch size* sebesar 8.**

***Kata kunci*—aplikasi, gizi, object recognition, YOLOv8.**

A. PENDAHULUAN

Kesehatan tubuh sangat penting bagi kehidupan seseorang karena segala aktivitas hanya dapat dilakukan ketika tubuh sehat. Menjaga pola makan yang seimbang dan terpenuhinya nutrisi gizi adalah faktor utama dalam menjaga kesehatan tubuh. Dengan menjaga pola makan yang seimbang akan menyebabkan nutrisi gizi yang ada di dalam tubuh menjadi seimbang. Pada orang dewasa, kekurangan nutrisi dapat meningkatkan risiko obesitas, hipertensi, penyakit jantung, stroke, diabetes, dan penyakit tidak menular (PTM) lainnya [1]. Maka dari itu untuk mencegah resiko kurangnya gizi yang dikonsumsi, pemerintah menetapkan Angka Kecukupan Gizi (AKG) bagi masyarakat di Indonesia melalui Peraturan Menteri Kesehatan (PMK) No.28 Tahun 2019 tentang Angka Kecukupan Gizi yang Dianjurkan bagi Masyarakat Indonesia [2].

Untuk mengatasi masalah tersebut, pada *Capstone Design* ini tim penulis membuat sebuah aplikasi Foodit yang berbasis Android. Aplikasi ini dirancang untuk membantu pengguna dalam menghitung dan mengkontrol kebutuhan zat gizi makro berdasarkan ketetapan AMDR perharinya. Pada aplikasi Foodit terdapat fitur dimana pengguna dapat meng-*input* makanan yang dikonsumsi dalam sehari (pagi, siang, dan malam). Setelah meng-*input* makanan, aplikasi akan menginformasikan kandungan karbohidrat, protein, dan lemak yang terkandung dalam makanan tersebut. Jika informasi yang diberikan sudah sesuai, maka pengguna dapat mengsubmit makanan tersebut sehingga aplikasi akan mengkalkulasi asupan gizi yang telah dikonsumsi pada hari tersebut. Setelah kalkukasi karbohidrat, protein, dan lemak, aplikasi akan memberikan peringatan jika pengguna mengkonsumsi makanan yang tidak sesuai dengan kebutuhannya. Pada aplikasi Foodit juga terdapat fitur riwayat, dimana pengguna dapat melihat kembali makanan sudah dimakan sebelumnya.

Pada fitur meng-*input* makanan terdapat 2 cara, yaitu secara manual dengan memasukkan nama makanan pada menu pencarian atau dengan pengambilan gambar melalui kamera *smartphone* selanjutkan aplikasi akan mengenali *object* makanan apa saja yang terdapat pada gambar tersebut. Fitur pengenalan makanan ini memanfaatkan model *object recognition* dan menggunakan algoritma YOLOv8 untuk membuat model *object recognition.* Untuk mengetahui tingkat akurasi model *object recognition* dalam mengenali *object* makanan, dapat dilihat berdasarkan nilai mAP. Nilai tertinggi mAP adalah 1,00 yang menandakan bahwa model dapat mengenali seluruh *object* dengan baik.

B. KAJIAN TEORI

Pada perancangan aplikasi Foodit ini tim penulis membaginya menjadi 4 subsistem, yaitu *machine learning, deep learning, android development,* dan *cloud computing*. Pada jurnal ini penulis akan membahas mengenai sub sistem *deep learning.* Terdapat beberapa teori yang mendasari

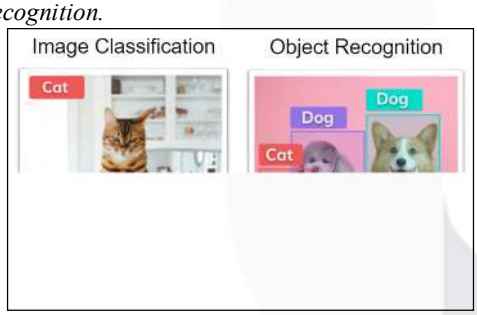
perancangan sub sistem *deep learning* ini, yaitu diantaranya *deep learning, object recognition,* YOLOv8, dan *hyperparameter*. Untuk penjelasan lebih lengkapkan sebagai berikut:

Deep Learning

Deep learning adalah bagian dari *machine learning* yang menungkinkan komputer untuk belajar berdasarkan pengalaman dan pemahaman dunia dengan konsep hierarki [3]. Selain itu *deep learning* merupakan subbidang *machine learning* yang menggunakan algoritma *Artificial Neural Network* (ANN), dimana algoritma ini terinspirasi dari stuktur otak manusia [4]. *Deep learning* membuat komputer belajar tanpa pengawasan manusia dan memberikan kemampuan untuk mengenali ucapan, menerjemahkan bahasa, mendeteksi objek, dan bahkan membuat keputusan berdasarkan data [5].

Object Recognition

Object recognition adalah bagian dari *artificial intelligence* yang dapat mengekstaksi informasi yang diperlukan atau wawasan kritis dari suatu gambar atau video [6]. *Object recognition* digunakan untuk membantu komputer untuk melihat gambar yang ada dan memecahnya menjadi beberapa piksel untuk mengenali pola atau bentuk tertentu [6]. Keberhasilan suatu model *oject recognition* bergantung pada kualitas data yang digunakan untuk melatih model tersebut. Lebih banyak data yang dilatih maka model akan lebih cepat mengenali dan mengklasifikasikan objek berdasarkan karakteristik yang sudah dikelahui [6]. Gambar 1merupakan perbedaan antara *image classification* dengan *object recognition.*

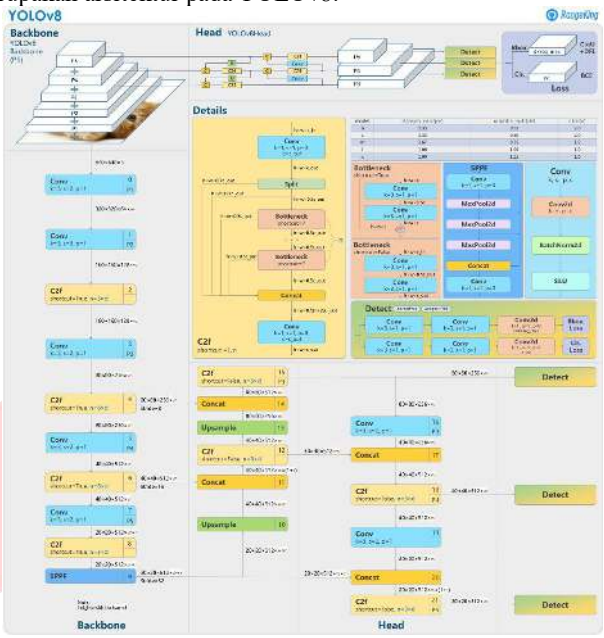


GAMBAR 1
Perbedaan Image Classification Dengan Object Recognition

Berdasarkan gambar diatas, kita dapat mengetahui pada *image classification* hanya dapat mengenali *single object* pada gambar atau video secara keseluruhan, sedangkan pada *object recognition* dapat mengidentifikasi *multiple object* yang ditandai dengan label yang telah ditentukan. Maka penulis menggunakan *object recognition* diimplementasikan pada aplikasi Foodit.

YOLOv8

YOLOv8 *Ultralytics* adalah model *State Of The Art* (SOTA) canggih yang dibangun setelah kesuksesan versi YOLO sebelumnya dan memperkenalkan fitur dan peningkatan baru yang lebih meningkatkan kinerja dan fleksibilitas [7]. YOLOv8 dirancang agar dapat cepat, akurat, dan mudah digunakan sehingga menjadi pilihan yang sangat tepat untuk berbagai deteksi dan pelacakan objek, segmentasi instan, klasifikasi gambar, dan tugas estimasi pose [7]. Gambar 2 merupakan arsitektur pada YOLOv8.



GAMBAR 2
ARSITEKTUR YOLOv8

Dari arsitektur tersebut terdapat beberapa komponen utama pada YOLOv8, yaitu:

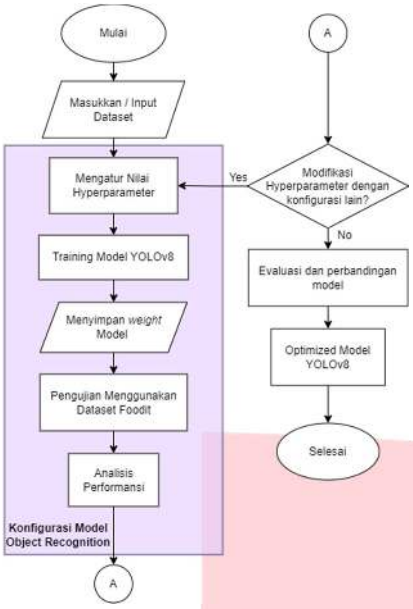
a. *Backbone*: YOLOv8 menggunakan ResNet-152 sebagai *backbone.*
b. *Neck*: YOLOv8 menggunakan *Feature Pyramid Network* (FPN).
c. *Head*: Menggunakan *head* yang terdiri dari beberapa *layer* konvolusi dan *layer* untuk memproses fitur dari *neck* serta menghasilkan *bounding boxes* dan kelas objek.
d. *Output*: *Output*-nya akan berupa koordinat *bounding box*, skor kepercayaan, dan kelas objek yang terdeteksi.

Hyperparameter

Hyperparameter berfungsi untuk melakukan konfigurasi pada model *machine learning* dan menentukan algoritma yang digunakan untuk memperbesar nilai mAP [8]. *Hyperparameter tuning* adalah proses untuk merancang model *machine learning* yang sempurna dengan konfigurasi *hyperparameter* yang ideal [9]. Parameter yang dapat diubah pada model YOLOv8 untuk meningkatkan kinerja model, yaitu: *epochs*, ukuran gambar input, *batch size*, *learning rate*, momentum, dan *optimizer*. Parameter-parameter ini ditetapkan sebelum dilakukan proses *training.*

III. METODE

Pada pembuatan model *object recognition* penulis melakukan beberapa metode yang dilakukan untuk mendapatkan hasil yang optimal. Gambar 3 adalah diagram alir *deep learning* untuk mendapatkan model *object recognition* yang terbaik. Metode yang dilakukan diantaranya yaitu mempersiapkan dan memasukkan dataset, melakukan konfigurasi model *object recognition,* evaluasi model dan diakhiri dengan menyimpan hasil *output* dalam best.pt file.



GAMBAR 3
Diagram Alir Deep Learning

A. Dataset

Dataset adalah sekumpulan data yang disediakan secara terstruktur dan nantinya akan digunakan untuk diuji dan dianalisa pada suatu sistem. Pada *Capstone Design* ini, *dataset* digunakan untuk melatih model sehingga dapat mengenali objek tertentu. *Dataset* yang digunakan yaitu *dataset* Foodit yang terdiri dari 10 kelas makanan, diantaranya adalah nasi putih, roti putih, mie ayam, mie bakso, telur balado, cah kangkung, tempe goreng, *chicken nugget*, dan sayap ayam goreng *crispy*. *Dataset* tersebut dibuat oleh penulis dengan pengambilan gambar untuk setiap kelas makanan dan melakukan anotasi pada Roboflow. Pada *dataset* Foodit juga tim penulis melakukan proses augmentasi, dimana proses augmentasi ini bertujuan untuk menambah variasi dari *dataset* yang sudah dianotasi sebelumnya. *Dataset* Foodit dibagi menjadi tiga kelompok, yaitu *train, valid*, dan *test* data. Data *train* digunakan saat proses *training* pada algoritma *Object recognition*. Algoritma yang telah dilatih selanjutnya akan diuji dengan data *valid* untuk memastikan kinerja sementara, dan jika kinerjanya tidak sesuai dengan target, *training* model dapat diulang untuk mencapai kinerja yang diinginkan. Data *test* digunakan untuk mengetahui akurasi akhir pada model yang sudah dilatih. Tabel 1 menunjukkan spesifikasi *dataset* yang digunakan pada *Capstone Design* ini.

TABEL 1
SPESIFIKASI DATASET

Kelompok	Jumlah Citra	Jumlah Label / Anotasi
Train	4.626	7.100
Valid	449	695
Test	226	367

B. Konfigurasi Object Recognition

Pada perancangan model *object recognition* penulis melakukan konfigurasi berdasarkan nilai *hyperparameter.* Konfigurasi ini dilakukan untuk membandingkan dan mendapatkan model *object recognition* yang terbaik. Konfigurasi dilakukan pada nilai *hyperpameter learning rate* dan *batch size,* sehingga dirancang sebanyak 3 skema model yang nantinya akan dibandingnya hasilnya setiap masing-masing model skema. Model skema tersebut diantaranya YOLOv8 orisinal, skema I, dan skema II. Rincian skema konfigurasi *hyperparameter* yang terdapat pada *Capstone Design* ini adalah sebagai berikut:

A. YOLOv8 orisinal: *Train* model YOLOv8 tanpa perubahan nilai *hyperparameter* pada modelnya. Rancangan ini menjadi acuan peningkatan performansi bagi rancangan lainnya.
B. Skema I: *Train* model YOLOv8 dengan merubah nilai *hyperparameter* berupa *learning rate* sebesar 0,001. Dimana *learning rate* berfungsi untuk mengontrol seberapa cepat atau lambat model belajar dari *dataset* yang diberikan selama proses training [10].
C. Skema II: Train model YOLOv8 dengan merubah hyperparameter berupa learning rate sebesar 0.001 dan batch size sebesar 8. Dimana batch size berfungsi untuk mengatur jumlah sampel yang diproses dalam satu epoch pada proses training [10].

Evaluasi Model Object Recognition

Dalam *capstone design* ini penulis akan mengggunakan parameter evaluasi seperti *precision*, *recall*, dan *mean Average Precision* (mAP). Evaluasi ini dilakukan untuk mengetahui seberapa optimal model dalam mengenali objek makanan. Untuk penjelasan lebih lengkap terkait parameter evaluasi tersebut sebagai berikut:

Precision adalah kemampuan model untuk mengidentifikasi objek yang relevan saja. Untuk mendapatkan nilai *precision* dapat dihitung dengan menggunakan persamaan 1.

$$P = \frac{TP}{TP+FP}, \quad (1)$$

dimana TP merupakan *True Positive* dan FP merupakan *False Positive.*

Recall adalah kemampuan model untuk menemukan semua kasus yang relevan (semua *ground-truth bounding boxes*). Untuk mendapatkan nilai *recall* dapat dihitung dengan menggunakan persamaan 2.

$$R = \frac{TP}{TP+FN}, \quad (2)$$

dimana TP merupakan *True Positive* dan FN merupakan *False Negative.*

Mean Average Precision (mAP) adalah metrik yang digunakan untuk mengukur keakuratan *object recognition* di semua kelas dalam *database* tertentu. mAP juga dapat dikatakan sebagai rata-rata dari nilai AP dari semua kelas dalam *dataset*. Secara matematis, mAP dapat dihitung menggunakan persamaan 3.

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i, \quad (3)$$

dimana variabel C merupakan jumlah kelas yang terdapat pada *dataset*. Sedangkan variabel i merupakan perulangan yang dimulai dari satu hingga jumlah kelas pada *dataset* atau C. AP (*Average Precision)* adalah nilai rata-rata *precision* pada tingkat *recall* yang berbeda-beda. Untuk

mendapatkan nilai AP dapat menghitung dengan menggunakan persamaan 4.

$$AP = \sum_n (R_n - R_{n-1}) P_n, \quad (4)$$

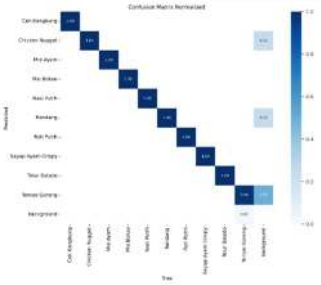
dimana R_n dan P_n adalah *recall* dan *precision* pada *threshold* ke-n.

IV. HASIL DAN ANALISIS

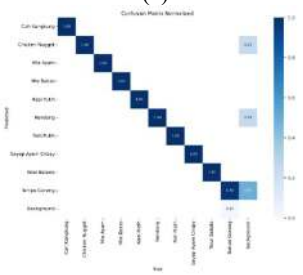
Setelah melakukan proses *training* model *object recognition* dari masing-masing model skema, maka selanjutnya akan dilakukan analisis model *object recognition* untuk melihat apakah pada saat proses *training* model dapat mengenali *dataset* dengan optimal atau tidak. Untuk penjelasan lebih lengkapnya dapat dilihat sebagai berikut:

A. Hasil Model *Object Recognition*

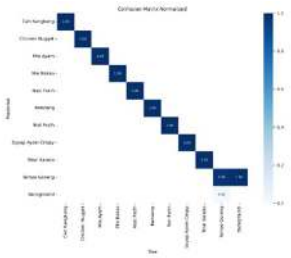
Hasil dari masing-masing model skema dapat dilihat pada *confusion matrix normalized* untuk mengetahui apakah model dapat mengenal objek makanan sesuai dengan kelas yang telah ditentukan. Semakin mendekati angka 1 makanya model dapat mengenali objek makanan dengan baik. Gambar 4 adalah *confusion matrix normalized* dari masing-masing model skema. Dari gambar tersebut, dapat dilihat ketiga model tersebut untuk kelas cah kangkung, mie ayam, mie bakso, nasi putih, roti putih, sayap ayam *crispy*, dan telur balado mendapatkan nilai 1.00, hal ini menunjukkan bahwa model dapat mengenali 100% seluruh *object* berdasarkan kelasnya dengan baik. Pada model YOLOv8 dan model skema 1 mengalami kesalahan dalam mengenali *object,* dimana untuk kelas *chicken nugget*, rendang, dan tempe goreng model mengenal *background* sebagai kelas tersebut. Sedangkan pada skema 2 model hanya mengalami kesalahan pada kelas tempe goreng, dimana model mengenal *background* sebagai tempe goreng.



(a)



(b)



(c)

GAMBAR 4
VISUALISASI CONFUSION MATRIX NORMALIZED PADA MODEL (A) YOLOV8 ORISINAL, (B) SKEMA 1, DAN (C) SKEMA II

B. Analisis Model *Object Recognition*

Analisis konfigurasi *hyperparameter* pada model *object recognition* ini dilakukan untuk membandingkan hasil skema *hyperparameter* dan mengetahui skema *hyperparameter* yang dapat menghasilkan mAP tertinggi. Seperti yang telah dijelaskan pada dokumen sebelumnya, skema yang akan diuji terdiri dari YOLOv8 orisinal, skema I, dan skema II. Melalui pengujian ini, tim penulis dapat menentukan model yang akan dipilih untuk di-*deploy* pada *server* atau *cloud.*

GAMBAR 5
Perbandingan Nilai Map

TABEL 2
Hasil Precision, Recall Dan Map 0.5

Konfigurasi	Kelas	Precision	Recall	mAP 0.5
YOLOv8 Orisinal	**All**	**0.996**	**0.993**	**0.994**
	Cah kangkung	0.998	1	0.995
	Chicken nugget	0.998	0.977	0.995
	Mie ayam	0.998	1	0.995
	Mie bakso	0.997	1	0.995
	Nasi putih	0.999	1	0.995
	Rendang	0.987	1	0.995
	Roti putih	0.998	1	0.995
	Sayap ayam crispy	0.999	1	0.995
	Telur balado	1	0.996	0.995
	Tempe goreng	0.989	0.961	0.987
Skema 1	**All**	**0.996**	**0.993**	**0.994**
	Cah kangkung	0.998	1	0.995
	Chicken nugget	0.998	0.977	0.995
	Mie ayam	0.998	1	0.995
	Mie bakso	0.997	1	0.995
	Nasi putih	0.999	1	0.995
	Rendang	0.987	1	0.995
	Roti putih	0.998	1	0.995
	Sayap ayam crispy	0.99	1	0.995

Konfigurasi	Kelas	Precision	Recall	mAP 0.5
	Telur balado	1	0.996	0.995
	Tempe goreng	0.989	0.961	0.987
Skema 2	**All**	**0.998**	**0.995**	**0.995**
	Cah kangkung	1	1	0.995
	Chicken nugget	1	0.98	0.995
	Mie ayam	1	1	0.995
	Mie bakso	0.997	1	0.995
	Nasi putih	0.999	1	0.995
	Rendang	0.999	1	0.995
	Roti putih	0.997	1	0.995
	Sayap ayam crispy	0.999	1	0.995
	Telur balado	1	0.996	0.995
	Tempe goreng	0.986	0.978	0.993

Dari hasil pengujian konfigurasi *hyperparameter* pada model *object recognition* yang telah dilakukan maka dapat dilihat hasil mAP pada Gambar 5 dan Tabel 2. Nilai mAP model didapatkan dari hasil rata-rata mAP masing-masing kelas, dimana nilai mAP tiap kelas didapatkan dari hasil perhitungan nilai *precision* dan *recall.* Untuk hasil mAP dari masing-masing kelas dapat dilihat pada Tabel 2. Maka pada skema YOLOv8 orisinal dan skema I dapat dilihat pada grafik mAP menunjukan 0.994. Sedangkan pada skema II mendapat nilai mAP sebesar 0.995.

V. KESIMPULAN

Perancangan dan analisis model *object recognition* pada aplikasi Foodit dilakukan dengan konfigurasi nilai *hyperparameter*. Nilai *hyperparameter* yang diubah adalah nilai *learning rate* dan *batch size* pada algoritma YOLOv8. Dimana nilai *learning rate* diubah dari 0,01 menjadi 0,001, sedangkan nilai *batch size* diubah dari 16 menajdi 8. Maka terdapat 3 model skema dari hasil konfigurasi nilai *hyperparameter,* yaitu YOLOv8 orisinal, skema I, dan skema II. Dari hasil dan analisis didapatkan pada skema YOLOv8 orisinal dan skema I dapat dilihat pada grafik mAP menunjukan 0.994. Sedangkan pada skema II mendapat nilai mAP sebesar 0.995. Dapat dilihat bahwa nilai mAP yang paling tinggi adalah model skema II, dengan nilai dari *hyperparameter learning rate* sebesar 0,001 dan *batch size* sebesar 8. Nilai *learning rate* bertujuan untuk mengontrol seberapa cepat atau lambat model belajar dari dataset yang diberikan selama proses *training*, pada konfigurasi ini dapat dilihat bahwa nilai mAP dari skema II dengan *learning rate* 0,001 lebih tinggi dibandingkan dengan YOLOv8 Orisinal dengan *learning rate* 0,01 [10]. Hal ini disebabkan karena, model skema II mempelajari *dataset* dengan lebih lambat maka akan besar kemungkinan model akan mempelajari dataset dengan sangat detail. Sedangkan *batch size* bertujuan untuk mengatur jumlah sampel yang diproses dalam satu epoch pada proses *training,* pada konfigurasi ini dapat dilihat bahwa nilai mAP dari skema II dengan *batch size* 8 lebih tinggi dibandingkan dengan skema I *batch size* 16 [10]. Hal ini disebabkan karena pada model skema II mempelajari dataset untuk setiap *epoch* lebih sedikit dibandingkan model skema I, sehingga model dapat mempelajari dataset dengan mudah. Maka dapat disimpulkan bahwa model yang akan dideploy pada *server* atau *cloud* adalah hasil model skema II.

REFERENSI

[1] D. T. Laswati, "Masalah Gizi Dan Peran Gizi Seimbang," *Agrotech J. Ilm. Teknol. Pertan.*, vol. 2, no. 1, pp. 69–73, 2019.

[2] M. K. R. Indonesia, "Peraturan Menteri Kesehatan Republik Indonesia Nomor 28 Tahun 2019 Tentang Angka Kecukupan Gizi Yang Dianjurkan Untuk Masyarakat Indonesia," *Carbohydr. Polym.*, vol. 1, no. 1, p. 17, 2019.

[3] K. G. Kim, "Deep learning book review," *Nature*, vol. 29, no. 7553, pp. 1–73, 2019.

[4] A. Raup, W. Ridwan, Y. Khoeriyah, S. Supiana, and Q. Y. Zaqiah, "Deep Learning dan Penerapannya dalam Pembelajaran," *JIIP - J. Ilm. Ilmu Pendidik.*, vol. 5, no. 9, pp. 3258–3267, 2022.

[5] I. N. T. Post, "Deep Learning : How Intelligent Machines Learn and Progress," 2021. [Online]. Available: https://www.g2.com/articles/deep-learning. [Accessed: 15-Jul-2023].

[6] Shreya Mattoo, "Object Recognition: Widening the Horizon of Computer Vision," 2022. [Online]. Available: https://www.g2.com/articles/object-recognition. [Accessed: 15-Jul-2023].

[7] C. Issues, D. Actions, P. Wiki, B. Tags, and F. Cli, "NEW - YOLOv8 ? in PyTorch > ONNX > OpenVINO > CoreML > TFLite." [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: 15-Jul-2023].

[8] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," *IBM J. Res. Dev.*, vol. 61, no. 4, pp. 1–20, 2017.

[9] I. Gridin, *Hyperparameter Optimization*. 2022.

[10] G. Plastiras, C. Kyrkou, and T. Theocharides, "Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing," *ACM Int. Conf. Proceeding Ser.*, 2018.