



**UNIVERSITAS INDONESIA**

**Rancang Bangun Sistem Pendeteksi Makanan Banyak Objek Menggunakan  
YOLOv5 pada Aplikasi *Mobile* Berbasis Flutter**

**SKRIPSI**

**Yusuf Abraham Bismo Kristanto**

**1806200274**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER**

**DEPOK**

**JUNI 2022**



**UNIVERSITAS INDONESIA**

**Rancang Bangun Sistem Pendeteksi Makanan Banyak Objek Menggunakan  
YOLOv5 pada Aplikasi *Mobile* Berbasis Flutter**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik**

**Yusuf Abraham Bismo Kristanto**

**1806200274**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER**


**DEPOK  
JUNI 2022**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Yusuf Abraham Bismo Kristanto**

**NPM : 1806200274**

**Tanda Tangan : **

**Tanggal : 20 Juni 2022**

## LEMBAR PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Yusuf Abraham Bismo Kristanto  
NPM : 1806200274  
Program Studi : Teknik Komputer  
Judul Skripsi : Rancang Bangun Sistem Pendeteksi Makanan Banyak Objek  
Menggunakan YOLOv5 pada Aplikasi Mobile Berbasis Flutter

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagaibagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana mata kuliah Skripsi pada Program Studi Teknik Komputer Fakultas Teknik Universitas Indonesia.

### DEWAN PENGUJI

Pembimbing : Dr. Prima Dewi Purnamasari, S.T., M.T., M.Sc. (.....)

Penguji : Yan Maraden, S.T., M.T., M.Sc. (.....)

Penguji : I Gde Dharma Nugraha, S.T., M.T., Ph.D (.....)

Ditetapkan di : Fakultas Teknik

Tanggal : 12 Juli 2022

## KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya saya dapat menyelesaikan laporan skripsi ini. Penulisan skripsi bertujuan untuk memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik. Dalam proses penulisan, penulis menyadari bahwa telah mendapat banyak dukungan yang membantu dalam proses riset maupun penulisan. Oleh karena itu penulis ingin mengucapkan terima kasih yang mendalam kepada berbagai pihak yang telah membantu selama pengerjaan seminar, yaitu:

1. Dr. Prima Dewi Purnamasari, S.T., M.T., M.Sc, selaku dosen pembimbing yang telah mengarahkan dan memberikan bantuan serta saran-saran dalam penyusunan seminar.
2. Dr. Ir. Dodi Sudiana M.Eng, selaku dosen pembimbing akademis yang selama ini telah memberikan arahan dan bantuan selama perkuliahan penulis di Fakultas Teknik Universitas Indonesia.
3. Segenap dosen program studi Teknik Komputer.
4. Keluarga penulis, Ir. Gabriel Andari Kristanto, Anton Kristanto, dan Benjamin Kristanto yang telah memberikan dukungan serta fasilitas kepada penulis selama menjalani masa perkuliahan.
5. Seluruh rekan-rekan yang menyemangati selama pembuatan skripsi, tim Bangkit Phoodto, asisten Laboratorium Jaringan, beserta dukungan segala teman di Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Penulis juga berharap agar penulisan skripsi ini dapat menjadi referensi serta membawa manfaat dan memberikan wawasan bagi pembaca. Penulis menyadari laporan skripsi ini jauh dari sempurna. Untuk itu dengan kerendahan hati penulis mengharapkan saran dan kritik yang sifatnya membangun dari semua pihak demi membangun laporan penelitian ini.

Depok, Juni 2022



Yusuf Abraham Bismo Kristanto

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertandatangan di bawah ini:

Nama : Yusuf Abraham Bismo Kristanto  
NPM : 1806200274  
Program Studi : Teknik Komputer  
Fakultas : Fakultas Teknik  
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

Rancang Bangun Sistem Deteksi Objek Makanan dengan Implementasi *You Only Look Once v5* dalam Aplikasi *Mobile* Berbasis Flutter

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 20 Juni 2022  
Yang menyatakan



Yusuf Abraham Bismo Kristanto

## ABSTRAK

Nama : Yusuf Abraham Bismo Kristanto  
Program Studi : Teknik Komputer  
Judul : Rancang Bangun Sistem Pendeteksi Makanan Banyak Objek  
Menggunakan YOLOv5 pada Aplikasi Mobile Berbasis Flutter

Seiring dengan perkembangan bidang *computer vision* terdapat lebih banyak solusi yang dapat diimplementasikan untuk bidang sehari-hari. Salah satu bidang yang paling erat dengan kegiatan sehari-hari adalah kegiatan mengonsumsi makanan. Dalam memperhatikan pola makan, penting dilakukan proses mengidentifikasi jenis makanan yang dikonsumsi. Dengan memanfaatkan perkembangan model *machine learning* deteksi objek yang bekerja secara waktu langsung, YOLOv5 dapat digunakan untuk melakukan deteksi objek untuk dapat mengidentifikasi berbagai jenis makanan dalam suatu gambar. Dengan menggunakan YOLOv5, deteksi terhadap makanan yang kerap kali dikonsumsi oleh masyarakat Indonesia dapat dilakukan dan ditingkatkan akurasi dengan pemrosesan gambar hingga mencapai nilai mAP 94,3%. Penggunaan implementasi model ini dalam aktivitas sehari-hari dapat memberikan nilai tambah kepada orang-orang yang ingin lebih memahami jenis makanan yang konsumsinya. Dari hasil pengujian *user experience* yang dilakukan terhadap aplikasi, hasil perbandingan terhadap *benchmark* mengindikasikan bahwa aplikasi memiliki kualitas penggunaan di atas rata-rata dengan nilai 1,37 untuk daya tarik, 1,58 untuk kejelasan, 1,23 untuk efisiensi, 1,38 untuk ketepatan, 1,13 untuk stimulasi, dan 1,01 untuk kebaruan.

Kata Kunci:  
*computer vision*; deteksi objek; *machine learning*; waktu langsung

## ABSTRACT

Name : Yusuf Abraham Bismo Kristanto  
Study Program : Teknik Komputer  
Title : Development of a Multi-object Food Detection System using YOLOv5 for Flutter-Based Mobile Applications

*With the advent of computer vision there are more solutions that can be implemented in everyday life. One of the areas most closely related to daily activities is the activity of consuming food. In paying attention to diet, it is important to identify the type of food consumed. By leveraging the development of object detection machine learning models that work in real time, YOLOv5 can be used to perform object detection to identify different foods within a single image. By using YOLOv5, detection of foods that are often consumed by Indonesian people can be carried out and the accuracy is increased by image processing up to a value of mAP 94.3%. The use of this model's implementation in daily activities can provide added value to people who want to better understand the types of food they consume. From the results of user experience testing carried out on the object detection application, the results of comparisons against benchmarks indicate that the application has above average usage quality with a value of 1.37 for attractiveness, 1.58 for clarity, 1.23 for efficiency, 1.38 for accuracy, 1.13 for stimulation, and 1.01 for novelty.*

Key words:

*Computer vision; object detection; machine learning; real time*



## DAFTAR ISI

|   |          |
|---|----------|
| HALAMAN PERNYATAAN ORISINALITAS .....   | iii      |
| LEMBAR PENGESAHAN .....   | iv       |
| KATA PENGANTAR.....   | v        |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA<br>ILMIAH UNTUK KEPENTINGAN AKADEMIS ..... | vi       |
| ABSTRAK .....   | vii      |
| ABSTRACT .....  | viii     |
| DAFTAR ISI.....   | ix       |
| DAFTAR GAMBAR.....  | xi       |
| DAFTAR TABEL .....  | xiv      |
| DAFTAR ISTILAH .....  | xv       |
| <b>BAB 1 PENDAHULUAN .....</b>  | <b>1</b> |
| 1.1. Latar Belakang .....   | 1        |
| 1.2. Rumusan Masalah .....  | 2        |
| 1.3. Tujuan Penelitian .....  | 3        |
| 1.4. Batasan Masalah .....  | 3        |
| 1.5. Metodologi Penelitian .....  | 4        |
| 1.6. Sistematika Penulisan .....  | 5        |
| <b>BAB 2 LANDASAN TEORI DETEKSI OBJEK DENGAN YOLO .....</b>                               | <b>6</b> |
| 2.1. CNN (Convolutional Neural Network) .....   | 6        |
| 2.2. Object Detection .....   | 6        |
| 2.2.1. Anatomi <i>Object Detection</i> .....  | 7        |
| 2.3. YOLO ( <i>You Only Look Once</i> ) .....   | 8        |
| 2.4. YOLOv5.....  | 11       |
| 2.4.1. Arsitektur YOLOv5 .....  | 12       |
| 2.4.2. Benchmark YOLOv5 .....   | 15       |
| 2.5. Tolak Ukur Evaluasi Model .....  | 16       |
| 2.5.1. Precision dan Recall .....   | 16       |
| 2.5.2. Intersection over Union .....  | 16       |
| 2.5.3. Mean Average Precision .....   | 17       |
| 2.6. Platform Pengembangan yang Digunakan .....   | 18       |
| 2.6.1. Flutter .....  | 18       |

|  |   |           |
|--|---|-----------|
| 2.6.2.                                       | Google Firebase .....   | 18        |
| 2.6.3.                                       | Google Cloud Platform .....   | 19        |
| 2.6.4.                                       | Google Colab Pro .....  | 19        |
| 2.6.5.                                       | Roboflow .....  | 20        |
| 2.7.   | User Experience Questionnaire (UEQ) .....   | 21        |
| 2.8.   | Penelitian Terdahulu .....  | 23        |
| 2.8.1.                                       | BTBU-60: <i>dataset for object detection in food field</i> [22] .....   | 23        |
| 2.8.2.                                       | <i>Real-time Food-Object Detection and Localization for Indian Cuisines using Deep Neural Networks</i> [23] ..... | 23        |
| <b>BAB 3 PERANCANGAN SISTEM .....</b>        |   | <b>25</b> |
| 3.1.   | System Requirement .....  | 25        |
| 3.2.   | Metode Penelitian .....   | 25        |
| 3.3.   | Pengumpulan Data .....  | 27        |
| 3.4.   | Data Preprocessing dan Pelabelan Data .....   | 28        |
| 3.5.   | Augmentasi dan Distribusi Data .....  | 30        |
| 3.6.   | Pelatihan Model .....   | 31        |
| 3.7.   | Evaluasi Model Deteksi Objek Makanan .....  | 33        |
| 3.8.   | Pengembangan Aplikasi .....   | 35        |
| <b>BAB 4 IMPLEMENTASI DAN ANALISIS .....</b> |   | <b>37</b> |
| 4.1.   | Implementasi Sistem .....   | 37        |
| 4.1.1.                                       | Pengembangan Model Deteksi Objek .....  | 37        |
| 4.1.2.                                       | Pengembangan Aplikasi Flutter .....   | 38        |
| 4.1.3.                                       | Pengembangan Backend Aplikasi dan Deployment .....  | 42        |
| 4.2.   | Dataset Pengujian .....   | 43        |
| 4.3.   | Skenario Pengujian .....  | 47        |
| 4.4.   | Pengujian Model .....   | 48        |
| 4.4.1.                                       | Pengujian Pengaruh Penambahan Kategori Makanan Terhadap Akurasi .....   | 48        |
| 4.4.2.                                       | Pengujian Pengaruh Augmentasi Data Terhadap Akurasi .....   | 51        |
| 4.4.3.                                       | Pengujian Model pada Aplikasi Deteksi Objek .....   | 55        |
| 4.4.4.                                       | Pengujian dengan Sudut Gambar Berbeda-beda .....  | 59        |
| 4.4.5.                                       | Pengujian Sistem Deteksi Banyak Objek .....   | 62        |
| 4.5.   | User Experience Testing .....   | 64        |
| 4.6.   | Dampak Terhadap Lingkungan Hidup dan Masyarakat .....   | 68        |
| <b>BAB 5 KESIMPULAN .....</b>                |   | <b>70</b> |
| 5.1.   | Kesimpulan .....  | 70        |
| 5.2.   | Saran .....   | 70        |
| <b>DAFTAR ACUAN .....</b>                    |   | <b>72</b> |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1. Komponen yang Berada di Dalam Sebuah Pendeteksi Objek Berjenis Satu Tahap dan Dua Tahap [10].....                         | 7  |
| Gambar 2.2. Ilustrasi dari Proses Prediksi YOLO Menggunakan Sistem Prediksi yang Dilakukan oleh Masing-masing <i>Grid Cell</i> . .... | 8  |
| Gambar 2.3. Arsitektur dari Model Deteksi Objek You Only Look Once [11] ....  | 10 |
| Gambar 2.4. Ilustrasi Arsitektur Backbone, Neck, dan Head YOLOv5 [14] .....   | 12 |
| Gambar 2.5. Sebuah Ilustrasi Dense Block. DenseNet Mengambil Semua Feature Map yang Mendahului Sebagai Input [15].....                | 13 |
| Gambar 2.6. Ilustrasi DenseNet Normal [16] .....  | 13 |
| Gambar 2.7. Ilustrasi DenseNet dengan Modifikasi CSP (Cross Stage Partial) [16] .....   | 14 |
| Gambar 2.8. Ilustrasi Perbedaan FPN, PANet, dan NAS-FPN [18].....   | 14 |
| Gambar 2.9. Perbandingan Berbagai Versi YOLOv5 dengan Algoritma EfficientDet.....   | 15 |
| Gambar 2.10. Tampilan Roboflow yang Digunakan Sebagai Platform untuk Merancang Dataset.....   | 21 |
| Gambar 3.1. Rancangan Perancangan Sistem Deteksi Objek dengan YOLOv5 Ultralytics.....   | 27 |
| Gambar 3.2. Tampilan Graphical User Interface (GUI) dari Platform Roboflow untuk Pelabelan Data .....                                 | 28 |
| Gambar 3.3. Contoh Gambar yang Telah Diberikan Label Data Sesuai dengan Kategori Objek Makanan dalam Roboflow .....                   | 29 |
| Gambar 3.4. Alur Arsitektur dan Pelatihan Model dari YOLOv5 Pytorch [25].....   | 31 |

|   |    |
|---|----|
| Gambar 3.5. Perubahan Konfigurasi yang Dilakukan pada YOLOv5s .....   | 32 |
| Gambar 3.6. Flowchart Metode Evaluasi Model Hasil Pelatihan pada Dataset ....   | 34 |
| Gambar 3.7. Use Case Diagram untuk Aplikasi Phoodto .....   | 35 |
| Gambar 4.1. Diagram Alur Aktivitas yang Mendeskripsikan Cara Kerja Aplikasi Secara Keseluruhan .....  | 39 |
| Gambar 4.2. Tampilan Login dan Registrasi Aplikasi.....   | 40 |
| Gambar 4.3. Tampilan Laman Utama dan Laman User Profile .....   | 41 |
| Gambar 4.4. Tampilan Console Google Firebase pada Layanan Cloud Storage yang Menjadi Tempat Penyimpanan Gambar Hasil Deteksi Model Deteksi Objek. ....                  | 42 |
| Gambar 4.5. Kumpulan Data di Dalam Dataset yang Terdiri dari Gambar Berbagai Kategori Makanan yang Telah Diberikan Label yang Terkait pada Jenis Makanan Tertentu.....  | 45 |
| Gambar 4.6. Perkembangan Hasil mAP Seiring dengan Peningkatan Jumlah Kategori dalam Dataset.....  | 50 |
| Gambar 4.7. Perbandingan Kurva Precision-Recall Antara Model Hasil Training Sebelum (kiri) dan Sesudah (kanan) Augmentasi Data. ....                                    | 52 |
| Gambar 4.8 Hasil prediksi yang dilakukan oleh model dengan data yang sudah diaugmentasi .....   | 53 |
| Gambar 4.9. Kode yang Digunakan untuk Mengunduh Model Deteksi Objek yang Telah Dirancang ke dalam Sistem <i>Backend</i> .....   | 55 |
| Gambar 4.10. Kode yang Digunakan untuk Memetakan Hasil Prediksi Kepada Gambar dan Menyimpannya di Dalam Penyimpanan Lokal. ....   | 56 |
| Gambar 4.11. Hasil Javascript Object Notation (JSON) yang Dikembalikan dalam Bentuk HTTP Response Ketika Terdapat Request yang Ditujukan kepada Endpoint REST API. .... | 56 |

|  |    |
|--|----|
| Gambar 4.12. Hasil yang Ditampilkan oleh Aplikasi Sesuai dengan Hasil Prediksi yang Ditetapkan oleh Sistem <i>Backend</i> yang Mengimplementasikan Model Deteksi Objek Makanan. .... | 57 |
| Gambar 4.13. Contoh Hasil Pengujian yang Dilakukan untuk Proses Deteksi Banyak Objek.....  | 63 |
| Gambar 4.14. Tampilan Kuisisioner UEQ yang Disebarkan Melalui Google Forms .....   | 65 |
| Gambar 4.15. Bagan Persebaran Jenis Kelamin Responden User Experience Questionnaire (UEQ) .....  | 65 |
| Gambar 4.16. Bagan Persebaran Daerah Provinsi Asal Responden User Experience Questionnaire (UEQ) .....   | 66 |
| Gambar 4.17. Distribusi Jawaban Hasil Kuisisioner UEQ Untuk Tiap Item .....  | 66 |
| Gambar 4.18. Rata-rata Nilai UEQ dari Desain Aplikasi Phoodto.....   | 67 |
| Gambar 4.19. Diagram Benchmark UEQ untuk Aplikasi Phoodto .....  | 67 |

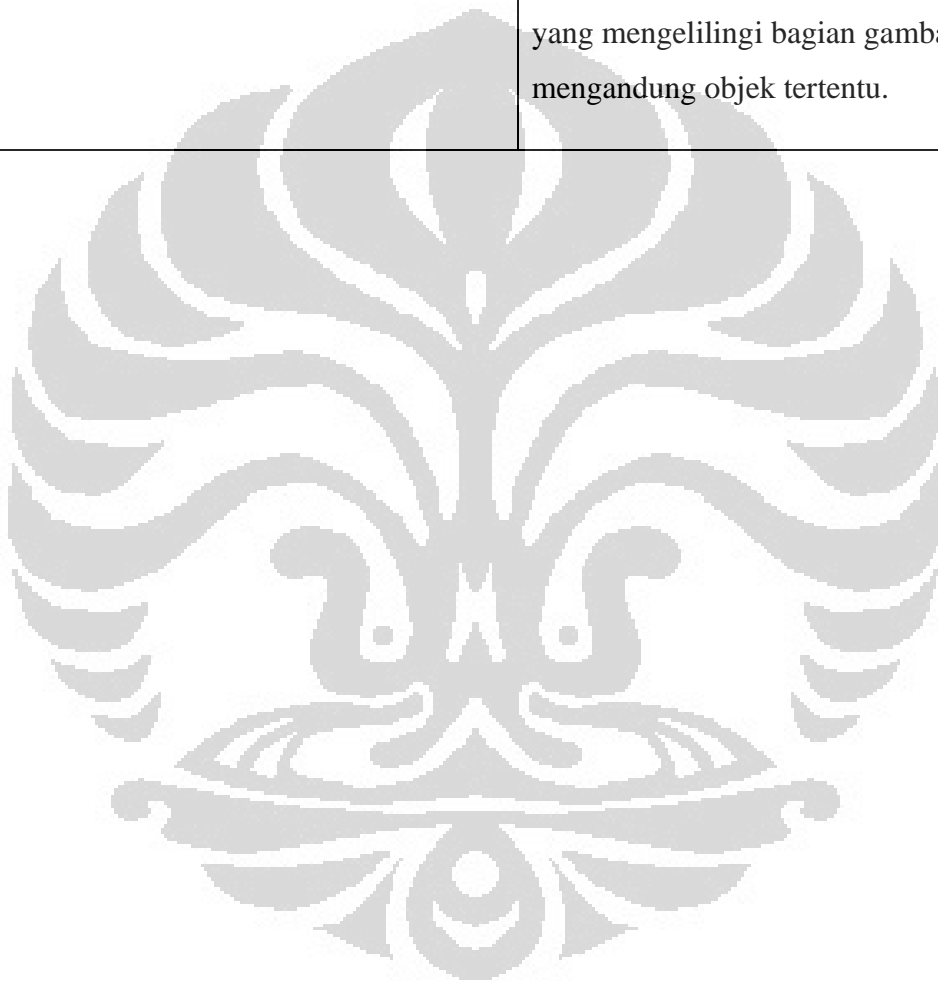
## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 2.1. Spesifikasi Perangkat yang Disediakan saat Menggunakan Google Colab Pro .....   | 20 |
| Tabel 4.1. Spesifikasi Jumlah Data Setiap Kategori pada Dataset Akhir Sebelum Augmentasi Gambar .....                            | 44 |
| Tabel 4.2. Spesifikasi Jumlah Data Setiap Kategori pada Dataset Akhir Setelah Augmentasi Gambar .....                            | 46 |
| Tabel 4.3. Daftar Kategori yang Mendeskripsikan Perkembangan Dataset Deteksi Objek Makanan yang Dilakukan.....                   | 49 |
| Tabel 4.4. Perkembangan Hasil Model Deteksi Objek Makanan Seiring dengan Perkembangan Ukuran Dataset. ....                       | 50 |
| Tabel 4.5. Perbandingan Data Hasil Training Model Terbaik pada Dataset Sebelum Augmentasi Data dan Setelah Augmentasi Data. .... | 51 |
| Tabel 4.6. Perbandingan Mean Average Precision (mAP) Untuk Setiap Kategori Dataset Sebelum dan Sesudah Augmentasi Data.....      | 54 |
| Tabel 4.7. Spesifikasi Perangkat Android Emulator yang Digunakan untuk Menjalankan Aplikasi Secara Lokal .....                   | 57 |
| Tabel 4.8. Hasil Pengujian Implementasi Model Deteksi Objek Makanan ke dalam Aplikasi .....                                      | 58 |
| Tabel 4.9. Contoh Pengambilan Gambar untuk Pengujian Sudut Berbeda-beda ..   | 60 |
| Tabel 4.10. Hasil Pengujian Deteksi Objek Makanan pada Sudut Gambar yang Berbeda-beda .....                                      | 61 |
| Tabel 4.11. Hasil Pengujian Sistem Deteksi Banyak Objek pada Dua Kategori Sekaligus di dalam Satu Gambar .....                   | 62 |

## DAFTAR ISTILAH

| Istilah           | Penjelasan  |
|-------------------|---|
| Machine Learning: | Salah satu perkembangan Artificial Intelligence yang memungkinkan mesin atau komputer untuk mempelajari suatu hal dengan melakukan komputasi algoritma menggunakan data sebagai input.                  |
| Computer Vision:  | Salah satu divisi dari machine learning yang memungkinkan komputasi algoritma untuk mendeteksi fitur tertentu pada gambar sehingga memungkinkan pemecahan masalah klasifikasi atau deteksi pada gambar. |
| Object Detection: | Pemrosesan gambar yang dapat mengidentifikasi lebih dari satu objek di dalam suatu gambar.  |
| YOLO:             | You Only Look Once yang merupakan algoritma deteksi objek yang dikembangkan oleh Joseph Redmon dan Alexey B.  |
| Dataset:          | Kumpulan data yang digunakan sebagai input untuk dapat melakukan training dengan machine learning.  |
| Model:            | Hasil pelatihan machine learning yang menghasilkan suatu <i>weights</i> yang telah dilatih pada suatu dataset tertentu  |

|               |   |
|---------------|---|
| Label:        | Spesifikasi yang diberikan pada data di dalam dataset agar model dapat mengidentifikasi dan belajar untuk mengasosiasikan data dengan label yang spesifik terhadap data tersebut. |
| Bounding box: | Jenis label yang digunakan dalam proses deteksi objek. Berupa kotak yang mengelilingi bagian gambar yang mengandung objek tertentu.   |





# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Menurut sebuah penelitian pada penduduk di Amerika Serikat, orang-orang yang memiliki defisiensi terhadap nutrisi esensial serta mengonsumsi lebih banyak lemak dan sodium memiliki angka mortalitas yang lebih tinggi [1]. Penelitian tersebut juga mengaitkan konsumsi nutrisi dan menurunnya angka mortalitas pada populasi. Konsumsi nutrisi penting seperti vitamin A, vitamin E, dan kalsium dapat mencegah berbagai macam penyakit seperti penyakit kardiovaskular, kanker. Sehingga, penemuan tersebut menyimpulkan bahwa pemasukan yang lebih tinggi dari nutrien yang terdefisiensi dapat diasosiasikan dengan berkurangnya risiko penyakit kardiovaskular dan kanker pada orang dewasa.

Salah satu hal yang menjadi faktor terhadap pengkonsumsian makanan yang kurang sehat adalah pengetahuan nutrisi yang terbatas serta tingkah laku seseorang [2]. Selain itu, terdapat juga faktor harga jenis makanan tertentu yang lebih mahal di negara berkembang. Faktor kedua berada di luar tangan individu sementara faktor pertama dapat diubah dengan adanya perubahan sikap. Terlepas dari situasi geografis dan ekonomis, perlu disadari kepentingan mengonsumsi makanan dengan gizi yang cukup serta membatasi konsumsi makanan dengan sodium dan lemak yang tinggi karena keduanya merupakan faktor kunci dalam menjaga kesehatan.

Mengidentifikasi pola makan merupakan salah satu kunci untuk mengetahui dan memahami makanan yang dikonsumsi. Dengan kemampuan untuk mengidentifikasi jenis makanan, maka dapat diperoleh kapabilitas untuk mengedukasi kandungan nutrisi yang masuk ke dalam pola konsumsi. Pada akhir-akhir ini, terjadi peningkatan akan kesadaran untuk melakukan pencatatan terhadap apa saja yang dikonsumsi agar dapat dipantau dengan baik. Dengan meningkatnya perhatian akan pentingnya mencatat pola konsumsi makan dan dampak kesehatannya maka permasalahan identifikasi makanan dapat menjadi pengaplikasian dari *computer vision* yang praktis [3].

Seiring dengan berkembangnya *computer vision* beserta dengan model-model *transfer learning* yang semakin didemokratisasi untuk dipakai dan diimplementasikan, terdapat semakin banyak situasi dimana bidang ini dapat diaplikasikan. Terdapat semakin banyak riset baru yang dikembangkan untuk merilis metode *transfer learning* baru yang dapat diimplementasikan secara langsung untuk deteksi maupun dikembangkan dengan melakukan training lebih lanjut [4].

Pengaplikasian metode deteksi objek pada makanan dapat mempermudah dalam memberikan umpan balik mengenai kandungan nutrisi serta potensi dampaknya kepada pengguna. Oleh karena itu, pengidentifikasian makanan dapat secara signifikan mengubah gaya hidup dan pola makan apabila diintegrasikan dalam kehidupan sehari-hari.

Meskipun terjadi peningkatan perhatian dan kesadaran mengenai menjaga pola makan dan kandungan nutrisinya, pada masa sekarang masih banyak aplikasi yang membutuhkan entri secara manual ke dalam sistem yang dipakai. Ini menjadi suatu halangan karena menyulitkan dan memakan waktu. Oleh karena itu, aplikasi-aplikasi seperti ini tidak mendapat adopsi secara luas untuk masyarakat secara umum. Ditambah lagi dengan keterbatasan waktu di masa sekarang, semakin sulit untuk mengingat dan menginput semua hal yang dikonsumsi ke dalam sebuah aplikasi untuk mengetahui kandungan nutrisinya.

Sehingga, untuk memecahkan masalah ini dibutuhkan cara untuk mengidentifikasi makanan dan kandungan nutrisinya secara cepat dan terotomasi. Metode yang ditempuh adalah dengan membuat sebuah sistem deteksi objek yang dapat mengidentifikasi komponen makanan secara individu, kemudian memberikan umpan balik kepada pengguna mengenai detail dari makanan tersebut.

## **1.2. Rumusan Masalah**

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara merancang sebuah model deteksi objek yang dapat digunakan untuk mengklasifikasikan komponen makanan di dalam suatu gambar dengan tingkat akurasi yang tinggi?

2. Bagaimana cara mengimplementasikan model deteksi objek makanan yang dapat diaplikasikan ke perangkat mobile agar dapat digunakan secara langsung oleh pengguna aplikasi?

### **1.3. Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan mengimplementasikan sistem deteksi objek makanan dalam suatu gambar menggunakan algoritma YOLOv5 untuk mampu mengklasifikasikan komponen makanan individu ke dalam kelas tertentu.
2. Menguji performa, akurasi, dan menjalankan model deteksi objek YOLOv5.
3. Merancang dan mengimplementasikan sistem deteksi objek dalam sebuah aplikasi berbasis Flutter yang mampu mendeteksi objek makanan di dalam sebuah gambar dan memberikan deskripsi makanan tersebut.

### **1.4. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah pembentukan model deteksi objek makanan yang berfungsi untuk mendeteksi objek makanan individu di dalam sebuah gambar. Untuk lebih detailnya berikut batasan masalah penelitian ini:

1. Pengujian performa dari model mencakup akurasi dari deteksi kelas objek, seiring dengan penambahan data pada tahap perancangan dataset, serta perbandingan akurasi sebelum dan sesudah pemrosesan gambar.
2. Deteksi objek makanan dibatasi pada 40 kelas objek makanan yang dikumpulkan melalui pengumpulan data melalui sumber online dan proses pemberian label secara mandiri.
3. Deploy sistem deteksi objek makanan yang dirancang akan dilakukan di Google Cloud Platform (GCP) dengan sumber daya memori 2 GB dan prosesor 4 CPU.
4. Aplikasi yang mengimplementasikan sistem deteksi objek mampu untuk melakukan deteksi terhadap minimal satu gambar dan maksimal tergantung keberadaan objek yang terdeteksi di dalam sebuah gambar.

### 1.5. Metodologi Penelitian

Metode yang digunakan selama penelitian ini adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pendalaman mengenai teori yang berkaitan dengan deteksi objek makanan, *deep learning*, beserta dengan metode-metode *deep learning* yang digunakan untuk permasalahan deteksi objek yang berkaitan dengan topik skripsi.

2. Konsultasi dengan dosen pembimbing

Pada tahap ini dilakukan pembahasan mengenai jalur yang paling baik untuk ditempuh di dalam riset dan penelitian yang dilakukan.

3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang akan digunakan sebagai sumber utama untuk pelatihan model *machine learning*.

4. Pengembangan dan Implementasi Model

Pada tahap ini akan dilakukan pelatihan model *machine learning* untuk dapat melakukan deteksi objek makanan dalam sebuah gambar.

5. Pengembangan Aplikasi

Pada tahap ini akan dilakukan pengembangan aplikasi *frontend* dan *backend* agar model deteksi objek dapat diimplementasikan dan digunakan oleh pengguna secara langsung.

6. Analisis dan Implementasi Model

Pada tahap ini akan dilakukan analisis terhadap berbagai faktor seperti performa model terhadap berbagai kategori yang dapat dideteksi, serta cara pengimplementasian model ke dalam sistem deteksi objek.

7. Kesimpulan

Tahap ini merupakan tahap pengambilan kesimpulan dari hasil penelitian dan analisis yang dilakukan.

## 1.6. Sistematika Penulisan

Sistematika Penulisan pada penelitian ini dibagi ke dalam 5 bab, yaitu:

### BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, batasan masalah dan metodologi yang digunakan dalam penyusunan skripsi.

### BAB II DASAR TEORI

Bab ini menjelaskan teori yang menjadi dasar perancangan yang digunakan dalam penelitian skripsi.

### BAB III PERANCANGAN SISTEM

Bab ini menjelaskan tahap perancangan sistem dengan membentuk sistem pelatihan model.

### BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini membahas dan menganalisis hasil dari implementasi model dan proses pengembangan aplikasi serta hasil *user experience testing*.

### BAB V PENUTUP

Bab ini berisikan kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

## BAB 2

### LANDASAN TEORI DETEKSI OBJEK DENGAN YOLO

#### 2.1. CNN (Convolutional Neural Network)

Sebuah Convolutional Neural Network atau CNN adalah sebuah algoritma *Deep Learning* yang dapat mengambil input sebuah gambar, kemudian melakukan penetapan kepentingan, yang terdiri dari *weights* dan *bias* pada berbagai objek atau aspek berbeda pada sebuah gambar [5]. Sehingga, algoritma ini dapat membedakan aspek atau objek dalam gambar tersebut berbeda dari satu sama lain. CNN memiliki perbedaan dengan metode primitif karena CNN mampu untuk mempelajari sendiri filter dan karakteristik yang diperlukan untuk dapat melakukan permasalahan klasifikasi atau deteksi.

CNN mampu untuk mendeteksi dependensi spasial dan temporal di dalam sebuah gambar melalui aplikasi dari filter yang relevan [6]. Ini berarti bahwa CNN memiliki kemampuan untuk menafsirkan lokasi dari hal yang diklasifikasikan dalam sebuah gambar.

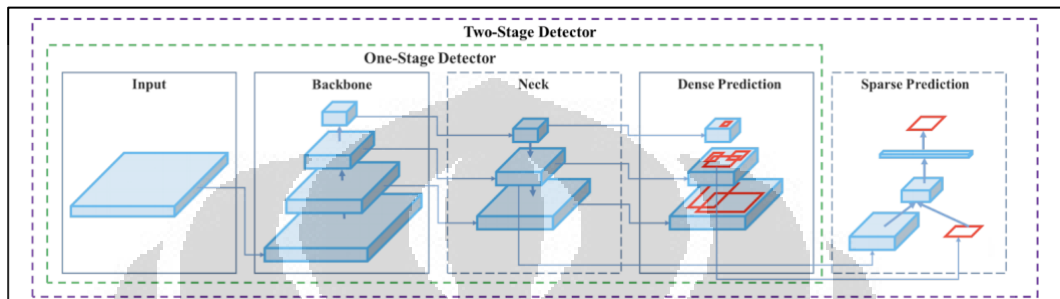
Sebuah convolution adalah pengaplikasian sebuah filter kepada input yang mengakibatkan aktivasi. Pengaplikasian filter ini secara repetitif dapat menyebabkan sebuah pemetaan dari aktivasi yang disebut sebagai *feature map*. *Feature map* memberikan indikasi mengenai fitur-fitur spesifik yang dapat dideteksi dimanapun di dalam input gambar yang diberikan.

#### 2.2. Object Detection

*Object detection* atau deteksi objek adalah permasalahan computer vision yang bersinggungan dengan permasalahan identifikasi dan lokalisasi objek dari kelas tertentu di dalam sebuah gambar [7]. Deteksi objek merupakan tugas untuk mengidentifikasi berbagai kelas spesifik yang berada di dalam sebuah gambar [8]. Saat ditemukan sebuah objek di dalam sebuah gambar, kelas objek ini dikotakkan ke dalam sebuah *bounding box* dan kelasnya diidentifikasi oleh model pendeteksi objek. Permasalahan *object detection* pada umumnya dilatih dan dievaluasi

terhadap dataset yang terstandarisasi, misalnya COCO dataset [9]. COCO dataset merupakan dataset yang memiliki 80 kelas objek. Dari hasil pelatihan dan evaluasi tersebut, diasumsikan bahwa model deteksi objek ini mampu untuk generalisasi terhadap permasalahan deteksi objek lainnya dengan data pelatihan yang baru.

### 2.2.1. Anatomi *Object Detection*



Gambar 2.1. Komponen yang Berada di Dalam Sebuah Pendeteksi Objek Berjenis Satu Tahap dan Dua Tahap [10]

Pendeteksi objek dapat dibagi menjadi dua kategori, yakni pendeteksi satu tahap dan pendeteksi dua tahap [11]. Pendeteksi dua tahap membagi permasalahan menjadi dua, yaitu lokalisasi objek dan klasifikasi setiap *bounding box*. Sedangkan, pendeteksi satu tahap dapat melakukan prediksi lokalisasi dan klasifikasi pada waktu yang bersamaan.

Di dalam sebuah pendeteksi objek terdapat beberapa bagian, yang pertama adalah input, kemudian backbone, dan juga neck. Seperti terlihat pada Gambar 2.1, terdapat perbedaan di antara pendeteksi objek satu tahap dengan pendeteksi objek dua tahap. Di dalam pendeteksi objek dua tahap, lokalisasi dan klasifikasi untuk setiap *bounding box* berada pada dua tahap yang berbeda. Sedangkan, pada pendeteksi objek satu tahap, lokalisasi dan klasifikasi *bounding box* dilakukan pada waktu yang bersamaan.

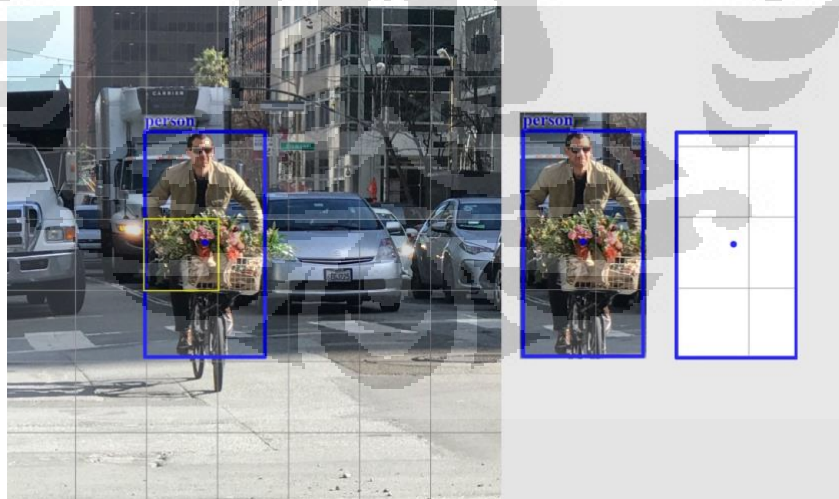
Secara umum, cara kerja sebuah pendeteksi objek berawal dengan mengambil gambar sebagai sebuah input. Kemudian terjadi kompresi fitur-fitur yang didapatkan dari input melalui sebuah *backbone Convolutional Neural Network*. Untuk dapat melakukan deteksi objek, perlu dilakukan penetapan lebih dari satu *bounding box* yang mengelilingi bagian gambar yang terdeteksi. Untuk dapat melakukan hal ini maka perlu dilakukan pencampuran dari *feature layer* pada

bagian backbone. Tahap pencampuran lapisan *feature layer* dari backbone dilakukan pada bagian neck. Salah satu contoh dari pendeteksi objek satu tahap adalah YOLO, atau *You Only Look Once*, dan karena dapat melakukan lokalisasi dan prediksi pada waktu yang bersamaan, maka dinamakan nama tersebut.

### 2.3. YOLO (*You Only Look Once*)

YOLO adalah sebuah sistem deteksi objek yang ditargetkan untuk pemrosesan secara *real-time*. YOLO merupakan singkatan dari *You Only Look Once* dan dikembangkan oleh Joseph Redmon dalam sebuah *framework* yang dinamakan Darknet [11]. Darknet merupakan framework riset yang memproduksi solusi deteksi objek dalam waktu langsung di dalam *computer vision*.

Untuk dapat melakukan deteksi objek di dalam sebuah foto, YOLO akan menempatkan sebuah *bounding box* yang mengelilingi daerah gambar yang merepresentasikan objek tersebut. Cara kerja YOLO adalah dengan membagi gambar ke dalam sebuah grid  $S \times S$ , dimana apabila titik tengah dari suatu objek berada dalam *grid cell* tersebut, maka *grid cell* tersebutlah yang bertanggungjawab untuk mendeteksi objek tersebut.



Gambar 2.2. Ilustrasi dari Proses Prediksi YOLO Menggunakan Sistem Prediksi yang Dilakukan oleh Masing-masing *Grid Cell*.

Gambar 2.2 adalah contoh ilustrasi penggunaan YOLO pada dataset PASCAL VOC. PASCAL VOC (Visual Object Classes) 2012 terdiri dari 20 kategori yang di dalamnya termasuk berbagai kategori yang digunakan sebagai



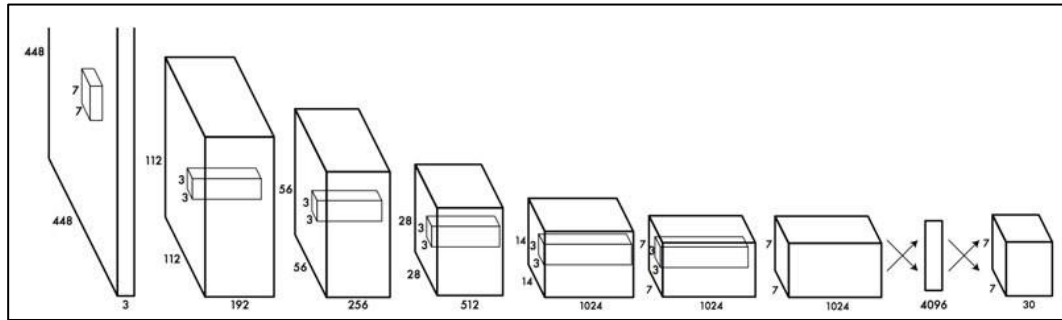
salah satu *benchmark* lain yang digunakan untuk permasalahan deteksi objek [12]. Dataset ini terdiri dari sebanyak 1464 gambar untuk *training*, 1449 gambar untuk validasi beserta *testing*, dan terdiri dari berbagai kategori seperti kendaraan bermotor, peralatan rumah seperti kursi dan meja makan, hewan, dan manusia.

Contohnya adalah pada Gambar 2.2 *grid cell* yang berwarna kuning mencoba untuk memprediksi keberadaan seseorang di dalam *cell* tersebut. Setiap *grid cell* memprediksi sebanyak  $B$  *bounding box* dan *confidence score* untuk kotak tersebut. *Confidence score* ini mencerminkan tingkat keyakinan yang dimiliki oleh model bahwa kotak tersebut memiliki objek dan nilai akurasi yang dimiliki oleh kotak hasil prediksi tersebut. Nilai *confidence score* tersebut adalah nilai Intersection over Union (IoU) yang merupakan nilai *overlap* antara kotak hasil prediksi dan *ground truth*, yang merupakan nilai koordinat *bounding box* pada dataset. *Confidence* didefinisikan sebagai  $\text{Pr}(\text{Object}) * \text{IoU}$ .

Setiap *bounding box* memiliki lima elemen prediksi:  $(x, y, w, h)$  dan skor *box confidence*. Nilai  $x$  koordinat titik tengah pada axis horizontal, nilai  $y$  adalah koordinat titik tengah pada axis vertikal, nilai  $w$  adalah lebar dari kotak pembatas, dan nilai  $h$  adalah tinggi dari kotak pembatas. Tingkat *confidence* tersebut merefleksikan seberapa besar kemungkinan kotak tersebut memiliki sebuah objek dan tingkat akurasi dari *bounding box* tersebut. Nilai  $(x, y, w, h)$  dinormalisasi terhadap lebar dan tinggi dari gambar sehingga nilai-nilai tersebut berada di antara 0 dan 1.

Setiap *grid cell* memprediksi sebanyak  $B$  *bounding box* dengan setiap *bounding box* memiliki satu *confidence score*,  $C$  *class probability*, yang dinotasikan dengan  $\text{Pr}(\text{Class}|\text{Object})$ . Probabilitas ini dikondisikan pada *grid cell* yang di dalamnya terdapat sebuah objek, dan hanya dilakukan satu set kalkulasi untuk setiap *grid cell*. Pada saat pengujian *conditional class probability* dan hasil prediksi *confidence* pada setiap kotak dikalikan, sehingga menghasilkan persamaan seperti berikut:

$$\text{Pr}(\text{Class}|\text{Object}) \times \text{Pr}(\text{Object}) * \text{IoU} = \text{Pr}(\text{Class}) \times \text{IoU} \quad (2.1)$$



Gambar 2.3. Arsitektur dari Model Deteksi Objek You Only Look Once [11]

Dalam menganalisa PASCAL VOC, YOLO menggunakan *grid*  $7 \times 7$ , dan dalam contoh Gambar 2.2 terdeteksi sejumlah 2 *boundary box*, dengan 20 kelas *conditional class probability* karena PASCAL VOC memiliki total sebanyak 20 kategori. Oleh karena itu, YOLO memiliki bentuk ukuran prediksi (S, S,  $B \times 5 + C$ ) yang akhirnya menjadi (7,7,30). YOLO menggunakan jaringan CNN untuk memprediksi tensor dengan cara mengurangi dimensi spasial ke  $7 \times 7$  dengan 1024 channel output pada setiap lokasi. YOLO melakukan regresi linear dengan menggunakan dua lapisan yang bersifat *fully connected* untuk membuat prediksi *boundary box*. Untuk dapat membentuk sebuah prediksi final, skor *confidence* pada *box* yang bernilai tinggi disimpan sebagai prediksi final.

Secara singkat, *box confidence score*, *conditional class probability*, dan *class confidence score* dapat disimpulkan sebagai berikut [11]:

$$\text{box confidence score} \equiv P_r(\text{object}) \cdot \text{IoU} \quad (2.2)$$

$$\text{conditional class probability} \equiv P_r(\text{class}_i | \text{object}) \quad (2.3)$$

$$\text{class confidence score} \equiv P_r(\text{class}_i) \cdot \text{IoU} \quad (2.4)$$

dimana  $P_r(\text{object})$  adalah probabilitas bahwa sebuah kotak di dalamnya memiliki sebuah objek, IoU adalah IoU (intersection over union) antara kotak hasil prediksi dan ground truth,  $P_r(\text{class}_i | \text{object})$  adalah probabilitas bahwa objek berada dalam kategori kelas  $\text{class}_i$ , dengan mensyaratkan bahwa terdapat sebuah objek, dan  $P_r(\text{class}_i)$  adalah probabilitas bahwa objek berada dalam kategori kelas tertentu.

Sampai saat ini sudah ada beberapa versi YOLO yang dikembangkan yaitu [13]:

YOLO original: YOLO merupakan network deteksi objek pertama yang mampu untuk mengkombinasikan permasalahan menggambarkan kotak *bounding box* dan mengidentifikasi label kelas di dalam sebuah network yang bersifat *end-to-end*.

YOLOv2: YOLOv2 melakukan pengembangan iteratif yang meliputi penambahan Batchnorm, resolusi yang lebih tinggi, dan anchor box.

YOLOv3: YOLOv3 dikembangkan lebih lanjut untuk menambahkan skor *objectness* kepada prediksi *bounding box*, menambahkan koneksi pada lapisan *backbone*, dan melakukan prediksi pada tiga tahap berbeda granularitas guna meningkatkan performanya pada objek yang lebih kecil.

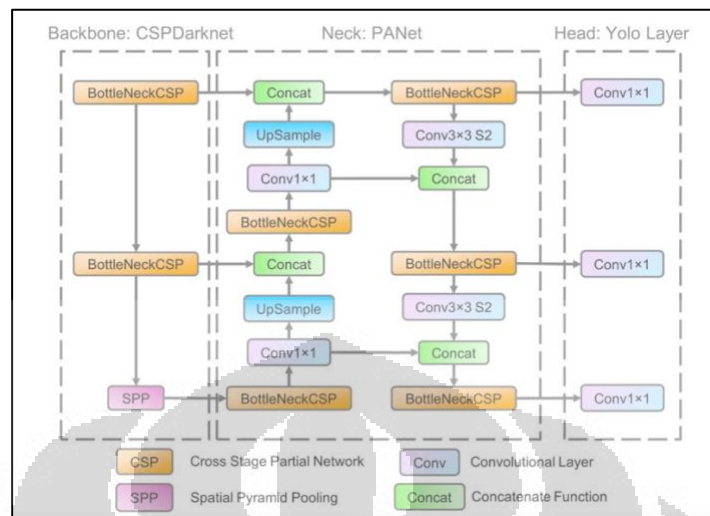
YOLOv4: YOLOv4 mengembangkan sebuah *backbone* baru yaitu CSPDarknet53 untuk diimplementasikan dan berhasil meningkatkan performa mAP dan juga waktu prediksi aplikasi.

YOLOv5: YOLOv5 adalah sebuah pengembangan relatif baru yang dirilis setelah riset mengenai YOLOv4 dipublikasikan di dalam *paper*. Perbedaan antara YOLOv5 dengan versi YOLO sebelumnya adalah YOLOv5 diimplementasikan di dalam Pytorch ketimbang Darknet.

## 2.4. YOLOv5

*You Only Look Once* v5 (YOLOv5) merupakan iterasi ke-5 dari YOLO, sebuah model machine learning yang digunakan untuk mendeteksi objek [10]. YOLOv5 adalah sebuah pengembangan dari model YOLO original. Mengikuti anatomi deteksi objek, arsitektur dari YOLOv5 memiliki struktur input, backbone, dan neck. Backbone biasanya pre-trained pada sebuah dataset klasifikasi gambar yang lebih besar, seperti ImageNet dan berguna untuk meng-encode informasi relevan mengenai input, yakni gambar yang digunakan. Head berguna untuk melakukan prediksi kelas objek dan informasi mengenai *bounding box*. Selain itu terdapat sebuah neck, yang mendefinisikan layer di antara backbone dan head yang berguna untuk mengkoleksi *feature map* dari tahap-tahapan berbeda pada *network*.

### 2.4.1. Arsitektur YOLOv5



Gambar 2.4. Ilustrasi Arsitektur Backbone, Neck, dan Head YOLOv5 [14]

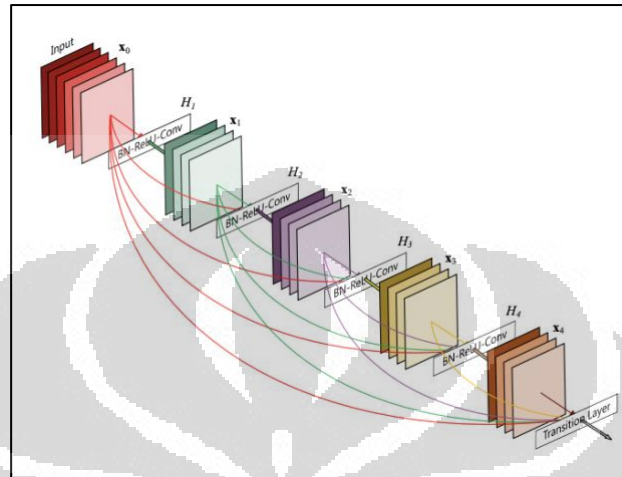
Berikut pada Gambar 2.4 adalah ilustrasi dari arsitektur YOLOv5 yang digunakan untuk penelitian dan pelatihan model sistem deteksi objek. YOLOv5 secara umum menggunakan CSP Bottleneck atau CSP Darknet untuk memformulasikan fitur gambar pada bagian *backbone*. Pada bagian berikutnya YOLOv5 menggunakan PANet sebagai *neck* untuk bagian agregasi fitur, dan mengikuti YOLO versi sebelumnya, YOLOv5 menggunakan YOLO Layer untuk bagian *head*.

Backbone dari sebuah jaringan detektor objek biasanya pre-trained pada klasifikasi ImageNet. Dengan adanya pre-training ini maka *weight* pada jaringan telah diadaptasi pada fitur-fitur yang relevan pada sebuah gambar [10]. Meskipun demikian, *weight* pada jaringan ini akan berubah lagi ketika dilakukan deteksi objek. Pada YOLOv5, dilakukan pemilihan dari antara ketiga backbone berikut, yakni:

- CSPResNext50
- CSPDarknet53
- EfficientNet-B3

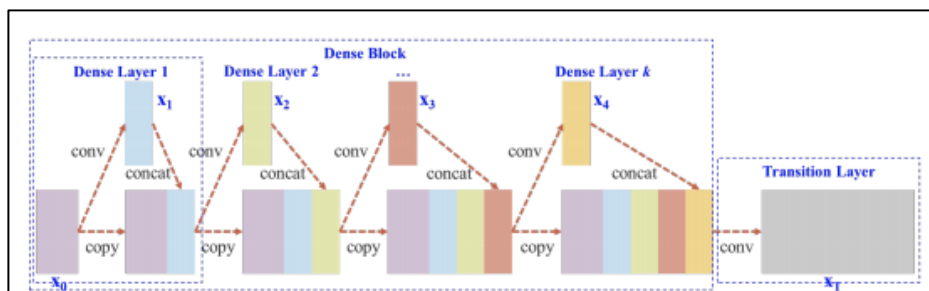
CSPResNext50 dan CSPDarknet53 keduanya didasari atas DenseNet. DenseNet didesain untuk menghubungkan lapisan pada CNN dengan tujuan untuk menghilangkan permasalahan *vanishing gradient*. Ini merupakan permasalahan

dimana sulit untuk dilakukan backwards propagation pada sebuah jaringan yang sangat dalam. Selain itu, DenseNet juga bertujuan untuk meningkatkan propagasi fitur, mendorong jaringan untuk menggunakan fitur kembali, dan mengurangi jumlah dari parameter jaringan [15].

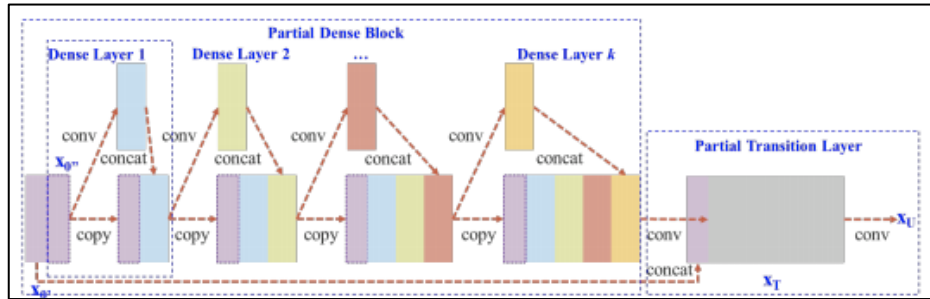


Gambar 2.5. Sebuah Ilustrasi Dense Block. DenseNet Mengambil Semua Feature Map yang Mendahului Sebagai Input [15]

CSP atau *Cross Stage Partial* merupakan pengembangan baru yang dilakukan di atas DenseNet. CSP memitigasi permasalahan yang membutuhkan komputasi inferensi yang berat dari perspektif arsitektur jaringan. Di dalam CSPResNext50 dan CSPDarknet53, DenseNet telah dimodifikasi untuk memisahkan map fitur pada lapisan dasar dengan menyalinnya kemudian mengirimkannya melalui blok dense dan mengirim satu lagi langsung kepada tahap berikutnya [16]. Dengan dilakukannya hal ini maka CSPResNext50 dan CSPDarknet53 mampu untuk menghilangkan *bottleneck* komputasi pada DenseNet dan meningkatkan proses *learning* dengan melakukan *passing* dari map fitur yang bersifat tidak diubah.



Gambar 2.6. Ilustrasi DenseNet Normal [16]

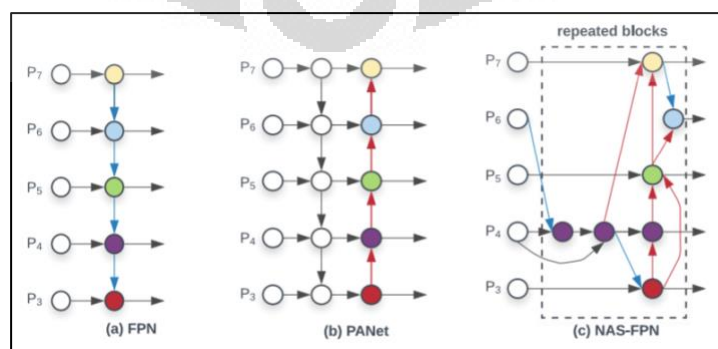


Gambar 2.7. Ilustrasi DenseNet dengan Modifikasi CSP (Cross Stage Partial) [16]

Pada Gambar 2.7 terlihat bahwa terdapat sebuah salinan dari fitur map yang dipass ke tahap selanjutnya sedangkan pada Gambar 2.6 tidak terjadi hal tersebut. Hal tersebut yang membedakan CSP DenseNet dan DenseNet biasa.

Penelusuran Efficientnet sebagai backbone tidak dilakukan kembali karena meskipun memiliki performa baik untuk klasifikasi gambar, namun penelitian oleh tim pengembang YOLO menyimpulkan bahwa terdapat model yang lebih baik untuk kasus deteksi objek yang digunakan oleh YOLO [10]. Pada akhirnya, dipilih model CSPDarknet53 sebagai *backbone* untuk model YOLOv5.

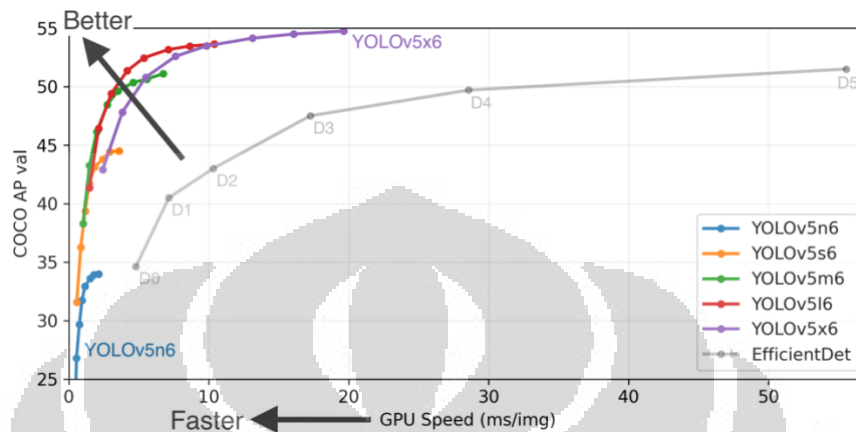
Pada bagian neck YOLOv5 menggunakan PANet untuk agregasi fitur pada jaringan. PANet digunakan untuk meningkatkan kualitas proses *instance segmentation* dengan cara melakukan preservasi dari informasi spasial [17]. Dibandingkan dengan FPN yang terbatas dengan flow informasi yang hanya berjalan satu arah, PANet menambahkan sebuah jalur agregasi jaringan dari bawah ke atas atau dikenal dengan istilah *bottom-up*. Dibandingkan dengan NAS-FPN, dan FPN, PANet memiliki tingkat akurasi yang lebih tinggi ketimbang yang lain, meskipun demikian memiliki parameter dan harga komputasi yang lebih tinggi [18].



Gambar 2.8. Ilustrasi Perbedaan FPN, PANet, dan NAS-FPN [18]

Terakhir, bagian *head* yang digunakan di dalam arsitektur YOLOv5 adalah sebuah *anchor-based head* yang digunakan dalam *head* YOLOv4 dan YOLOv3.

#### 2.4.2. Benchmark YOLOv5



Gambar 2.9. Perbandingan Berbagai Versi YOLOv5 dengan Algoritma EfficientDet

EfficientDet adalah algoritma deteksi objek satu tahap yang dikembangkan oleh Google dan mendapatkan julukan sebagai algoritma deteksi objek yang memiliki performa yang paling baik sebelum ada perkembangan YOLOv5. Untuk dapat melakukan perbandingan yang adil di antara algoritma deteksi objek yang berbeda, digunakan standar Average Precision atau AP dan performa algoritma deteksi objek dibandingkan setelah dilatih pada dataset COCO. Average Precision adalah sebuah metrik yang digunakan untuk mengukur presisi pada setiap ambang batas atau *threshold* tertentu. Setelah dilakukan pengembangan dan rilis dari algoritma YOLOv5, perbandingan AP yang dilakukan terhadap COCO Dataset menunjukkan bahwa versi YOLOv5 mampu untuk melakukan deteksi dengan lebih cepat dan dengan akurasi yang lebih tinggi dibandingkan EfficientDet.

Terlihat bahwa dalam Gambar 2.9 terdapat kurva berwarna oranye yang menggambarkan performa algoritma deteksi objek yang digunakan di dalam penelitian ini, yaitu algoritma YOLOv5s. Terlihat bahwa performa AP dari algoritma tersebut sebanding dengan versi EfficientDet-D2 pada saat kecepatan GPU dilambatkan. Selain itu terlihat juga bahwa terdapat versi EfficientDet-D0 dan EfficientDet-D1 yang memiliki performa AP yang berada di bawah performa YOLOv5s dengan kecepatan GPU yang lebih lambat.

Perbandingan yang dilakukan antara kedua sistem deteksi objek *state of the art* menunjukkan bahwa YOLOv5 dapat melakukan proses deteksi objek dengan lebih baik dengan performa lebih cepat dibandingkan EfficientDet. Sehingga, untuk proses pengembangan algoritma sistem deteksi objek di dalam penelitian ini digunakan penggunaan algoritma YOLOv5.

## 2.5. Tolak Ukur Evaluasi Model

### 2.5.1. Precision dan Recall

Precision adalah metode pengukuran untuk mengetahui tingkat akurasi model dalam menemukan true positive (TP) dari semua prediksi positif. Semua prediksi positif berarti prediksi positif yang sesuai dengan data dan prediksi positif yang tidak sesuai dengan data. Recall adalah ukuran untuk mengetahui tingkat true positive dari semua prediksi yang bisa dilakukan. [19]. Semua prediksi yang dapat dilakukan terdiri dari semua prediksi positif yang sesuai dengan data serta prediksi negatif yang seharusnya positif. Berikut adalah definisi dari precision dan recall secara matematis:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2.5)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2.6)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (2.7)$$

True positive berarti berhasil memprediksi dengan benar, false negative berarti gagal untuk memprediksi objek yang ada. False positive berarti memprediksi sebagai positif padahal seharusnya tidak positif. True negative berarti nilai yang tidak positif dan seharusnya tidak positif.

### 2.5.2. Intersection over Union

IoU merupakan pengukuran terhadap nilai *overlap* di antara dua boundary. Ini digunakan untuk dapat mengetahui nilai dari pembatas hasil prediksi overlap



dengan nilai aslinya, yang disebut *ground truth* [7]. Berikut adalah rumus yang digunakan untuk mengkalkulasikan IoU.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} \quad (2.8)$$

IoU adalah kalkulasi dari *area of overlap* yang berarti daerah dimana keduanya beririsan, dibagi dengan *area of union* yang berarti piksel yang dapat ditemukan di daerah hasil prediksi ataupun *ground truth*.

Untuk permasalahan deteksi objek, kalkulasi dari precision dan recall ditentukan berdasarkan pada nilai IoU untuk sebuah threshold atau nilai pembatas tertentu. Apabila nilai IoU dari hasil prediksi lebih tinggi dibandingkan dengan nilai IoU threshold, maka prediksi tersebut dapat diklasifikasikan sebagai True Positive atau TP. Sedangkan apabila nilai IoU hasil prediksi lebih rendah dibandingkan dengan IoU threshold maka prediksi tersebut diklasifikasikan sebagai False Positive atau FP.

### 2.5.3. Mean Average Precision

Average precision didefinisikan sebagai rata-rata ukuran presisi yang berbobot pada setiap ambang batas atau *threshold*. Bobot yang digunakan adalah peningkatan dalam recall dibandingkan bobot sebelumnya. Cara untuk mendapatkan nilai Average precision adalah dengan mencari luas dari daerah di bawah kurva precision-recall. Mean Average Precision atau mAP didapatkan dari kalkulasi rata-rata AP pada semua kelas yang ada pada threshold IoU tertentu. Kalkulasi untuk mAP dilakukan dengan rumus berikut:

$$\frac{1}{N} \sum_{i=1}^N AP_i \quad (2.9)$$

Dalam rumus tersebut N adalah jumlah kelas yang ada dan AP adalah nilai Average precision dari masing-masing kelas tersebut. Dalam PASCAL VOC2007 challenge, AP untuk satu kelas objek dikalkulasi untuk threshold IoU 0,5. Sehingga mAP dirata-rata antara semua objek kelas. Untuk COCO 2017 challenge, mAP dikalkulasi antara semua kategori objek dan 10 threshold IoU.

## 2.6. Platform Pengembangan yang Digunakan

### 2.6.1. Flutter

Flutter adalah sebuah framework *open source* yang diciptakan oleh Google untuk membangun aplikasi *multi-platform* yang dikompilasi secara *native* dengan tampilan UI yang menarik menggunakan satu basis kode. Flutter dapat digunakan untuk mengembangkan aplikasi yang berjalan pada Android, iOS, Linux, macOS, Windows, Google Fuschia, dan pada web. Berbeda dengan aplikasi yang dikembangkan secara *native*, Flutter memungkinkan fleksibilitas untuk mengembangkan aplikasi untuk jangkauan pasar yang lebih luas dengan satu jenis bahasa pemrograman.

Aplikasi Flutter dikembangkan dalam bahasa pemrograman khusus yang bernama Dart. Flutter menggunakan *widget* yang dapat digunakan untuk membangun aplikasi yang berfungsi secara lengkap. Secara umum, terdapat tiga jenis *widget* di dalam Flutter, yakni *Stateful widget*, *Stateless widget*, dan *Inherited widget*.

Widget di dalam Flutter dikembangkan dengan mengambil inspirasi dari React, yang merupakan sebuah library Javascript yang digunakan untuk membuat user interface dari sebuah website. Ide utama dari widget di dalam Flutter adalah bahwa UI yang dirancang dibangun dengan widget-widget tertentu. Dengan menggunakan widget, Flutter mampu untuk mendefinisikan tampilan yang diberikan kepada pengguna aplikasi berdasarkan konfigurasi dan state tertentu. Saat state dari sebuah widget berubah, maka widget tersebut melakukan proses yang dinamakan rebuilding, yang dibandingkan oleh framework Flutter dengan state sebelumnya untuk melakukan perubahan yang diperlukan akibat perubahan state tersebut.

### 2.6.2. Google Firebase

Firebase adalah *Backend-as-a-Service* atau BaaS yang menyediakan layanan *backend* pada umumnya yang dapat digunakan untuk membangun sebuah aplikasi. Layanan *backend* yang tersedia pada Firebase antara lain adalah layanan

autentikasi, penyimpanan data, penyimpanan gambar pada server awan, dan sebagainya.

Layanan Firebase yang digunakan dapat mempercepat proses pembangunan aplikasi dengan menyediakan fungsionalitas yang lazim digunakan beserta dengan layanan penyediaan sebuah server untuk penyimpanan informasi. Cara kerja dari penggunaan Google Firebase adalah dengan integrasi sertifikasi ke dalam aplikasi dan penggunaan *package* untuk Firebase sehingga aplikasi mendapatkan akses kepada *console* Firebase akun tersebut.

### 2.6.3. Google Cloud Platform

Google Cloud Platform atau GCP adalah sebuah platform yang menyediakan jasa atau layanan untuk menjalankan kode pada server awan. Dengan melakukan hal tersebut, akses terhadap kode tersebut tidak lagi terbatas pada cakupan lokal saja, namun dapat diakses melalui internet. GCP dapat digunakan sebagai tempat untuk menjalankan sebuah layanan tertentu sehingga fungsionalitas layanan tersebut tersedia untuk digunakan oleh aplikasi lainnya.

Di dalam Google Cloud Platform terdapat banyak layanan yang dapat digunakan untuk proses *deployment* dari kode pemrograman ke server awan. Salah satunya yang digunakan di dalam penelitian ini adalah Cloud Run. Dengan layanan Cloud Run aplikasi dapat dimasukkan ke dalam sebuah wadah atau lebih dikenal dengan istilah *container* sehingga dapat melalui proses *deployment* ke server yang dikelola secara *serverless*. Ini berarti bahwa penggunaan server secara langsung tidak dilakukan, dan proses alokasi server dilakukan dan dikelola oleh *platform* GCP.

### 2.6.4. Google Colab Pro

Google Colab Pro adalah sebuah layanan versi berbayar dari Google Colab yang disediakan oleh Google untuk melakukan *training* proses *machine learning* secara online. Google Colab Pro merupakan dokumen yang dapat dieksekusi yang dapat digunakan untuk menjalankan program Python tanpa memerlukan proses setup dan instalasi. Setiap hasil sesi Google Colab Pro yang digunakan kemudian

dapat tersimpan di dalam Google Drive dari akun yang digunakan layaknya sebuah dokumen biasa.

Tabel 2.1. Spesifikasi Perangkat yang Disediakan saat Menggunakan Google Colab Pro

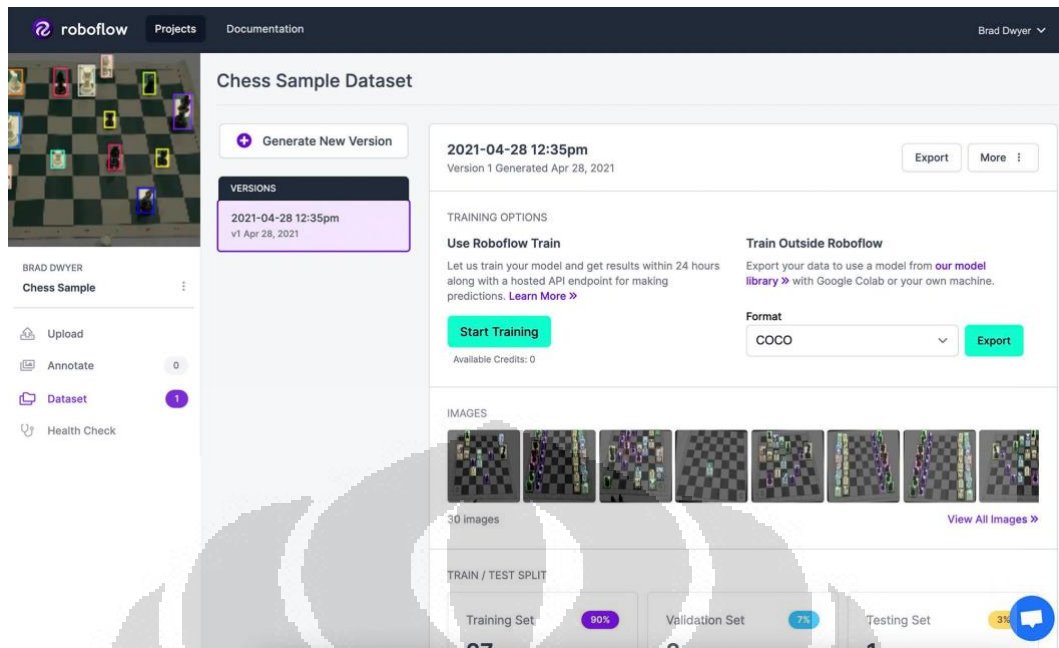
| Fitur                          | Spesifikasi Perangkat |
|--------------------------------|-----------------------|
| Graphics Processing Unit (GPU) | K80, P100, T4         |
| Central Processing Unit (CPU)  | 2 x vCPU              |
| Random Access Memory (RAM)     | 32 GB                 |

Dengan menggunakan Google Colab Pro, terdapat peningkatan ketersediaan jenis Graphics Processing Unit (GPU) yang disediakan sehingga mampu untuk mempercepat proses *training* model. Dengan sifat Google Colab Pro yang dapat digunakan dengan mengakses jaringan internet, platform ini memberikan fleksibilitas terhadap penyewaan sumber daya secara online sehingga tidak membutuhkan perangkat fisik.

#### 2.6.5. Roboflow

Roboflow adalah sebuah platform yang digunakan untuk mengorganisir, menganotasi, dan mempersiapkan gambar untuk digunakan pada proses pelatihan model *machine learning*. Proses *dataset preprocessing* ini sepenuhnya dapat dilakukan di dalam browser dengan peralatan yang disediakan oleh Roboflow secara langsung pada website.

Roboflow menyediakan dataset yang dapat digunakan secara langsung serta menyediakan layanan untuk melakukan pembentukan dataset baru. Untuk menunjang proses pembentukan dataset baru, di dalam Roboflow dapat dilakukan proses penyimpanan dataset yang belum dianotasi, tools untuk anotasi gambar, serta proses ekspor dataset ke dalam format tertentu. Dengan layanan *hosting* dari dataset data yang diupload ke platform Roboflow akan tersimpan pada akun Roboflow tersebut.



Gambar 2.10. Tampilan Roboflow yang Digunakan Sebagai Platform untuk Merancang Dataset

## 2.7. User Experience Questionnaire (UEQ)

Kuisiometer User Experience atau UEQ digunakan untuk melakukan pengukuran terhadap kualitas penggunaan sebuah produk, atau dalam hal ini, sebuah aplikasi. Untuk dapat menilai kualitas sebuah produk, dapat digunakan dua nilai kualitas produk yang terkait pada user experience, yaitu usability goals dan user experience goals [20]. Usability goals adalah kualitas interaksi yang terkait dengan hal-hal yang perlu dilakukan oleh seorang user untuk mencapai tujuan mereka. Sedangkan, user experience goals adalah kualitas interaksi yang terkait dengan perasaan subjektif saat melakukan interaksi dengan produk.

Kuisiometer UEQ dibangun dengan memperhatikan usability goals dan user experience goals dari sebuah produk. Dalam kuisiometer ini pertanyaan dikonstruksi secara empiris dengan 6 skala yang merepresentasikan aspek terpenting dari UX untuk berbagai jenis produk. Nilai pada Kuisiometer User Experience ini dinilai dengan menggunakan skala 7 poin yang terdiri dari sepasang istilah dengan arti berlawanan yang memiliki rentang dimensi semantik di antara kedua istilah tersebut. Terdapat 6 poin utama yang diuji di dalam UEQ yakni:

1. Daya tarik: Impresi utama pengguna saat menggunakan produk, tingkat kesukaan atau tidak menggunakan produk.
2. Kejelasan: Impresi pengguna mengenai kemudahan untuk memahami cara untuk menggunakan produk.
3. Efisiensi: Impresi pengguna bahwa mencapai tujuan dengan menggunakan produk dapat dilakukan dengan cepat dan efisien, dengan organisasi interface yang jelas.
4. Ketepatan: Perasaan pengguna mengenai aspek keamanan dan kontrol saat menggunakan produk.
5. Stimulasi: Impresi pengguna saat sesuatu menarik dan menyenangkan untuk menggunakan produknya.
6. Kebaruan: Impresi pengguna bahwa desain produk termasuk inovatif, kreatif, dan menangkap perhatian.

Format kuisioner ini mendukung peserta yang mengisi kuisioner untuk mengutarakan perasaan, impresi, serta sikap yang muncul ketika menggunakan aplikasi [21]. Penggunaan kuisioner ini dilakukan atas dasar pengujian keandalan dan validitas dari kuisioner yang dilakukan oleh riset yang mendalam pada UEQ.

## 2.8. Aplikasi Phoodto

Phoodto adalah sebuah aplikasi yang dikembangkan sebagai bagian dari *capstone project* yang dikerjakan pada program Bangkit 2021. Pengembangan aplikasi ditujukan untuk dapat memberikan *feedback* kesehatan kepada orang-orang yang menggunakan aplikasi dan memiliki keterbatasan makanan karena disebabkan alergi ataupun faktor kesehatan. Tim Phoodto kemudian mendapatkan inkubasi untuk mengembangkan aplikasinya secara lebih lanjut [22]. Cara kerja dari aplikasi yang dikembangkan adalah gambar yang diupload akan memberikan hasil *feedback* mengenai makanan tersebut.

Metode yang digunakan di dalam aplikasi tersebut adalah metode klasifikasi gambar atau *image classification*. Metode tersebut memiliki keterbatasan yaitu ketika dilakukan pengembangan lebih lanjut proses deteksi yang mampu dilakukan oleh aplikasi terbatas pada satu kategori per gambar saja. Proses pengembangan

selanjutnya yang dapat dilakukan adalah mengubah algoritma *machine learning* menjadi menggunakan metode deteksi objek, sehingga proses pendeteksian bisa dilakukan secara lebih mendetil dan spesifik pada suatu gambar makanan.

## **2.9. Penelitian Terdahulu**

### **2.9.1. BTBU-60: *dataset for object detection in food field* [23]**

Dalam riset ini dilakukan pengembangan dataset untuk objek makanan yang bersifat makanan sehari-hari. Dataset objek makanan yang dikembangkan dibandingkan dengan dataset makanan yang sudah tersedia dalam riset lain seperti Food-101, UEC Food-100, dan PFID. Setiap dataset yang tersedia ini memiliki variabilitas di dalam ukuran dataset maupun daftar kategori makanan yang telah diberikan label.

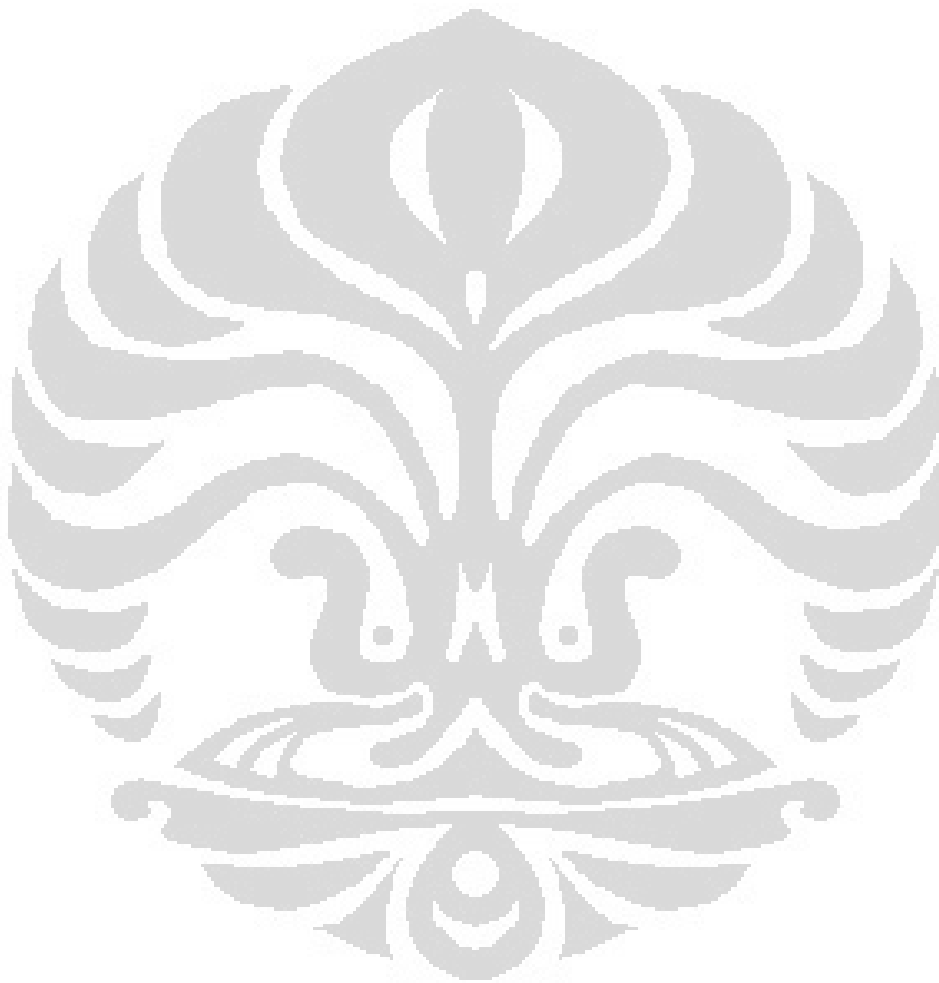
Dalam penelitian ini dilakukan pengembangan dataset makanan yang terdiri dari 60k gambar dan 78k komponen atau instance di dalam gambar-gambar tersebut. Pengembangan dari dataset tersebut dilakukan oleh 30 mahasiswa S2 dengan menggunakan sistem labelling otomatis untuk klasifikasi gambar dan deteksi objek. Pengembangan algoritma dilakukan dengan menggunakan Faster R-CNN dan menguji perbedaan performa dengan arsitektur ekstraksi fitur yang berbeda pada dataset BTBU-60.

### **2.9.2. *Real-time Food-Object Detection and Localization for Indian Cuisines using Deep Neural Networks* [24]**

Riset food-object detection ini dilakukan pada makanan India dan terdiri dari 60 kelas yang memiliki total sebanyak 4200 gambar dimana setiap kelas memiliki 70 sampel. Data yang digunakan untuk penelitian ini menggunakan berbagai sumber online dan diberikan label secara manual. Penelitian ini menggunakan MobileNet\_V2 dan Inception\_V2 sebagai arsitektur untuk menguji dataset yang telah dikembangkan.

Kedua algoritma tersebut dibandingkan satu sama lain dengan menggunakan SSD karena lebih praktis dari segi ukuran algoritma dan efisiensi.

Hasil dari riset adalah model deteksi objek makanan yang dikembangkan kemudian dibandingkan antara hasil SSD MobileNet V2 dan SSD Inception V2. Akurasi dari model SSD yang dihasilkan mencapai 97,6% dengan tingkat confidence di atas 95% untuk 1750 dari 2000 gambar yang diuji.





## BAB 3

### PERANCANGAN SISTEM

#### 3.1. System Requirement

Penelitian ini bertujuan untuk dapat menghasilkan suatu aplikasi yang dapat mendeteksi objek makanan dalam sebuah gambar dan memberikan *feedback* kepada user dengan syarat seperti berikut:

1. Sistem dapat mendeteksi komponen objek makanan dalam sebuah gambar dengan tingkat akurasi prediksi yang tinggi.
2. Sistem dapat di-*deploy* dan dijalankan pada *platform* komputasi awan.
3. Sistem dapat diaplikasikan untuk deteksi objek makanan dalam implementasi deteksi dalam sebuah aplikasi mobile.

Sistem deteksi objek makanan dalam penelitian ini dibuat untuk mengenali sejumlah kelas objek makanan yang berbeda. Model deteksi objek makanan ini diimplementasikan pada sebuah *server* yang berfungsi sebagai perangkat yang akan melakukan pemrosesan deteksi objek makanan dari gambar yang didapatkan dari pengguna aplikasi. Pengguna dapat melakukan pengambilan gambar melalui aplikasi Phoodto kemudian mengirimnya ke *server* yang akan melakukan perhitungan berdasarkan model yang sudah dilatih untuk mendeteksi objek makanan tertentu.

Model deteksi objek makanan yang diimplementasikan harus memiliki waktu respons yang cepat. Hal ini ditujukan terutama untuk meningkatkan kualitas penggunaan aplikasi yang akan semakin lancar apabila hasil diagnosa terhadap jenis makanan yang difoto dapat diperoleh dengan lebih cepat.

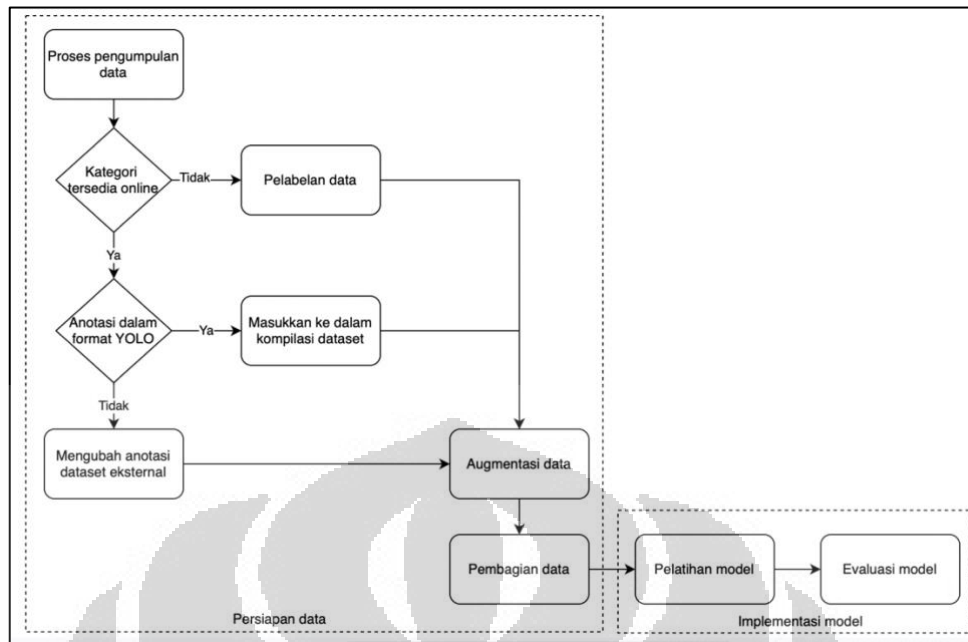
#### 3.2. Metode Penelitian

Penelitian ini dimulai dengan mengumpulkan data berupa gambar makanan sehari-hari yang akan digunakan untuk pelatihan model deteksi objek. Gambar yang dikumpulkan kemudian perlu melalui proses pelabelan agar model dapat mengidentifikasi dan mempelajari fitur-fitur yang dimiliki oleh gambar tersebut.

Data diperoleh secara mandiri melalui berbagai sumber online dan diberikan label dengan menggunakan tools anotasi online bernama Roboflow. Gambar yang telah diberikan label berdasarkan kategori dan diformat dalam bentuk YOLOv5 Pytorch Ultralytics. Dataset yang sudah diberikan label kemudian melalui proses augmentasi data untuk menambahkan jumlah gambar di dalam dataset sehingga dapat meningkatkan kualitas model. Dataset kemudian dilatih menggunakan algoritma YOLOv5s yang dikembangkan dalam Pytorch. Pelatihan model dilakukan pada platform Google Colab Pro yang menggunakan spesifikasi GPU Tesla T4 dan kapasitas memori hingga 16 GB.

Model pendeteksi objek yang dilatih pada Google Colab kemudian dievaluasi berdasarkan berbagai metrik yang dihasilkan oleh model deteksi objek. Metrik yang digunakan untuk mengevaluasi model antara lain adalah Precision, Recall, dan Mean Average Precision (mAP). Nilai dari MAP dikalkulasi berdasarkan luas wilayah di bawah Kurva Precision-Recall dari setiap kategori yang ada di dalam dataset. Nilai tersebut kemudian dirata-rata antara semua kategori yang ada dan menghasilkan sebuah metrik bernama Mean Average Precision. Metrik ini menjadi panduan untuk mengevaluasi hasil model yang telah dilatih. Apabila metode yang digunakan telah mencapai hasil akurasi yang ditargetkan, maka model akan digunakan untuk melakukan prediksi.

Untuk dapat mengimplementasikan model YOLOv5 Ultralytics dalam aplikasi, model harus dijalankan dengan menggunakan Python pada sebuah server sehingga dapat menerima gambar dan memberikan hasil prediksi sesuai dengan fungsionalitas sebuah Application Programming Interface (API). Aplikasi kemudian akan memberikan feedback atas gambar yang diunggah dan diproses oleh model machine learning sehingga dapat menghasilkan prediksi yang akurat dalam waktu inferensi yang cukup rendah.



Gambar 3.1. Rancangan Perancangan Sistem Deteksi Objek dengan YOLOv5 Ultralytics

### 3.3. Pengumpulan Data

Pengumpulan data untuk pembentukan model deteksi objek makanan didasarkan atas ketersediaan gambar yang belum diberikan label secara online. Proses pelabelan data sepenuhnya dilakukan secara mandiri dan tanpa menggunakan sumber dataset yang telah diberikan label dan tersedia secara online. Latar belakang pelabelan data secara mandiri adalah karena kurangnya ketersediaan data gambar makanan Indonesia yang telah diberikan label yang tersedia di sumber online. Oleh karena itu, proses pemilihan kategori data berdasarkan tipe-tipe makanan Indonesia yang sering dijumpai dan makanan asing yang sering dimakan oleh orang Indonesia. Setelah terjadi pemilihan kategori yang akan dikumpulkan di dalam dataset, dilakukan pengumpulan data secara bertahap.

Sumber data utama adalah data gambar yang diperoleh dengan *data scraping* melalui sumber online Google Images. Data yang didapatkan dari sumber online ini dikompilasi ke struktur folder yang terdiri dari kategori makanan yang dikumpulkan untuk proses pelabelan data. Prosedur yang dilakukan adalah mencari gambar dari daftar kategori yang telah ditentukan pada search engine Google. Gambar yang didapatkan dari sumber online tersebut kemudian akan diseleksi kembali untuk melalui proses berikutnya yakni pelabelan data. Proses ini dilakukan

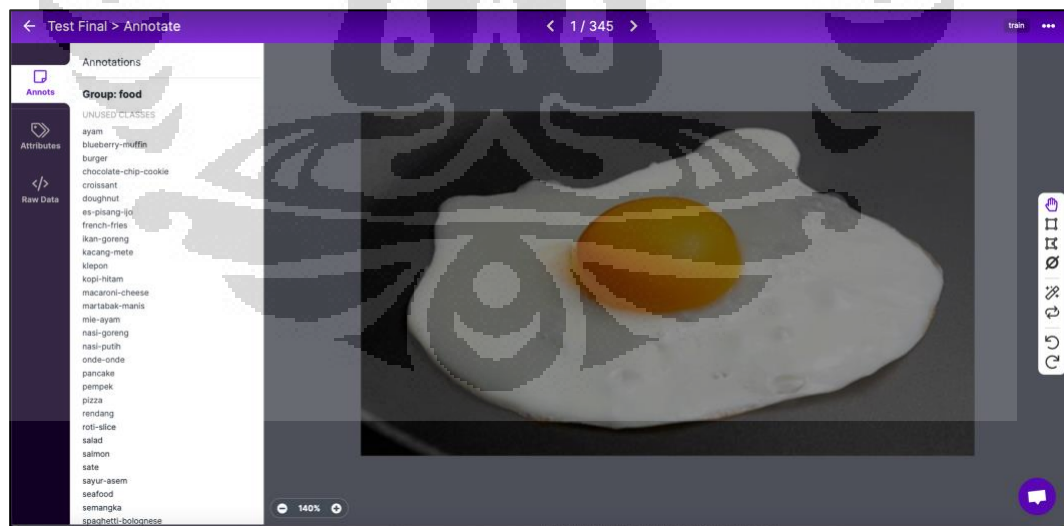
karena untuk pembentukan dataset yang baik diperlukan kualitas gambar makanan yang jelas dan dapat dibedakan antara satu kategori dengan kategori lain.

Setelah melalui tahap pencarian dan tahap seleksi dari data yang akan digunakan selama proses training, maka tahap berikutnya adalah melakukan pelabelan data.

### 3.4. Data Preprocessing dan Pelabelan Data

Untuk dapat mempersiapkan data yang dikumpulkan untuk training dengan menggunakan YOLOv5 Ultralytics, maka perlu dilakukan pemberian label anotasi pada gambar awal yang dikumpulkan melalui sumber online. Setelah dilakukan pelabelan data perlu dilakukan export model dalam bentuk YOLOv5 Pytorch sehingga dapat dilatih dalam framework YOLOv5 Ultralytics.

Pelabelan data dilakukan dalam rangka membatasi objek di dalam sebuah *bounding box* yang akan membantu model untuk mempelajari fitur dari sebuah objek dan dapat melakukan klasifikasi objek tersebut berdasarkan hasil pembelajarannya. Pelabelan data pada gambar dilakukan secara manual dengan menggunakan platform Roboflow.



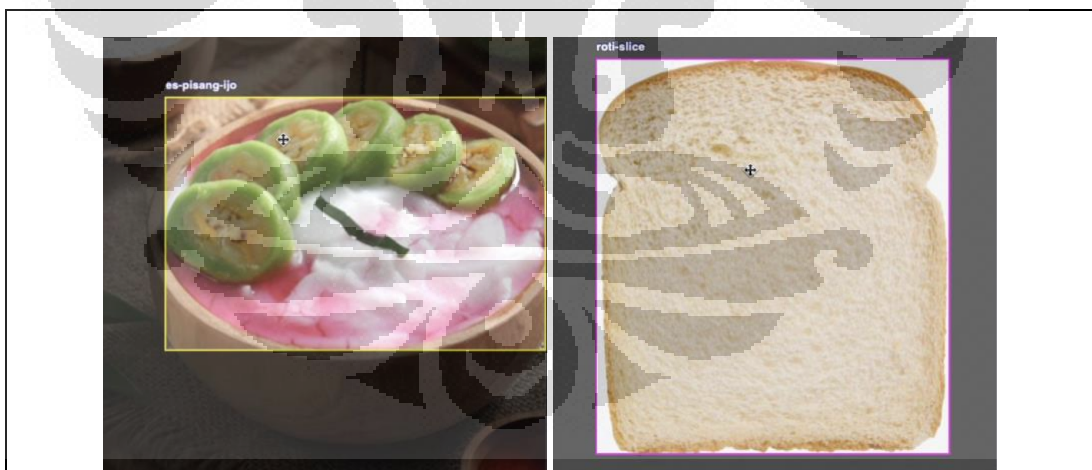
Gambar 3.2. Tampilan Graphical User Interface (GUI) dari Platform Roboflow untuk Pelabelan Data

Roboflow adalah sebuah platform yang digunakan untuk pengembangan model pada tahap proses pengumpulan data. Roboflow dapat digunakan untuk melakukan hosting gambar yang belum diberikan label atau dianotasi dan dapat

diberikan label dengan menggunakan *tools* pelabelan data. Data yang telah dikumpulkan melalui sumber online diupload ke platform Roboflow kemudian diberikan label dengan menggunakan *bounding box* untuk setiap kelas yang dapat diprediksi.

Teknik yang diambil dalam proses pelabelan data adalah dengan melakukan pengumpulan gambar yang hanya memiliki satu objek tertentu di dalam sebuah gambar, sehingga mayoritas gambar terfokuskan pada satu jenis kategori makanan saja. Pendekatan ini diambil karena dianggap bahwa akan mempermudah proses deteksi objek untuk mengenal setiap kategori makanan dengan fitur-fitur yang lebih detil dan jelas dibandingkan jika di dalam setiap gambar terdapat banyak komponen kategori makanan yang berukuran kecil.

Untuk deteksi objek YOLOv5 Pytorch disyaratkan bahwa label dari data disimpan dalam format khusus yang telah ditentukan. Format tersebut adalah dalam format file .txt mengikuti format <object-class> <x> <y> <width> <height> untuk setiap gambar. Format tersebut akan menghasilkan gambar dengan anotasi seperti berikut:



Gambar 3.3. Contoh Gambar yang Telah Diberikan Label Data Sesuai dengan Kategori Objek Makanan dalam Roboflow

Untuk setiap objek pada gambar tersebut, maka terdapat sebuah baris baru di dalam file .txt tersebut. Kelas objek berupa angka integer yang merepresentasikan kelas tertentu di dalam daftar kelas yang berada di dalam file .txt terpisah berisi semua nama kelas yang diprediksi oleh model deteksi objek.

Koordinat x dan y merupakan koordinat titik tengah dari kotak pembatas yang telah dinormalisasi sehingga bernilai angka desimal antara 0 dan 1. Koordinat lebar atau *width* dan tinggi atau *height* merupakan ukuran lebar dan tinggi dari kotak pembatas yang telah dinormalisasi menjadi angka desimal antara 0 dan 1.

### 3.5. Augmentasi dan Distribusi Data

Untuk dapat meningkatkan variasi dari dataset dan meningkatkan kualitas pada saat *training*, maka dilakukan beberapa tahapan augmentasi data. Augmentasi data dapat meningkatkan generalisasi kinerja model dengan meningkatkan keragaman contoh pembelajaran untuk model. Oleh karena itu augmentasi data penting untuk digunakan dalam melakukan pelatihan model deteksi objek. Jenis augmentasi yang dilakukan pada dataset yang digunakan adalah sebagai berikut:

1. Flip

Melakukan flip pada gambar adalah melakukan pembalikan pada gambar sehingga seolah-olah terdapat sebuah gambar baru yang sebenarnya merupakan gambar lama yang diputar dengan menggunakan teknik augmentasi data.

2. Shear

Shear adalah melakukan distorsi pada gambar pada axis horizontal maupun axis vertikal.

3. Exposure

Melakukan augmentasi exposure adalah mengubah gamma exposure dari sebuah gambar untuk menjadi lebih terang atau menjadi lebih gelap.

Dari data yang didapatkan, dataset akhir memiliki distribusi data yang dibagi secara 80% di bagian data untuk training serta 20% data untuk testing atau validasi. Selain membagi data tersebut dihasilkan juga sebuah file *train.txt* dan *test.txt*. Proses pembagian ini ditujukan untuk dapat memiliki gambar untuk training sekaligus gambar untuk validasi agar model yang dilatih dapat dievaluasi dengan mAP.



Dalam proses pelatihan model perlu dipastikan bahwa konfigurasi dari model untuk pelatihan sudah sesuai dengan ketentuan yang ditetapkan, yakni dengan menggunakan YOLOv5s. Untuk dapat melakukan konfigurasi tersebut maka perlu dilakukan pengaturan konfigurasi untuk penggunaan arsitektur YOLOv5s seperti sebagai berikut:

```
# parameters
nc: 40 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, BottleneckCSP, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, BottleneckCSP, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, BottleneckCSP, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]],
  [-1, 3, BottleneckCSP, [1024, False]], # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, BottleneckCSP, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, BottleneckCSP, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, BottleneckCSP, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, BottleneckCSP, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]
```

Gambar 3.5. Perubahan Konfigurasi yang Dilakukan pada YOLOv5s



Konfigurasi yang dilakukan pada Gambar 3.5 sesuai dengan arsitektur YOLOv5s yang dipilih untuk menjadi arsitektur model deteksi objek makanan. Secara umum, konfigurasi pada Gambar 3.5 adalah interpretasi dari Gambar 3.4 yang dapat digunakan oleh YOLOv5 Pytorch untuk mengetahui bahwa arsitektur yang digunakan adalah YOLOv5s. Yang mengindikasikan bahwa konfigurasi model pada Gambar 3.5 merupakan konfigurasi YOLOv5s adalah nilai konstanta variabel *depth\_multiple* dan *width\_multiple* yang memiliki nilai masing-masing 0,33 dan 0,50.

Pada saat pelatihan model, training dapat dihentikan saat terlihat bahwa setiap epoch terdapat peningkatan dari hasil mAP yang mengindikasikan bahwa terjadi peningkatan dalam kualitas prediksi dari model deteksi objek. Selain itu dapat juga digunakan indikator peningkatan metrik lain yang dapat menjadi indikasi kenaikan performa model. Ini dapat dilakukan untuk mencegah terjadinya overfitting pada dataset yang digunakan. Di dalam YOLOv5 disimpan dua jenis model yakni last.pt dan best.pt. Model last.pt adalah file *weights* yang menyimpan epoch terakhir sedangkan model best.pt adalah file *weights* yang menyimpan epoch dengan hasil training yang terbaik berdasarkan hasil mAP.

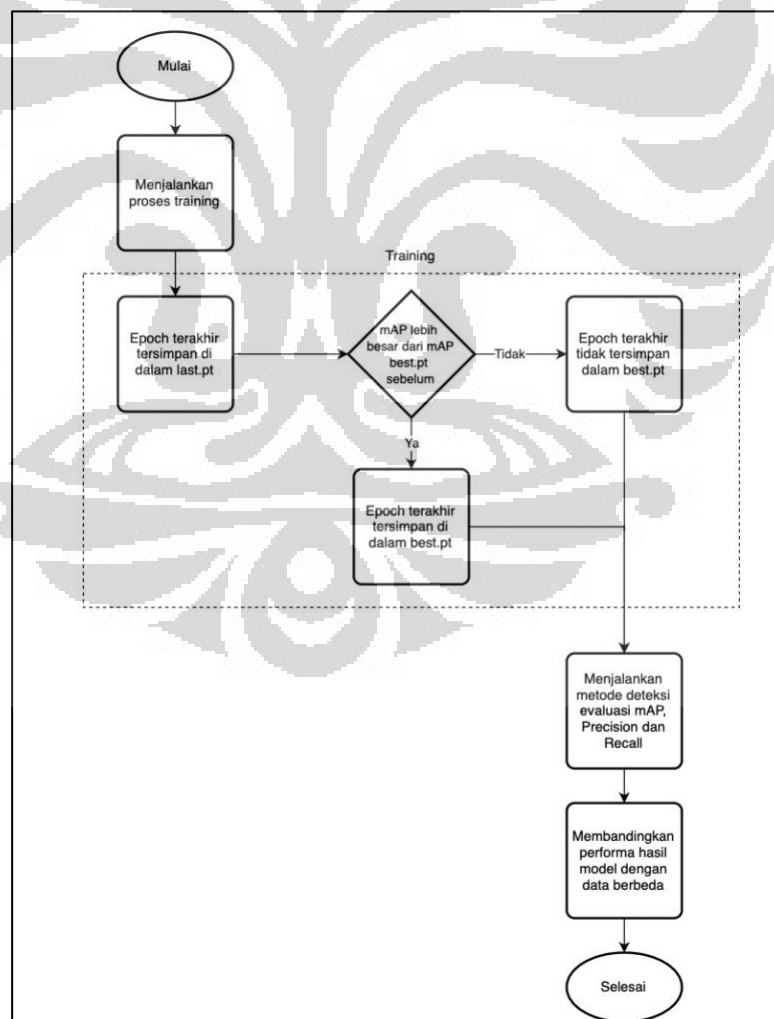
### 3.7. Evaluasi Model Deteksi Objek Makanan

Dalam proses evaluasi model deteksi objek dengan menggunakan YOLOv5, diperlukan penyimpanan pada file hasil training *weights* yang disimpan di dalam folder backup. Dalam proses training, sebuah file *weights* akan disave pada setiap kali iterasi ke dalam file bernama last.pt. Selain itu ada juga penyimpanan *weights* yang terdeteksi dengan evaluasi mAP yang tertinggi yang bernama best.pt.

Pada saat mengevaluasi model, perlu dilakukan perbandingan hasil file *weights* yang didapatkan dari hasil training dengan membandingkan nilai mAP bersama Precision dan Recall. Dengan menjalankan YOLOv5 dengan metode evaluasi mAP maka dapat dievaluasi nilai dari mAP yang didapatkan pada file *weights* yang ditentukan. Sehingga, dapat dilakukan perbandingan performa model pada saat evaluasi.

Dengan membandingkan nilai mAP, Precision, dan Recall dari model deteksi objek yang berbeda tersebut maka dapat dilakukan analisis dan perbandingan antara berbagai model yang berbeda dan memilih hasil yang terbaik dengan mempertimbangkan faktor performa model pada dataset training. Apabila sebuah file *weights* memiliki metrik yang lebih tinggi ketimbang yang lain, maka file *weights* tersebut yang melakukan fitting paling baik ke use-case dari model tersebut.

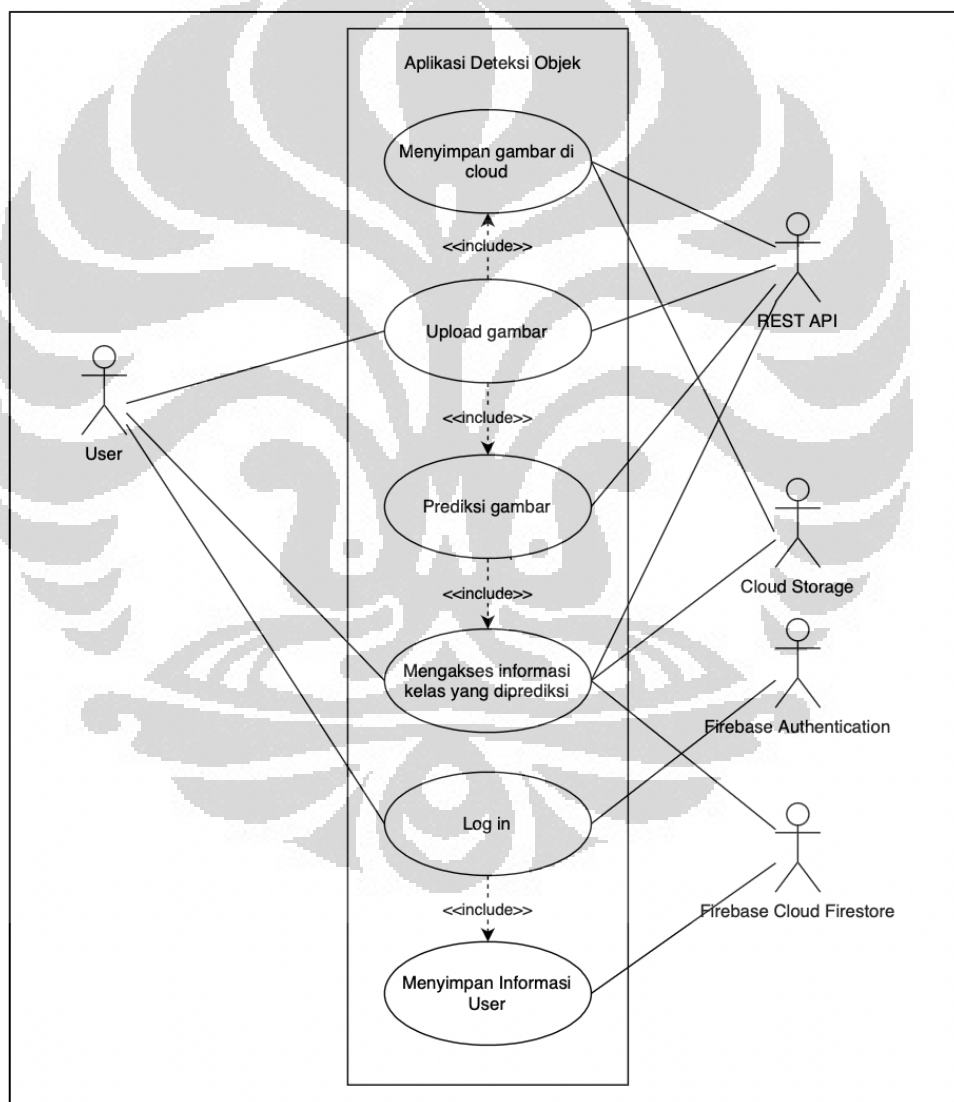
Perbandingan yang dilakukan pada model deteksi objek yang berbeda ini akan menjadi panduan selama proses pengembangan dataset yang dilakukan dan juga memberikan *feedback* terhadap progres yang dibangun dalam konteks pemilihan kategori-kategori pada dataset tersebut. Bagan *flowchart* mengenai proses evaluasi model yang dilakukan pada keseluruhan terlihat dalam Gambar 3.6.



Gambar 3.6. Flowchart Metode Evaluasi Model Hasil Pelatihan pada Dataset

### 3.8. Pengembangan Aplikasi

Untuk tahap pengembangan aplikasi digunakan bahasa pemrograman Dart dan platform Flutter. Proses pengembangan aplikasi mobile berbasis Flutter tidak hanya melibatkan bagian *frontend* saja, yakni Flutter sendiri, namun juga melibatkan integrasi dari sistem lain ke dalam aplikasi sehingga dapat memberikan fungsionalitas yang lengkap kepada user. Sistem-sistem yang digunakan dalam aplikasi antara lain adalah REST API, Cloud storage, Firebase Authentication, dan Firebase Cloud Firestore.



Gambar 3.7. Use Case Diagram untuk Aplikasi Phoodto

Untuk dapat melakukan interaksi dengan aplikasi dengan fungsionalitas penuh, maka diperlukan integrasi dari sistem-sistem berikut di dalam aplikasi Flutter. Alur dari penggunaan aplikasi adalah user dapat melakukan penciptaan user baru, login dan autentikasi user, penyimpanan informasi pribadi, kemudian juga fungsionalitas utama yakni pengunggahan gambar dan prediksi gambar yang akan dilakukan oleh perangkat server-side. Setelah mendapatkan hasil prediksi user pun akan mendapatkan feedback dari jenis makanan tersebut beserta deskripsi lain yang menyertai makanan tersebut yang tersimpan dalam Firebase Cloud Firestore.

Hasil output yang akan ditampilkan kepada pengguna aplikasi adalah sebuah gambar yang memperlihatkan hasil deteksi objek, kategori dari makanan yang terdeteksi, bersamaan dengan deskripsi dan skor *confidence* yang diberikan oleh model pendeteksi objek makanan tersebut. Deskripsi yang diberikan oleh aplikasi adalah sebuah deskripsi yang diambil dari kompilasi proses pencarian informasi terkait dari jenis makanan tersebut di tautan teratas Google kemudian dipersingkat sesuai kebutuhan. Dengan memberikan hasil output tersebut kepada pengguna aplikasi, diharapkan bahwa proses deteksi dan pemberian deskripsi secara otomatis akan membantu pengguna untuk memahami jenis makanan yang diambil fotonya baik melalui kamera secara langsung maupun melalui proses upload gambar dari galeri.

## BAB 4

### IMPLEMENTASI DAN ANALISIS

Bab ini akan membahas mengenai implementasi dari sistem dan pengujian yang dilakukan terhadap sistem deteksi objek menggunakan YOLOv5 serta analisa terhadap model pada tahap training dan pada tahap implementasi. Penelitian ini difokuskan kepada pembentukan model pendeteksi objek makanan yang dapat mendeteksi komponen makanan individu yang sudah ditentukan dengan keterbatasan sumber daya dan waktu.

#### 4.1. Implementasi Sistem

Untuk dapat membangun aplikasi dengan mengimplementasikan model deteksi objek yang telah dilatih dalam framework YOLOv5, maka diperlukan proses pengembangan model machine learning, arsitektur *frontend* sebagai tampilan utama aplikasi kepada user, dan *backend* aplikasi untuk menjalankan fungsionalitas prediksi.

Model deteksi objek yang dikembangkan dengan YOLOv5 dilakukan dengan menggunakan layanan Google Colab Pro. Layanan tersebut menyediakan GPU NVIDIA Tesla T4 dengan RAM sebesar 16 GB dan versi CUDA 11.2. Alasan penggunaan Google Colab Pro adalah karena proses pelatihan gambar membutuhkan kebutuhan RAM GPU yang tinggi, yang tidak dapat dipenuhi jika menggunakan layanan Google Colab yang gratis maupun pada perangkat pribadi. Pada bagian *frontend* Flutter digunakan sebagai landasan untuk membangun aplikasi *mobile*. Sedangkan, pada bagian *backend* digunakan Python untuk mengimplementasikan model deteksi objek yang dikembangkan dengan YOLOv5.

##### 4.1.1. Pengembangan Model Deteksi Objek

Model pendeteksi objek makanan dirancang untuk dijalankan pada aplikasi yang dapat berfungsi pada perangkat smartphone untuk dilakukan pengujian. Aplikasi smartphone yang dikembangkan dengan Flutter akan mengintegrasikan model machine learning agar dapat melakukan pengujian deteksi objek makanan.

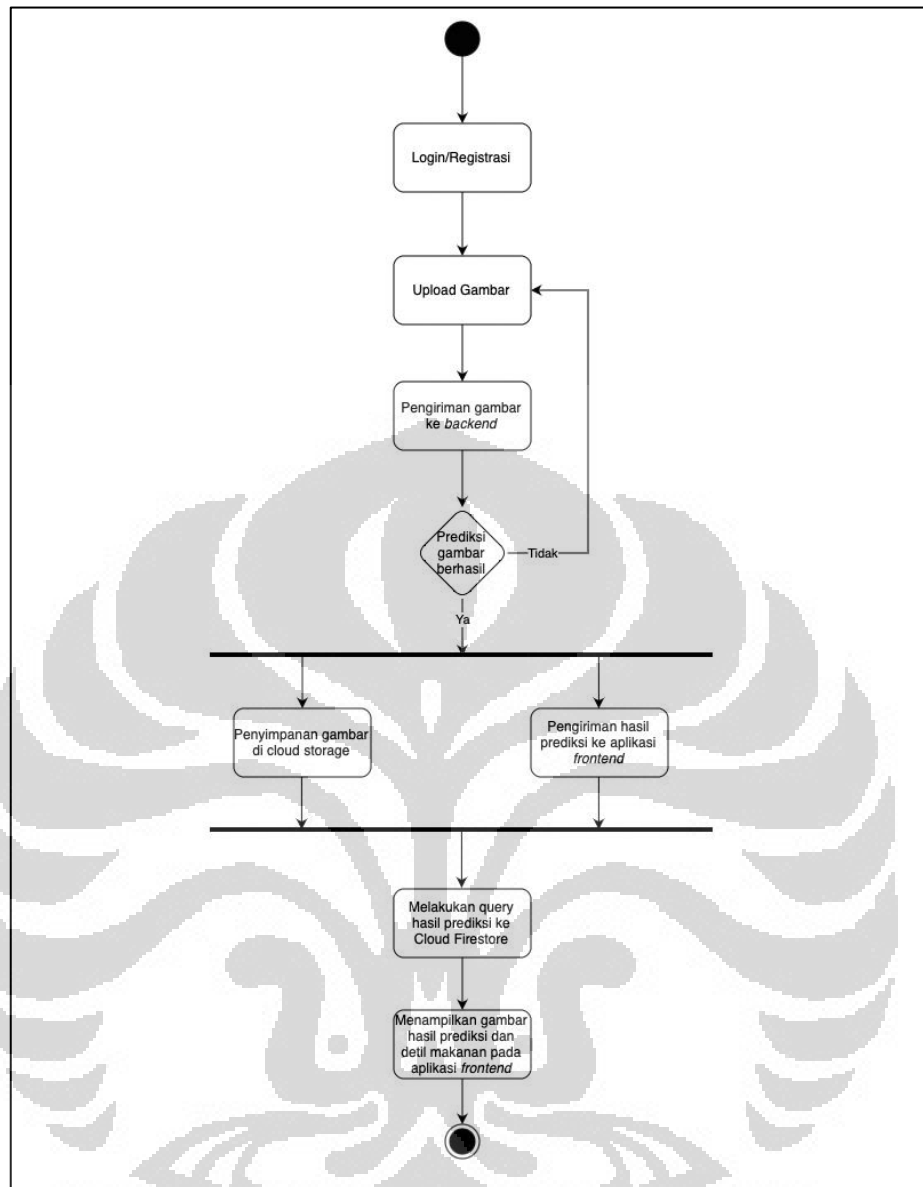
Untuk itu maka model yang dilatih dengan menggunakan YOLOv5 dengan spesifikasi terkecil yaitu YOLOv5s dan diintegrasikan ke dalam aplikasi melalui penggunaan REST API yang dirancang menggunakan Flask yang menjadi pusat melakukan prediksi model dan logika bisnis lainnya.

Pembentukan model YOLOv5 dilakukan dengan menggunakan PyTorch. Model yang dipilih untuk membangun sistem deteksi objek makanan adalah model YOLOv5 dengan arsitektur yang paling kecil, yakni YOLOv5s yang dipilih sesuai dengan kebutuhan dari sistem deteksi objek. Proses pembentukan model menggunakan kode dalam bahasa pemrograman Python dan bash scripting yang dibentuk dari gabungan library-library.

#### 4.1.2. Pengembangan Aplikasi Flutter

Pembentukan aplikasi dilakukan dengan menggunakan Flutter. Walaupun program Phoodto secara keseluruhan merupakan ide dan konsep yang diperoleh bersama tim Bangkit, hasil pengembangan konsep *user interface* aplikasi beserta dengan implementasi teknologi dikembangkan secara mandiri. Flutter dipilih sebagai platform pengembangan karena fleksibilitas antar-platform yang tersedia. Latar belakang pemilihan bahasa pemrograman Dart dan platform Flutter untuk mengembangkan aplikasi adalah karena *coverage* dari platform yang dikembangkan dalam Flutter lebih besar dibandingkan dengan *native* platform, misalnya pengembangan khusus dalam Android ataupun iOS. Dengan menggunakan Flutter maka hasil pengembangan aplikasi bisa menjangkau lebih banyak orang yang memiliki kebutuhan sesuai yang disediakan oleh aplikasi deteksi objek makanan ini.

Proses pengembangan dilakukan dengan mengembangkan bagian tampilan aplikasi, fungsionalitas aplikasi, kemudian error handling untuk menangani situasi *edge case*. Untuk bagian fungsionalitas aplikasi, aplikasi Flutter berkomunikasi dengan model machine learning melalui Application Programming Interface (API) yang dijalankan pada perangkat berbeda. Kedua sistem ini berkomunikasi sehingga dapat menampilkan hasil prediksi sesuai dengan fungsi aplikasi yang ditujukan.



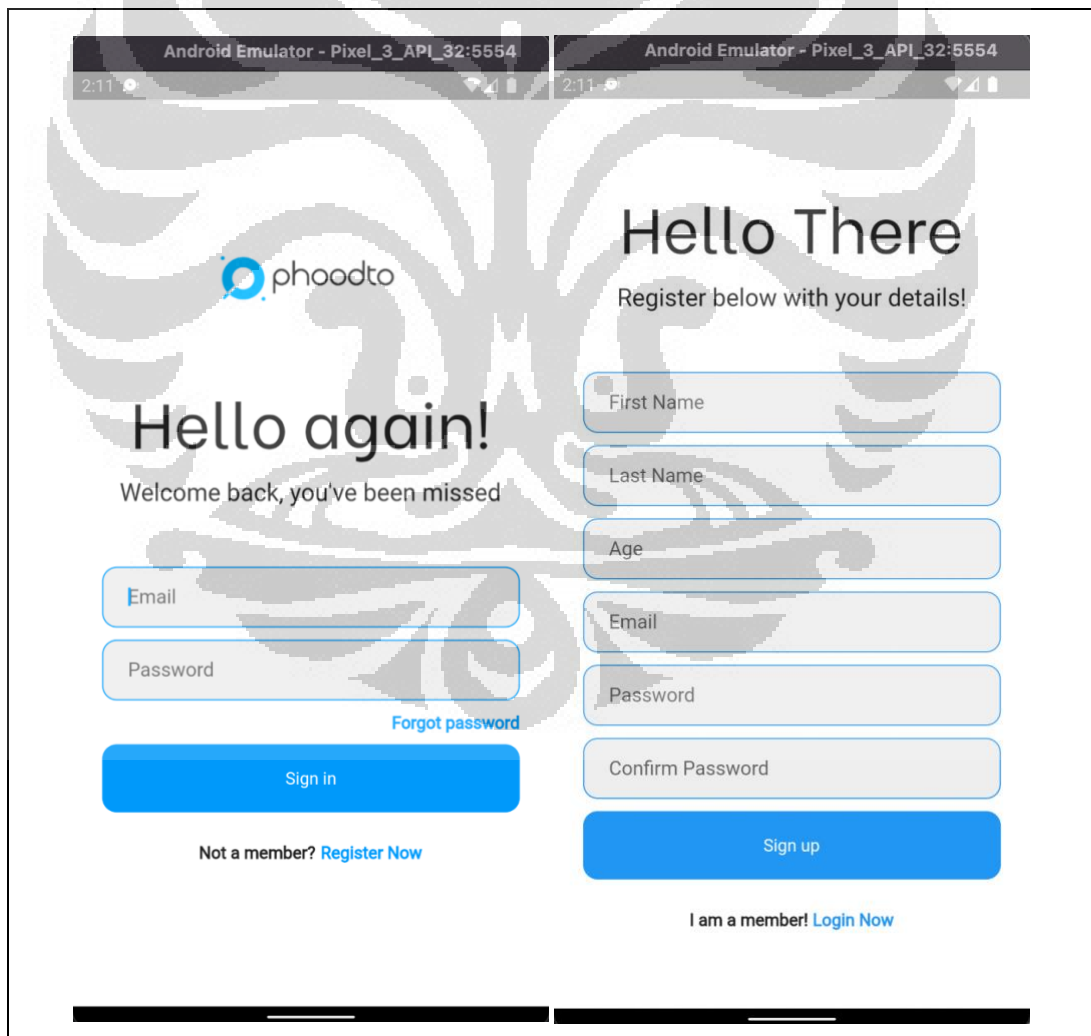
Gambar 4.1. Diagram Alur Aktivitas yang Mendeskripsikan Cara Kerja Aplikasi Secara Keseluruhan

Alur penggunaan aplikasi adalah user dapat melakukan pembuatan akun untuk mengakses aplikasi, kemudian pada laman utama user dapat menginput gambar melalui akses dari gallery ataupun dari kamera. Berikutnya terjadi pertukaran informasi yang menghasilkan hasil gambar prediksi yang berisi *bounding box* ditampilkan kepada pengguna.

Hasil prediksi nama makanan dalam response kemudian dapat ditampilkan kepada pengguna aplikasi. Kode nama makanan yang didapatkan dari server kemudian digunakan oleh aplikasi untuk melakukan query ke Cloud Firestore untuk

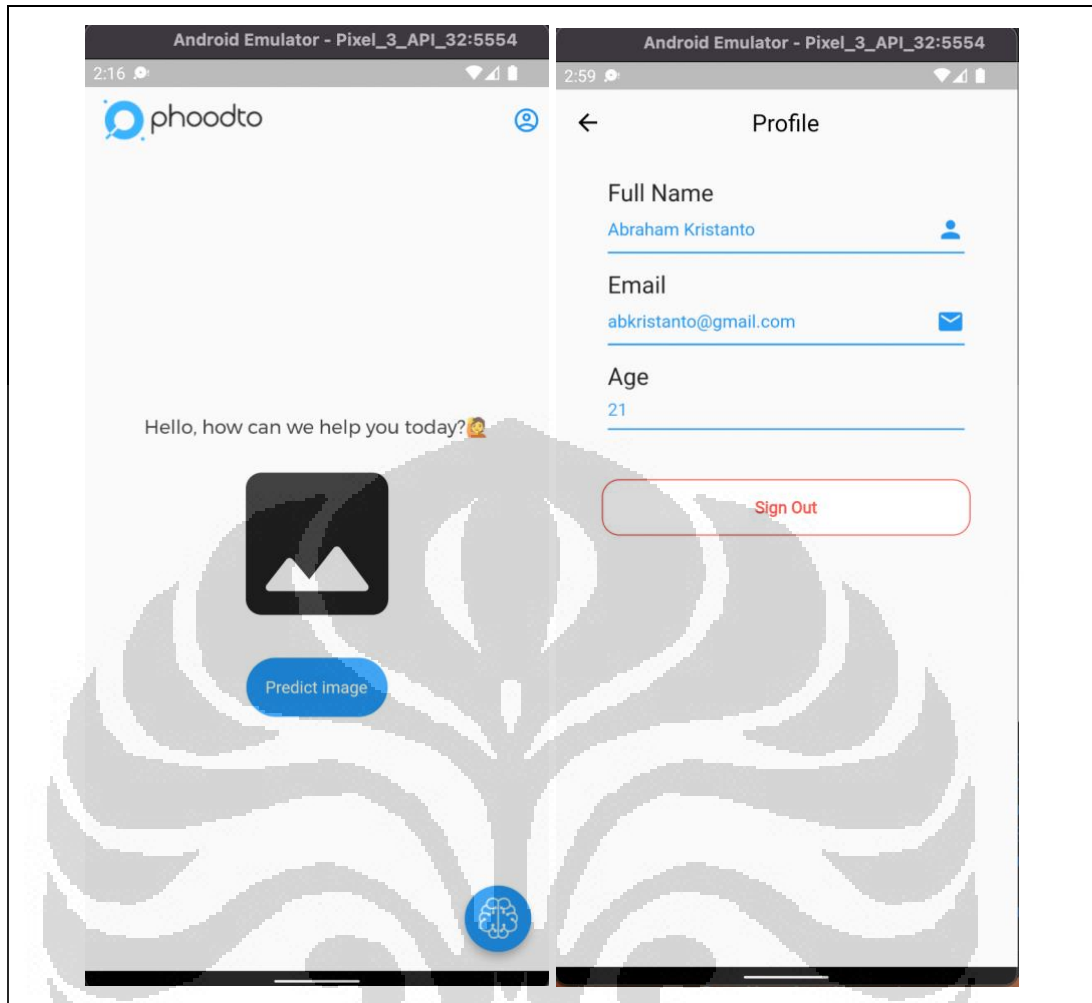
mendapatkan informasi mengenai deskripsi makanan. Informasi hasil prediksi akhir dan deskripsi makanan ini yang merupakan hasil akhir yang ditampilkan kepada pengguna aplikasi.

Untuk dapat melaksanakan beberapa fungsionalitas kunci seperti autentikasi, query, dan juga melakukan request dan response HTTP, digunakan beberapa package dari Flutter yang disediakan secara open source untuk dapat digunakan untuk developer. Package yang digunakan untuk mengembangkan aplikasi bagian *frontend* adalah seperti package http, cloud\_firestore, dll. Package ini membantu dalam proses pengembangan aplikasi sehingga dapat mengakses fungsionalitas dari aplikasi untuk melakukan keperluan Create, Read, Update, Delete (CRUD), autentikasi, dan juga melakukan HTTP Request.



Gambar 4.2. Tampilan Login dan Registrasi Aplikasi





Gambar 4.3. Tampilan Laman Utama dan Laman User Profile

Untuk keperluan pembuatan user, autentikasi, dan penyimpanan informasi menggunakan layanan Google Firebase yang dapat disambungkan dengan aplikasi yang dikembangkan menggunakan Flutter. Layanan Firebase yang digunakan antara lain adalah Firebase Authentication untuk autentikasi dan Cloud Firestore untuk keperluan penyimpanan data dalam bentuk collection yang dikirimkan kepada aplikasi dalam bentuk Javascript Object Notation (JSON). Di dalam Cloud Firestore disimpan data user dan data koleksi makanan yang dapat diprediksi oleh aplikasi.

### 4.1.3. Pengembangan Backend Aplikasi dan Deployment

Cara kerja dari bagian *backend* dari aplikasi adalah dengan memberikan endpoint prediksi yang dapat digunakan oleh bagian *frontend* aplikasi. Endpoint tersebut menerima gambar yang akan diproses oleh model deteksi objek makanan yang menggunakan library pytorch. Setelah gambar diproses oleh model, hasil gambar yang didapatkan diupload ke cloud storage, kemudian link upload cloud storage tersebut didapatkan kemudian dikirimkan dalam hasil result yang didapatkan.



Gambar 4.4. Tampilan Console Google Firebase pada Layanan Cloud Storage yang Menjadi Tempat Penyimpanan Gambar Hasil Deteksi Model Deteksi Objek.

Setelah diproses, endpoint ini akan mengembalikan sebuah response berupa Python dictionary yang berisi URL gambar, beserta koordinat hasil prediksi beserta nomor dan nama kelas yang berhasil diprediksi oleh model, bersama dengan nilai skor confidence dari prediksi tersebut. Hasil yang dikirimkan melalui HTTP Request oleh aplikasi dapat diinterpretasikan oleh bahasa pemrograman Dart dengan mengkonversikannya menjadi bentuk Map yang berisi pasangan kunci dan nilai.

Agar aplikasi dapat digunakan di luar lingkup kerja lokal, maka bagian *backend* dari aplikasi perlu dijalankan menggunakan layanan *cloud*. Untuk layanan *cloud* yang dipilih adalah menggunakan Google Cloud Platform (GCP). Langkah yang dilakukan adalah dengan menggunakan Docker, dibangun sebuah Docker

Image sehingga image tersebut dapat dijalankan menggunakan layanan Cloud Run pada GCP. Setelah bagian *backend* dari aplikasi dapat dijalankan pada GCP dan didapatkan endpoint dari *backend* tersebut maka aplikasi dapat langsung dijalankan pada perangkat HP asli.

#### 4.2. Dataset Pengujian

Dataset yang digunakan untuk penelitian menggunakan gambar yang diperoleh melalui tahap pengambilan data, labelling, dan proses augmentasi data. Data yang diperoleh melalui tahap pengambilan data memiliki ukuran gambar yang bervariasi karena sumber pengambilan data yang online. Data yang dikumpulkan bersumber gambar yang diambil dari Google Images dan diberikan label dengan menggunakan Roboflow. Dalam memberikan label, prioritas diberikan kepada gambar yang memiliki satu jenis kategori di dalam gambar tersebut, sehingga saat training pada satu gambar fokus penuh diberikan pada satu, atau maksimal dua kategori untuk menghasilkan model yang optimal.

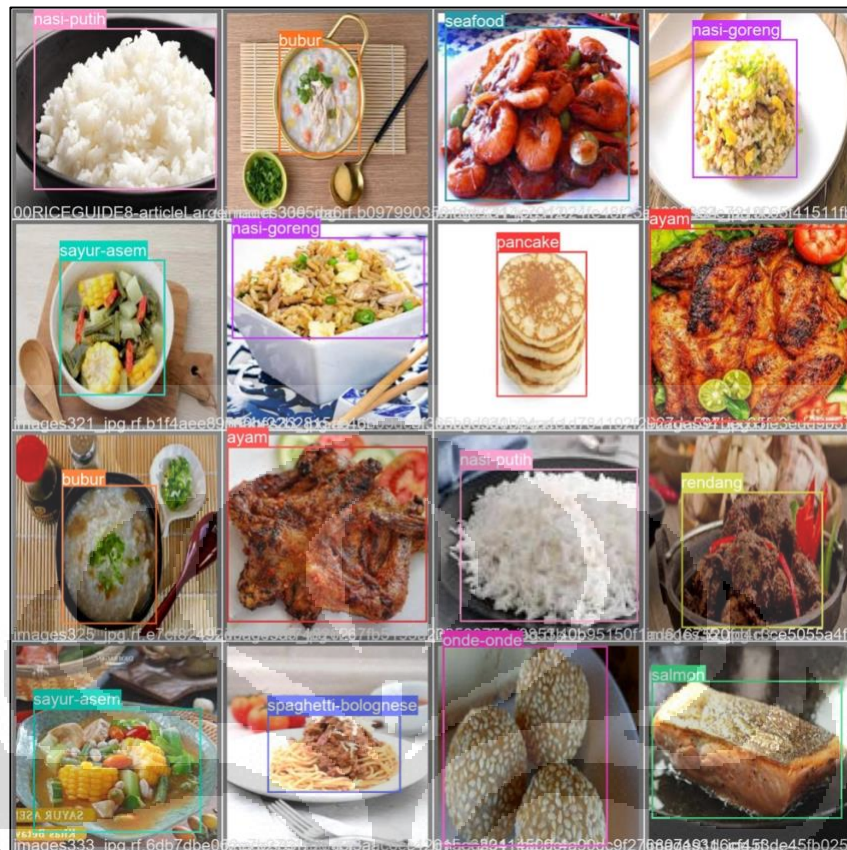
Setelah melalui proses labelling dan preprocessing maka gambar tersebut di-resize ke ukuran 640x640 piksel. Gambar di dalam dataset di-resize ke ukuran tersebut karena mengacu pada dokumentasi untuk training gambar pada arsitektur YOLOv5s, yang merupakan arsitektur YOLOv5 yang terkecil.

Untuk menambah variasi dari data, data yang telah didapatkan melalui proses pengambilan data kemudian diaugmentasi untuk menambah jumlah data agar model pendeteksi objek dapat memiliki sampel untuk pembelajaran yang lebih banyak guna meningkatkan akurasi dari model. Augmentasi yang dilakukan pada dataset terdapat horizontal flip, image cropping, peningkatan cahaya, dan pengurangan cahaya.

Jumlah data yang dikumpulkan secara total adalah sebanyak 40 kategori makanan Indonesia yang unik dan berbeda satu sama lain. Proses penambahan data sampai dengan 40 kategori dilakukan dengan proses seleksi untuk dapat memiliki potensi untuk menghasilkan model machine learning dengan tingkat akurasi yang tinggi.

Tabel 4.1. Spesifikasi Jumlah Data Setiap Kategori pada Dataset Akhir Sebelum Augmentasi Gambar

| Kelas           | Jumlah Data | Kelas                 | Jumlah Data |
|-----------------|-------------|-----------------------|-------------|
| Seafood         | 118         | Pempek                | 79          |
| Nasi putih      | 110         | Telur dadar           | 78          |
| Pancake         | 108         | Blueberry muffin      | 77          |
| Mie ayam        | 99          | Spaghetti bolognese   | 77          |
| Klepon          | 98          | Ikan goreng           | 76          |
| French fries    | 97          | Salmon                | 76          |
| Macaroni cheese | 97          | Chocolate chip cookie | 75          |
| Telur balado    | 96          | Sushi nigiri          | 75          |
| Nasi goreng     | 87          | Salad                 | 74          |
| Burger          | 86          | Kacang mete           | 72          |
| Doughnut        | 85          | Croissant             | 71          |
| Kangkung        | 85          | Ayam                  | 67          |
| Telur mata sapi | 84          | Kopi hitam            | 66          |
| Bubur           | 83          | Sate                  | 65          |
| Roti slice      | 83          | Pizza                 | 60          |
| Rendang         | 82          | Onde-onde             | 59          |
| Sayur asem      | 81          | Sushi makizushi       | 59          |
| Soto ayam       | 81          | Es pisang ijo         | 57          |
| Steak           | 81          | Martabak manis        | 57          |
| Red velvet      | 80          | Semangka              | 54          |
| <b>Total</b>    |             |                       | <b>3096</b> |



Gambar 4.5. Kumpulan Data di Dalam Dataset yang Terdiri dari Gambar Berbagai Kategori Makanan yang Telah Diberikan Label yang Terkait pada Jenis Makanan Tertentu.

Data yang dikumpulkan tidak bisa seragam pada seluruhnya karena keterbatasan sedia gambar makanan pada sumber online, dan tidak memungkinkan untuk melengkapi gambar makanan tersebut dengan foto-foto yang diambil secara pribadi. Ini telah diuji ketika data sudah mencapai 40 kategori dan ingin menambahkan kategori-kategori tertentu yang memiliki jumlah data yang lebih sedikit dibandingkan dengan kelas yang lain. Hasilnya adalah bahwa karena gambar yang dipilih kurang optimal, sehingga hasil training dari model adalah nilai akurasi mAP yang turun dibandingkan dengan sebelumnya, padahal tujuan dari penambahan data adalah meningkatkan akurasi dari model tersebut.

Setelah melalui proses augmentasi data hasil dataset akhir yang digunakan untuk training adalah seperti berikut.

Tabel 4.2. Spesifikasi Jumlah Data Setiap Kategori pada Dataset Akhir Setelah Augmentasi Gambar

| Kelas               | Jumlah Data | Kelas                 | Jumlah Data |
|---------------------|-------------|-----------------------|-------------|
| Seafood             | 221         | Telur dadar           | 142         |
| Nasi putih          | 199         | Pempek                | 141         |
| Pancake             | 190         | Rendang               | 141         |
| Klepon              | 179         | Blueberry muffin      | 140         |
| Telur balado        | 178         | Ikan goreng           | 140         |
| French fries        | 177         | Salmon                | 137         |
| Macaroni cheese     | 172         | Sushi nigiri          | 136         |
| Mie ayam            | 170         | Chocolate chip cookie | 135         |
| Nasi goreng         | 160         | Salad                 | 132         |
| Roti slice          | 157         | Croissant             | 127         |
| Kangkung            | 156         | Kacang mete           | 125         |
| Telur mata sapi     | 156         | Kopi hitam            | 122         |
| Doughnut            | 153         | Ayam                  | 121         |
| Burger              | 151         | Sate                  | 118         |
| Soto ayam           | 147         | Pizza                 | 107         |
| Steak               | 146         | Es pisang ijo         | 105         |
| Bubur               | 145         | Martabak manis        | 103         |
| Red velvet          | 142         | Onde onde             | 101         |
| Sayur asem          | 142         | Sushi makizushi       | 100         |
| Spaghetti bolognese | 142         | Semangka              | 99          |
| <b>Total</b>        |             |                       | <b>5574</b> |

### 4.3. Skenario Pengujian

Dalam penelitian ini akan dilakukan pengujian terhadap model yaitu pada tahap pelatihan berupa pengujian kualitas model deteksi objek makanan dan implementasi secara langsung dalam aplikasi deteksi objek makanan. Pengujian ini dilakukan dengan untuk dapat menghasilkan sistem deteksi objek makanan yang dapat memiliki performa yang terbaik.

Pada penelitian ini, dilakukan pengujian dan evaluasi terhadap model pada tahap pelatihan dan pada saat pengujian model untuk penggunaan dalam aplikasi nyata. Karena terdapat banyak faktor yang berubah dalam setiap gambar maka pemilihan jenis gambar yang bisa dimasukkan ke dalam kategori tertentu penting untuk dilakukan. Selama pelatihan model, dilakukan penskalaan model berdasarkan jumlah kategori data yang dilatih pada model tersebut sampai akhirnya berjumlah sebanyak 40 kategori. Dalam menentukan apakah dataset yang terbentuk dapat diskala lebih lanjut faktor yang digunakan adalah akurasi model serta waktu pelatihan model tersebut. Sedangkan, dalam langkah pemilihan data difaktorkan pemilihan kategori berdasarkan fitur dari makanan tertentu yang akan meningkatkan kemungkinan makanan tersebut terdeteksi dengan baik oleh sistem deteksi objek makanan.

Nilai akurasi model dihitung dengan menggunakan nilai mAP atau Mean Average Precision dari model tersebut. Tujuan dari digunakannya nilai mAP adalah karena hasil prediksi dari sistem deteksi objek makanan memiliki nilai true positive, false positive, dan false negative. Untuk permasalahan deteksi objek akurasi tidak memadai karena tidak memperhitungkan faktor selain nilai true positive dari suatu prediksi. Dalam perhitungan mAP diperhitungkan juga faktor false positive dan false negative di dalam setiap prediksi.

Selanjutnya model deteksi objek akan diimplementasikan di dalam sebuah sistem *backend* yang akan memungkinkan hasil prediksi terlihat dalam tampilan *frontend* yaitu berupa platform mobile Flutter yang mampu berfungsi di perangkat iOS maupun Android.

#### 4.4. Pengujian Model

Proses pengembangan dataset untuk training sepenuhnya dilakukan secara individu, sehingga proses untuk training tidak langsung dilakukan pada seluruh data karena dataset perlu ditingkatkan skalanya perlahan-lahan untuk memastikan data yang ditambahkan sesuai dengan kriteria akurasi prediksi yang baik. Sehingga, proses untuk training model dilakukan secara bertahap sesuai dengan ukuran dari dataset yang terdiri dari jumlah gambar dan juga total kelas yang terdapat di dalam dataset tersebut.

##### 4.4.1. Pengujian Pengaruh Penambahan Kategori Makanan Terhadap Akurasi

Sistem deteksi objek makanan yang dikembangkan memiliki target untuk bisa mendeteksi sebanyak 40 kategori makanan. Untuk bisa mencapai target tersebut dengan hasil prediksi mAP yang baik, penambahan data untuk dataset model training dilakukan secara bertahap, dengan memastikan setiap kategori yang ditambahkan mampu untuk menghasilkan model yang dapat memprediksi dengan baik. Training dataset dilakukan menggunakan YOLOv5s dengan hyperparameter setting standar seperti yang terdokumentasi di dalam Github YOLOv5 Ultralytics beserta dokumentasinya.

Daftar lengkap untuk setiap kategori yang berada dalam tiap-tiap batch *training* yang dilakukan untuk pengujian ini terdapat pada Tabel 4.3. Pada tabel tersebut terlihat terdapat perkembangan dataset yang awalnya berukuran 5 kategori, kemudian 8 kategori, berikutnya kedua dataset sebelumnya digabungkan menjadi 13 kategori. Setelah itu, dataset ditambah dengan 5 kategori dan menjadi bertotal 18 kategori, dan berikutnya 23 kategori. Berikutnya, dataset diperluas menjadi 25 kategori, kemudian ditambah 5 menjadi 30, 35, dan terakhir 40 kategori.

Pengembangan dataset dilakukan dengan kategori-kategori tersebut dengan latar belakang pengetahuan umum mengenai jenis makanan terkenal di Indonesia beserta makanan yang menjadi bahan konsumsi masyarakat. Akan tetapi, terdapat keterbatasan kategori yang dapat dimasukkan ke dalam dataset karena keterbatasan data berkualitas yang tersedia secara online dan waktu pengumpulan data.



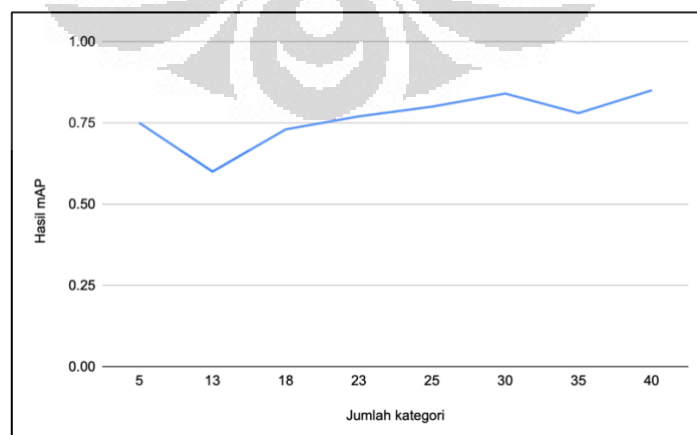
Tabel 4.3. Daftar Kategori yang Mendeskripsikan Perkembangan Dataset Deteksi Objek Makanan yang Dilakukan

| Nomor | 5 kategori | 13 kategori         | 18 kategori         | 23 kategori         | 25 kategori           | 30 kategori           | 35 kategori           | 40 kategori           |
|-------|------------|---------------------|---------------------|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1     | mie ayam   | pizza               | mie ayam            | mie ayam            | mie ayam              | mie ayam              | mie ayam              | mie ayam              |
| 2     | rendang    | pancake             | rendang             | rendang             | rendang               | rendang               | rendang               | rendang               |
| 3     | pempek     | burger              | pempek              | pempek              | pempek                | pempek                | pempek                | pempek                |
| 4     | ayam       | martabak manis      | ayam                | ayam                | ayam                  | ayam                  | ayam                  | ayam                  |
| 5     | sate       | steak               | sate                | sate                | sate                  | sate                  | sate                  | sate                  |
| 6     |            | roti slice          | pizza               | pizza               | pizza                 | pizza                 | pizza                 | pizza                 |
| 7     |            | salad               | pancake             | pancake             | pancake               | pancake               | pancake               | pancake               |
| 8     |            | telur mata sapi     | burger              | burger              | burger                | burger                | burger                | burger                |
| 9     |            | nasi putih          | martabak manis      | martabak manis      | martabak manis        | martabak manis        | martabak manis        | martabak manis        |
| 10    |            | salmon              | steak               | steak               | steak                 | steak                 | steak                 | steak                 |
| 11    |            | spaghetti bolognese | roti slice          | roti slice          | roti slice            | roti slice            | roti slice            | roti slice            |
| 12    |            | telur dadar         | salad               | salad               | salad                 | salad                 | salad                 | salad                 |
| 13    |            | nasi goreng         | telur mata sapi     | telur mata sapi     | telur mata sapi       | telur mata sapi       | telur mata sapi       | telur mata sapi       |
| 14    |            |                     | nasi putih          | nasi putih          | nasi putih            | nasi putih            | nasi putih            | nasi putih            |
| 15    |            |                     | salmon              | salmon              | salmon                | salmon                | salmon                | salmon                |
| 16    |            |                     | spaghetti bolognese | spaghetti bolognese | spaghetti bolognese   | spaghetti bolognese   | spaghetti bolognese   | spaghetti bolognese   |
| 17    |            |                     | telur dadar         | telur dadar         | telur dadar           | telur dadar           | telur dadar           | telur dadar           |
| 18    |            |                     | nasi goreng         | nasi goreng         | nasi goreng           | nasi goreng           | nasi goreng           | nasi goreng           |
| 19    |            |                     | klepon              | klepon              | klepon                | klepon                | klepon                | klepon                |
| 20    |            |                     | telur balado        | telur balado        | telur balado          | telur balado          | telur balado          | telur balado          |
| 21    |            |                     | ikan goreng         | ikan goreng         | ikan goreng           | ikan goreng           | ikan goreng           | ikan goreng           |
| 22    |            |                     | seafood             | seafood             | seafood               | seafood               | seafood               | seafood               |
| 23    |            |                     | es pisang ijo       | es pisang ijo       | es pisang ijo         | es pisang ijo         | es pisang ijo         | es pisang ijo         |
| 24    |            |                     |                     | doughnut            | doughnut              | doughnut              | doughnut              | doughnut              |
| 25    |            |                     |                     | sayur asem          | sayur asem            | sayur asem            | sayur asem            | sayur asem            |
| 26    |            |                     |                     |                     | macaroni cheese       | macaroni cheese       | macaroni cheese       | macaroni cheese       |
| 27    |            |                     |                     |                     | chocolate chip cookie | chocolate chip cookie | chocolate chip cookie | chocolate chip cookie |
| 28    |            |                     |                     |                     | croissant             | croissant             | croissant             | croissant             |
| 29    |            |                     |                     |                     | kopi hitam            | kopi hitam            | kopi hitam            | kopi hitam            |
| 30    |            |                     |                     |                     | sushi makizushi       | sushi makizushi       | sushi makizushi       | sushi makizushi       |
| 31    |            |                     |                     |                     |                       | french fries          | french fries          | french fries          |
| 32    |            |                     |                     |                     |                       | kacang mete           | kacang mete           | kacang mete           |
| 33    |            |                     |                     |                     |                       | blueberry muffin      | blueberry muffin      | blueberry muffin      |
| 34    |            |                     |                     |                     |                       | semangka              | semangka              | semangka              |
| 35    |            |                     |                     |                     |                       | onde onde             | onde onde             | onde onde             |
| 36    |            |                     |                     |                     |                       |                       | kangkung              | kangkung              |
| 37    |            |                     |                     |                     |                       |                       | bubur                 | bubur                 |
| 38    |            |                     |                     |                     |                       |                       | red velvet            | red velvet            |
| 39    |            |                     |                     |                     |                       |                       | soto ayam             | soto ayam             |
| 40    |            |                     |                     |                     |                       |                       | sushi nigiri          | sushi nigiri          |

Tabel 4.4. Perkembangan Hasil Model Deteksi Objek Makanan Seiring dengan Perkembangan Ukuran Dataset.

| No. | Jumlah kategori | Batch Size | Waktu per epoch (s) | GPU Memory (GB) | Precision | Recall | Hasil mAP | Standar Deviasi mAP |
|-----|-----------------|------------|---------------------|-----------------|-----------|--------|-----------|---------------------|
| 1   | 5               | 32         | 4                   | 7,14            | 0,71      | 0,74   | 0,75      | 0,03                |
| 2   | 13              | 32         | 5                   | 7,19            | 0,56      | 0,61   | 0,60      | 0,05                |
| 3   | 18              | 32         | 11                  | 7,14            | 0,70      | 0,67   | 0,73      | 0,07                |
| 4   | 23              | 32         | 16                  | 7,3             | 0,72      | 0,74   | 0,77      | 0,06                |
| 5   | 25              | 32         | 17                  | 7,14            | 0,76      | 0,74   | 0,80      | 0,03                |
| 6   | 30              | 64         | 42                  | 12,1            | 0,79      | 0,80   | 0,84      | 0,01                |
| 7   | 35              | 64         | 27                  | 12,4            | 0,73      | 0,75   | 0,78      | 0,04                |
| 8   | 40              | 64         | 34                  | 11,8            | 0,80      | 0,80   | 0,85      | 0,08                |

Seiring dengan penambahan data yang dilakukan secara bertahap, perlu dipastikan bahwa hasil Mean Average Precision (MAP) memiliki nilai yang sepadan untuk dapat dimasukkan ke dalam dataset. Untuk itu, setiap kali melakukan penambahan data, dipastikan sebelumnya bahwa MAP memiliki nilai yang cukup tinggi sebelum melanjutkan proses penambahan gambar dan labelling data. Proses ini dilakukan agar data yang ditambahkan ke dalam dataset merupakan data yang berkualitas tinggi dan tidak akan menghasilkan model hasil training yang memiliki kualitas buruk.



Gambar 4.6. Perkembangan Hasil mAP Seiring dengan Peningkatan Jumlah Kategori dalam Dataset

Terlihat bahwa untuk ukuran dataset dengan kelas sampai dengan 25 kelas, training dilakukan dengan menggunakan batch size sebesar 32. Dengan batch size sebesar 32, penambahan data dari 5 kategori sampai dengan 25 kategori menyebabkan waktu training untuk bertambah sampai lebih dari 4 kali sebelumnya, sehingga waktu training secara total pun bertambah menjadi 4 kali sebelumnya.

Setelah mencapai total sebanyak 30 kategori, batch size ditingkatkan menjadi 64 untuk membantu mempercepat waktu training dan memaksimalkan utilisasi dari sumber daya GPU yang tersedia pada Google Colab Pro. Meningkatkan batch size sampai dengan 64 memiliki dampak penggunaan sumber daya memori GPU menjadi lebih tinggi, namun tidak melampaui batas yang diberikan oleh Google Colab Pro. Sumber daya memori GPU yang digunakan mengalami peningkatan hingga mencapai angka Hasil akhir yang diperoleh dari proses training model adalah sebuah model yang dilatih pada dataset berukuran 40 kategori dan memiliki nilai MAP 0,913.

#### 4.4.2. Pengujian Pengaruh Augmentasi Data Terhadap Akurasi

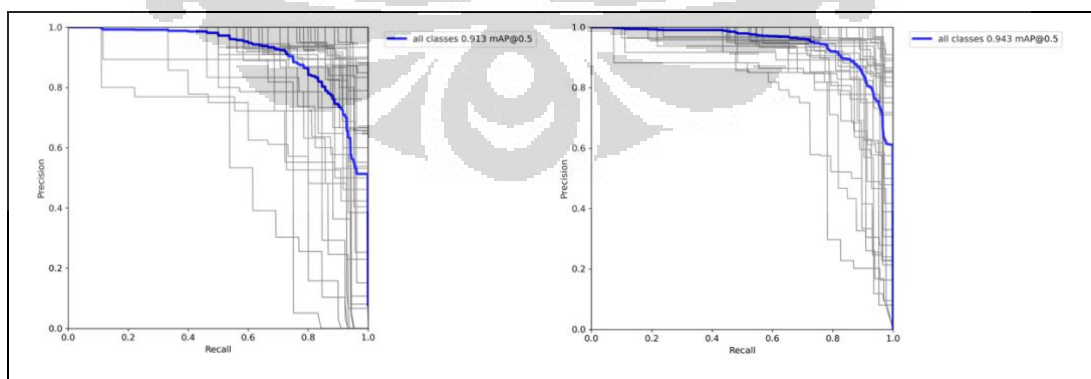
Pada saat pengembangan dataset dan proses terakhir yang dilakukan untuk menambahkan data, dilakukan augmentasi kepada dataset untuk memperbesar data training pada model. Setelah melakukan augmentasi, total gambar pada dataset bertambah dari sebelumnya bernilai 3096 gambar menjadi 5574 gambar. Secara umum, terlihat bahwa hasil dari augmentasi data membuat model menjadi memiliki performa yang lebih baik.

Tabel 4.5. Perbandingan Data Hasil Training Model Terbaik pada Dataset Sebelum Augmentasi Data dan Setelah Augmentasi Data.

| No. | Jumlah kategori            | Total Data | Hasil mAP 0.5 | Precision | Recall |
|-----|----------------------------|------------|---------------|-----------|--------|
| 1   | 40<br>(sebelum augmentasi) | 3096       | 0,913         | 0,839     | 0,869  |
| 2   | 40<br>(setelah augmentasi) | 5574       | 0,943         | 0,889     | 0,904  |

Secara garis besar, terlihat bahwa hasil Mean Average Precision dari model dengan tambahan data augmentasi menjadi memiliki nilai mAP yang lebih tinggi, yaitu 0,943 ketimbang 0,913. Metrik lainnya seperti precision dan juga recall mengalami peningkatan. Precision meningkat dari 0,839 ke 0,889 dan Recall meningkat dari 0,869 ke 0,904. Hasil ini berarti bahwa presisi dalam melakukan prediksi lebih tinggi, dan juga frekuensi model dapat mengenali suatu objek semakin meningkat. Ini menandakan bahwa performa dari model akan semakin baik dalam menghadapi skenario dunia nyata dengan bisa menangkap lebih banyak *occurrence* dari suatu objek makanan di dalam gambar ketimbang sebelumnya, dengan tingkat akurasi yang lebih tinggi.

Pada Gambar 4.7 terlihat bahwa Kurva Precision-Recall atau seringkali disebut Kurva P-R antara model yang dilatih dengan data sebelum diaugmentasi dan model yang dilatih dengan data setelah diaugmentasi. Dari hasil pengamatan, terlihat bahwa hasil Kurva P-R pada model yang dilatih dengan data yang sudah diaugmentasi memiliki tingkat kelandaian yang lebih panjang dibandingkan sebelum diaugmentasi. Selain itu terlihat bahwa tingkat kejatuhan kurva lebih lambat dibandingkan sebelumnya. Terakhir, terlihat bahwa kurva di kanan memiliki tingkat presisi yang lebih tinggi pada tingkat recall yang lebih tinggi juga. Ini berarti bahwa secara keseluruhan model memiliki kemampuan untuk dapat melakukan deteksi objek makanan dengan lebih baik, yang ditunjukkan dengan nilai mAP yang lebih tinggi sebanyak 0,03.



Gambar 4.7. Perbandingan Kurva Precision-Recall Antara Model Hasil Training Sebelum (kiri) dan Sesudah (kanan) Augmentasi Data.

Peningkatan yang terlihat pada Kurva P-R terlihat lebih jelas ketika diteliti lebih lanjut untuk mAP setiap kategori yang ada pada dataset ini. Dalam Tabel 4.6, terlihat dua kolom perbandingan mAP untuk tiap kelas kategori dengan data sebelum diaugmentasi dan setelah diaugmentasi. Pada tabel tersebut, telah ditandai dengan warna hijau setiap kelas kategori yang mengalami peningkatan yang signifikan, yang dapat dikategorisasikan sebagai peningkatan di atas nilai 0,1. Peningkatan tersebut terlihat terjadi pada kategori pancake, salmon, ayam, kopi hitam, sate, pizza, dan sushi makizushi.

Peningkatan dalam performa yang dapat dianalisis melalui Kurva P-R melalui pada tabel perbandingan mAP terjadi karena adanya peningkatan data yang signifikan untuk data training model. Selain adanya peningkatan data, perubahan yang terjadi selama augmentasi data dapat membuat model untuk beradaptasi pada situasi yang lebih fleksibel sehingga dapat meningkatkan Recall dari model sekaligus Precision dalam mendeteksi objek makanan. Permasalahan deteksi objek makanan tidak mudah karena suatu jenis makanan dapat memiliki bentuk yang bervariasi dan dengan situasi pencahayaan, kondisi zoom foto, serta bentuk yang seringkali tidak simetris. Oleh karena itu, memperbanyak skenario training agar model siap untuk menghadapi situasi tersebut dapat meningkatkan performa model terutama untuk pengaplikasian secara langsung.



Gambar 4.8 Hasil prediksi yang dilakukan oleh model dengan data yang sudah diaugmentasi

Tabel 4.6. Perbandingan Mean Average Precision (mAP) Untuk Setiap Kategori Dataset Sebelum dan Sesudah Augmentasi Data.

| Kelas           | mAP <i>pre-augmented</i> | mAP <i>post-augmented</i> | Kelas                 | mAP <i>pre-augmented</i> | mAP <i>post-augmented</i> |
|-----------------|--------------------------|---------------------------|-----------------------|--------------------------|---------------------------|
| Seafood         | 0,936                    | 0,906                     | Pempek                | 0,919                    | 0,927                     |
| Nasi putih      | 0,964                    | 0,971                     | Telur dadar           | 0,818                    | 0,855                     |
| Pancake         | 0,879                    | 0,990                     | Blueberry muffin      | 0,967                    | 0,965                     |
| Mie ayam        | 0,975                    | 0,940                     | Spaghetti bolognese   | 0,946                    | 0,955                     |
| Klepon          | 0,972                    | 0,990                     | Ikan goreng           | 0,919                    | 0,957                     |
| French fries    | 0,958                    | 0,971                     | Salmon                | 0,714                    | 0,833                     |
| Macaroni cheese | 0,966                    | 0,970                     | Chocolate chip cookie | 0,996                    | 0,995                     |
| Telur balado    | 0,981                    | 0,951                     | Sushi nigiri          | 0,926                    | 0,995                     |
| Nasi goreng     | 0,954                    | 0,91                      | Salad                 | 0,942                    | 0,983                     |
| Burger          | 0,996                    | 0,981                     | Kacang mete           | 0,996                    | 0,954                     |
| Doughnut        | 0,859                    | 0,926                     | Croissant             | 0,953                    | 0,995                     |
| Kangkung        | 0,973                    | 0,995                     | Ayam                  | 0,659                    | 0,911                     |
| Telur mata sapi | 0,911                    | 0,906                     | Kopi hitam            | 0,811                    | 0,923                     |
| Bubur           | 0,942                    | 0,987                     | Sate                  | 0,726                    | 0,905                     |
| Roti slice      | 0,995                    | 0,956                     | Pizza                 | 0,802                    | 0,98                      |
| Rendang         | 0,922                    | 0,872                     | Onde-onde             | 0,938                    | 0,987                     |
| Sayur asem      | 0,985                    | 0,995                     | Sushi makizushi       | 0,700                    | 0,866                     |
| Soto ayam       | 0,975                    | 0,991                     | Es pisang ijo         | 0,984                    | 0,970                     |
| Steak           | 0,875                    | 0,803                     | Martabak manis        | 0,848                    | 0,780                     |
| Red velvet      | 0,993                    | 0,995                     | Semangka              | 0,941                    | 0,995                     |

Dari perbandingan hasil antara model yang dilatih tanpa augmentasi data dan model yang dilatih dengan augmentasi data, terlihat bahwa terjadi peningkatan pada sebanyak 26 dari 40 kategori. Dalam Tabel 4.6 kategori-kategori yang mengalami peningkatan ditandai dengan warna hijau. Ini berarti bahwa sebanyak 65% dari kategori di dalam dataset mengalami peningkatan. Hasil ini tercerminkan dari peningkatan mAP semua kategori dari 0,913 menjadi 0,943. Selain itu terlihat beberapa kategori spesifik yang mengalami peningkatan pesat, dan teramati bahwa kelas yang mengalami peningkatan pesat merupakan kelas yang sebelumnya sulit untuk didiferensiasikan oleh model deteksi objek, misalnya seperti ayam dan sate. Sehingga, dengan peningkatan variabilitas data melalui augmentasi data yang dilakukan berhasil untuk menciptakan model yang memiliki akurasi lebih tinggi dibandingkan sebelumnya.

#### 4.4.3. Pengujian Model pada Aplikasi Deteksi Objek

Untuk dapat menjalankan model pada aplikasi deteksi objek, yaitu *frontend* aplikasi Flutter, dikembangkan sistem *backend* dalam bahasa pemrograman Python untuk dapat meng-host custom model yang telah dilatih. Model deteksi objek makanan dapat di-host dengan menggunakan cuplikan kode tersebut dengan menspesifikasikan lokasi dari file *weights* dari iterasi epoch yang menghasilkan mAP terbaik.

```
model = torch.hub.load("ultralytics/yolov5", "custom",
path="./best.pt")
```

Gambar 4.9. Kode yang Digunakan untuk Mengunduh Model Deteksi Objek yang Telah Dirancang ke dalam Sistem *Backend*.

Untuk mendapatkan data hasil prediksi digunakan cuplikan kode berikut yang berfungsi untuk mengkonversikan hasil prediksi ke dalam format JSON yang bisa digunakan oleh aplikasi *frontend*. Selain itu dilakukan juga penyimpanan gambar baru bernama image0.jpg di lokal yang merupakan gambar hasil prediksi beserta dengan *bounding box* yang telah didapatkan dari model yang memprediksi objek makanan di dalam sistem deteksi objek makanan yang telah dikembangkan.

```

results = model(img, size=640)
data = results.pandas().xyxy[0].to_json(orient="records")

results.render() # updates results.imgs with boxes and labels
for img in results.imgs:
    img_base64 = Image.fromarray(img)
    img_base64.save("static/image0.jpg", format="JPEG")

```

Gambar 4.10. Kode yang Digunakan untuk Memetakan Hasil Prediksi Kepada Gambar dan Menyimpannya di Dalam Penyimpanan Lokal.

Sistem deteksi objek yang dikembangkan dalam Python merupakan sebuah Application Programming Interface yang mengembalikan sebuah response seperti gambar berikut ketika mendapatkan request berupa gambar. Berikut adalah contoh ketika sistem deteksi objek mendeteksi “es-pisang-ijo”. Proses yang terjadi adalah terdapat dua field utama yang dikembalikan, yakni url dan data. Url merupakan link kepada cloud storage yang digunakan untuk menyimpan gambar hasil deteksi, kemudian data adalah data yang didapatkan dari hasil prediksi sistem deteksi objek, yang berisi koordinat objek makanan kemudian confidence score, kelas, beserta nama.

```

{'url': 'https://storage.googleapis.com/flutter-objectdetection.appspot.com/22/05/2022%2019%3A03%3A24.jpg',
 'data':
 ' [{"xmin":222.9797363281,"ymin":31.6440238953,"xmax":484.691528320
 3,"ymax":254.4212188721,"confidence":0.8106526732,"class":34,"name
 ":"steak"}, {"xmin":48.9029579163,"ymin":23.4830379486,"xmax":266.1
 913757324,"ymax":236.66015625,"confidence":0.7658928037,"class":34
 ,"name":"steak"}] ' }

```

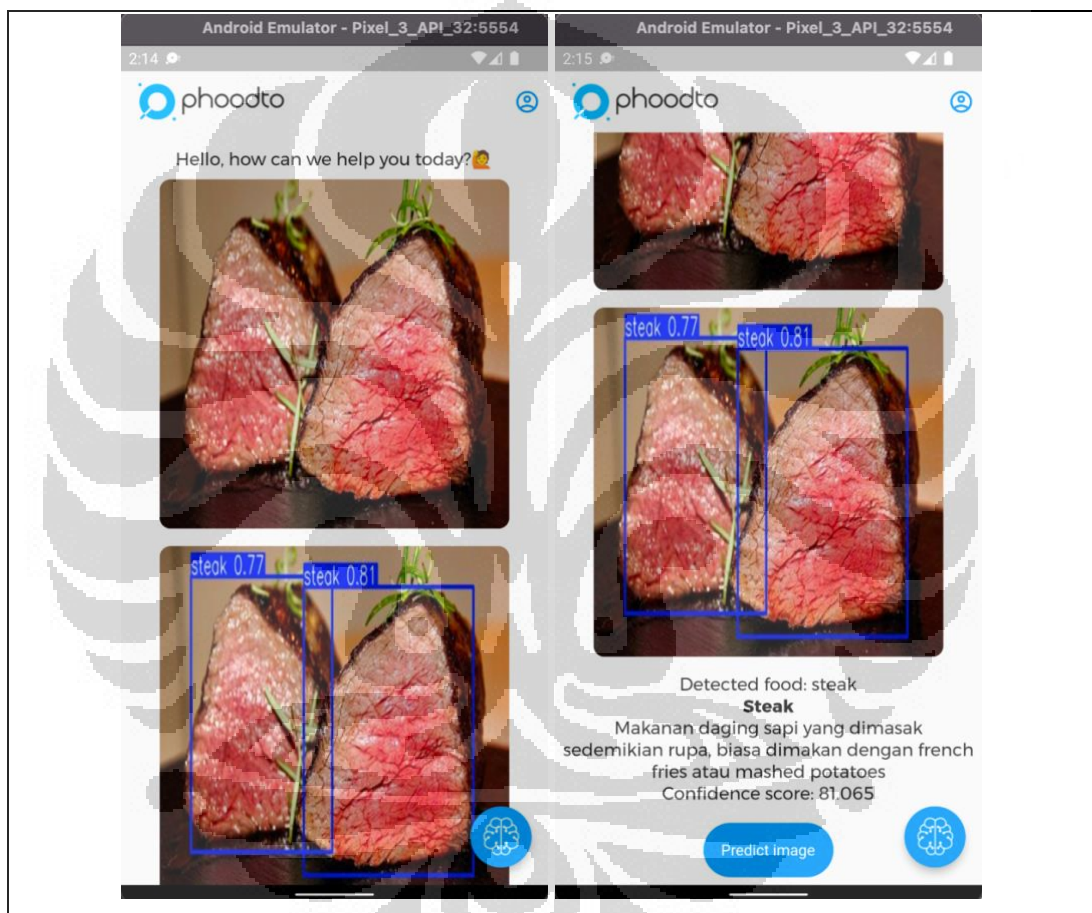
Gambar 4.11. Hasil Javascript Object Notation (JSON) yang Dikembalikan dalam Bentuk HTTP Response Ketika Terdapat Request yang Ditujukan kepada Endpoint REST API.

Data yang didapatkan dari API response tersebut kemudian digunakan oleh aplikasi *frontend* Flutter untuk menampilkan gambar dengan url yang diberikan beserta jenis makanan yang terdeteksi di dalam gambar tersebut. Kode nama makanan yang diberikan oleh API response tersebut digunakan untuk melakukan query kepada Firebase Firestore untuk mendapatkan nama lengkap makanan tersebut bersama dengan deskripsi lengkap dari makanan tersebut. Informasi tersebut tersimpan di dalam database Firestore dan dapat diperoleh oleh bagian *frontend* aplikasi ketika model selesai memprediksi dan API response mengembalikan sebuah kode makanan di dalam database tersebut.



Tabel 4.7. Spesifikasi Perangkat Android Emulator yang Digunakan untuk Menjalankan Aplikasi Secara Lokal

| Nama Perangkat    | Google Pixel 3 |
|-------------------|----------------|
| API Level         | 32             |
| Dimensi Perangkat | 2160 x 1080    |
| ABI               | arm64-v8a      |



Gambar 4.12. Hasil yang Ditampilkan oleh Aplikasi Sesuai dengan Hasil Prediksi yang Ditetapkan oleh Sistem *Backend* yang Mengimplementasikan Model Deteksi Objek Makanan.

Setelah dilakukan pengujian implementasi pada perangkat lokal, dilakukan pengujian terhadap implementasi sistem deteksi objek secara nyata pada perangkat asli dan sistem komputasi awan. Untuk melakukan hal tersebut, aplikasi perlu diunduh dari lokal ke perangkat keras yang digunakan yaitu *smartphone*. Selain itu, model deteksi objek makanan perlu di-*host* pada GCP sehingga dapat diakses melalui perangkat keras.

Kapasitas yang digunakan pada GCP untuk melakukan *hosting* model deteksi objek adalah dengan alokasi CPU yang hanya dilakukan saat pemrosesan *request*, dengan CPU sebanyak 4, memori sebesar 2 GiB, dan diatur dengan *auto-scaling* dengan jumlah *instance* maksimum sebesar 10 *instance*. Dengan pengaturan demikian, ditemukan bahwa terdapat keseimbangan antara kecepatan pemrosesan yang baik dan penggunaan sumber daya yang maksimum.

Tabel 4.8. Hasil Pengujian Implementasi Model Deteksi Objek Makanan ke dalam Aplikasi

| Pengujian ke-          | Delay Aplikasi (s) | Keterangan (jenis makanan) |
|------------------------|--------------------|----------------------------|
| 1                      | 1,896              | Martabak                   |
| 2                      | 2,313              | Kangkung                   |
| 3                      | 2,113              | Donat                      |
| 4                      | 2,054              | Kopi Hitam                 |
| 5                      | 2,369              | Sushi Makizushi            |
| 6                      | 2,118              | Pizza                      |
| 7                      | 1,726              | Telur mata sapi            |
| 8                      | 2,083              | Telur dadar                |
| 9                      | 1,990              | Red velvet                 |
| 10                     | 2,213              | Nasi goreng                |
| 11                     | 2,157              | Sate                       |
| 12                     | 1,869              | Sayur asem                 |
| 13                     | 1,699              | French fries               |
| 14                     | 1,762              | Macaroni cheese            |
| 15                     | 2,327              | Es pisang ijo              |
| 16                     | 1,926              | Mie ayam                   |
| 17                     | 1,995              | Salmon                     |
| 18                     | 1,961              | Rendang                    |
| 19                     | 1,902              | Pempek                     |
| 20                     | 1,861              | Ikan goreng                |
| <b>Rata-rata</b>       | <b>2,017</b>       |                            |
| <b>Standar deviasi</b> | <b>0,196</b>       |                            |

Pengujian implementasi model pada aplikasi dilakukan pada perangkat keras pribadi yaitu *smartphone* Xiaomi MI 8 dengan versi Android 10. Pengukuran waktu *delay* dilakukan secara manual berdasarkan waktu hasil prediksi ditampilkan

ke layar aplikasi dan pengujian dilakukan terhadap berbagai kategori yang berbeda untuk mengetahui performa sistem deteksi objek pada berbagai kategori.

Hasil pengujian yang dilakukan terlihat pada Tabel 4.8 yang merupakan hasil pengujian delay aplikasi sebanyak 20 kali dan didapatkan rata-rata dan standar deviasi. Hasil tersebut menunjukkan bahwa delay aplikasi secara rata-rata bernilai 2,017 detik, yang meliputi waktu pengiriman *request*, pemrosesan deteksi objek makanan dengan komputasi awan, kemudian penerimaan *response* dan penampilan hasil ke pengguna aplikasi. Jenis pengujian delay aplikasi ini dilakukan untuk mengetahui gambaran umum penggunaan aplikasi jika digunakan secara langsung pada perangkat keras.




Terlihat juga bahwa hasil delay aplikasi yang ada pada berbagai kategori yang diuji tidak memiliki varietas yang terlalu tinggi. Ini terlihat dari nilai standar deviasi antara hasil delay aplikasi hanya memiliki nilai 0,196 yang merupakan nilai yang cukup kecil, menandakan bahwa hasil dapat ditampilkan kepada user dengan waktu delay yang kurang lebih sama untuk berbagai jenis kategori yang berbeda.

#### **4.4.4. Pengujian dengan Sudut Gambar Berbeda-beda**

Untuk dapat menguji tanggapan sistem deteksi objek terhadap situasi berbeda-beda, perlu dilakukan pengujian deteksi objek dengan mempertimbangkan sudut pengambilan gambar yang berbeda-beda. Proses ini dapat dilakukan dengan memvariasikan cara pengambilan gambar secara mandiri maupun pencarian gambar makanan tertentu dari sudut berbeda pada internet.

Proses pengujian ini akan menunjukkan apakah sistem deteksi objek memiliki keterbatasan untuk deteksi pada jenis sudut pengambilan gambar tertentu karena faktor data *training* yang digunakan untuk melatih model. Hasil pengujian dilakukan pada tiga jenis sudut yang berbeda, yaitu horizontal, lancip atau 45 derajat, dan vertikal.

Tabel 4.9. Contoh Pengambilan Gambar untuk Pengujian Sudut Berbeda-beda

| Sudut Pengambilan Gambar | Contoh Gambar  |
|--------------------------|--|
| Horizontal               |    |
| Lancip                   |   |
| Vertikal                 |  |

Pengujian ini dilakukan terhadap jenis makanan yang tidak dipilih secara spesifik, namun mempertimbangkan beberapa hal seperti rupa fisik makanan yang terlihat pada sudut berbeda-beda yang diuji di antara hal lainnya. Makanan yang diuji antara lain adalah pancake, martabak, burger, telur mata sapi, dan french fries.

Tabel 4.10. Hasil Pengujian Deteksi Objek Makanan pada Sudut Gambar yang Berbeda-beda

| <b>Makanan</b>  | <b>Sudut</b> | <b>Hasil deteksi</b> | <b>Confidence score</b> |
|-----------------|--------------|----------------------|-------------------------|
| Pancake         | horizontal   | tidak berhasil       | -                       |
|                 | lancip       | berhasil             | 78%                     |
|                 | vertikal     | berhasil             | 67%                     |
| Martabak        | horizontal   | berhasil             | 88%                     |
|                 | lancip       | berhasil             | 85%                     |
|                 | vertikal     | tidak berhasil       | -                       |
| Burger          | horizontal   | berhasil             | 75%                     |
|                 | lancip       | berhasil             | 79%                     |
|                 | vertikal     | berhasil             | 55%                     |
| Telur mata sapi | horizontal   | tidak berhasil       | -                       |
|                 | lancip       | berhasil             | 81%                     |
|                 | vertikal     | berhasil             | 83%                     |
| French fries    | horizontal   | berhasil             | 68%                     |
|                 | lancip       | berhasil             | 76%                     |
|                 | vertikal     | berhasil             | 78%                     |

Terlihat bahwa secara keseluruhan hasil pengujian mengindikasikan bahwa pada sudut tertentu aplikasi deteksi objek dapat mendeteksi kategori makanan dengan baik, namun pada sudut lain terdapat kesulitan. Hal ini dapat dipengaruhi karena beberapa faktor. Faktor utama adalah karena rupa makanan yang tidak cocok untuk difoto pada sudut tertentu karena sulit didiferensiasikan dengan latar

belakang makanan. Faktor kedua adalah karena data yang digunakan belum mengakomodir sudut yang digunakan pada pengujian. Oleh karena itu, hasil pengujian ini menunjukkan bahwa ada keterbatasan di dalam pengambilan foto makanan pada sudut yang berbeda-beda, yang disebabkan oleh bentuk makanan yang berbeda-beda sehingga untuk data yang digunakan untuk pelatihan sistem deteksi objek makanan pun harus divariasikan.

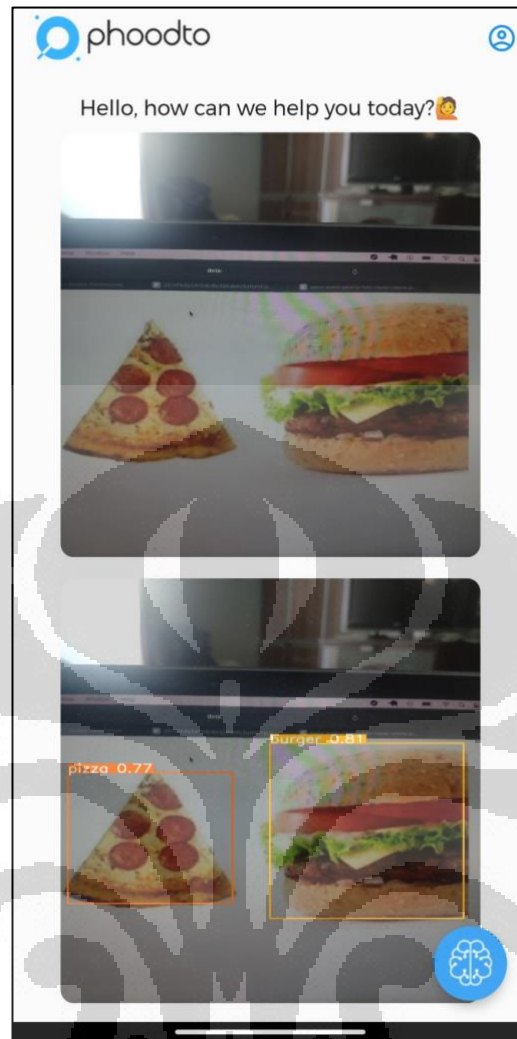
#### 4.4.5. Pengujian Sistem Deteksi Banyak Objek

Untuk dapat menguji bahwa sistem deteksi objek dapat melakukan deteksi terhadap lebih dari satu objek pada waktu yang bersamaan, maka dilakukan pengujian terhadap gambar yang memiliki lebih dari satu kategori makanan tertentu.

Tabel 4.11. Hasil Pengujian Sistem Deteksi Banyak Objek pada Dua Kategori Sekaligus di dalam Satu Gambar

| Pengujian ke- | Makanan              | Confidence score makanan 1 | Confidence score makanan 2 |
|---------------|----------------------|----------------------------|----------------------------|
| 1             | Pizza, burger        | 76%                        | 54%                        |
| 2             | Nasi, sate           | 83%                        | 68%                        |
| 3             | Nasi, telur balado   | 89%                        | 56%                        |
| 4             | Burger, french fries | 78%                        | Tidak terdeteksi           |
| 5             | Doughnut, kopi hitam | 65%                        | Tidak terdeteksi           |

Pengujian dilakukan sebanyak lima kali dan menunjukkan bahwa sistem deteksi objek memiliki kemampuan untuk melakukan deteksi lebih dari satu objek di dalam sebuah gambar. Terlihat pada pengujian satu sampai dengan tiga terdapat hasil confidence score yang didapatkan setelah proses deteksi kategori makanan di dalam gambar berhasil dilakukan. Namun, pada pengujian empat dan lima terlihat bahwa terdapat beberapa kategori yang tidak terdeteksi di dalam sebuah gambar.



Gambar 4.13. Contoh Hasil Pengujian yang Dilakukan untuk Proses Deteksi Banyak Objek

Hasil pengujian ketiga dan keempat dapat dipengaruhi karena banyak faktor, terutama dari sisi pengumpulan data dan juga konteks gambar nyata yang digunakan untuk pengujian. Salah satu faktor yang dapat menyebabkan sebuah kategori tidak terdeteksi adalah karena mayoritas gambar di dalam dataset merupakan gambar yang memiliki satu jenis kategori saja, dan model deteksi objek tidak memiliki banyak contoh situasional saat terdapat lebih dari satu kategori yang seringkali bersama seperti burger dan french fries.

Untuk dapat meningkatkan kemampuan sistem deteksi objek maka diperlukan penambahan data situasional yang memiliki lebih dari satu kategori ketimbang data yang hanya memiliki satu kategori per gambar saja. Jika hal tersebut dilakukan, maka diperkirakan bahwa model deteksi objek akan memiliki tingkat recall yang lebih tinggi terhadap objek kedua maupun keberadaan objek-objek lain

yang mungkin berada di dalam sebuah gambar. Penambahan data situasional dapat dilakukan dengan mempertimbangkan jenis-jenis makanan yang kira-kira bisa berada di dalam suatu gambar tertentu pada waktu yang bersamaan.

#### 4.5. User Experience Testing

Untuk dapat melakukan pengujian UEQ dilakukan pengumpulan data dengan menggunakan kuisisioner Google Forms, dimana setelah periode pemakaian aplikasi seorang user dapat mengisi pengalaman mereka menggunakan aplikasi. Tampilan awal dari kuisisioner yang disebar secara online adalah seperti pada Gambar 4.13. Dalam kuisisioner tersebut awalnya dilakukan pengenalan mengenai aplikasi kemudian dilakukan penyebaran informasi yang sekiranya bisa bermanfaat bagi pengisi kuisisioner seperti jenis makanan yang bisa terdeteksi, kemudian slide presentasi yang menjelaskan fungsionalitas aplikasi dan cara kerjanya.

Setelah membaca pengenalan diri dan aplikasi pada Google Forms, pengisi kuisisioner dapat melakukan pengisian kuisisioner dengan mengikuti petunjuk yang tertera, yang menghimbau bahwa pengisian kuisisioner perlu dilakukan dengan cepat sehingga jawaban yang diberikan terjaga autentitasnya setelah pemakaian aplikasi. Pengisian kuisisioner ini terdiri dari 26 pertanyaan yang memiliki skala 1 sampai 7 dengan penempatan skala jawaban yang berbeda-beda, sehingga penjawab kuisisioner tidak dapat asal mengisi kuisisioner tanpa membacanya terlebih dahulu.

Pengujian *user experience* yang dilakukan pada desain aplikasi Phoodto dengan UEQ dilakukan pada sejumlah 30 orang dengan sebanyak 18 orang laki-laki dan 12 orang perempuan dengan mayoritas rentang umur antara 20-29 tahun. 18 responden berasal dari DKI Jakarta, 7 dari Jawa Barat, 1 dari Jawa Timur, 1 dari Kepulauan Riau, 1 dari Banten, 1 dari Sumatera Utara, dan 1 dari Sumatera Selatan.



**Phoodto Application User Experience Testing Form**

Halo, perkenalkan saya Abraham Kristanto, seorang mahasiswa Teknik Komputer Universitas Indonesia yang sedang melaksanakan riset tugas akhir dengan judul 'Perancangan Sistem Multi-object Detection Untuk Makanan Dengan Implementasi You Only Look Once v5 Dalam Aplikasi Mobile Berbasis Flutter'. Selama 6 bulan terakhir, saya merancang sebuah aplikasi yang menggunakan teknologi artificial intelligence dan machine learning untuk mendeteksi komponen makanan yang berada di dalam suatu gambar. Tujuan dari pembentukan aplikasi ini adalah untuk semakin mengedukasi pengguna aplikasi mengenai diet mereka dan memberikan informasi mengenai makanan yang dikonsumsi sehari-hari.

Komponen makanan yang dapat dideteksi pada aplikasi ini bertotal 40, dan daftarnya dapat dilihat di list berikut:  
<https://docs.google.com/spreadsheets/d/11ROMSKEhVSe8rbcfEWuLK9-ROMy9nFhQqSpck5pEP0g/edit?usp=sharing>

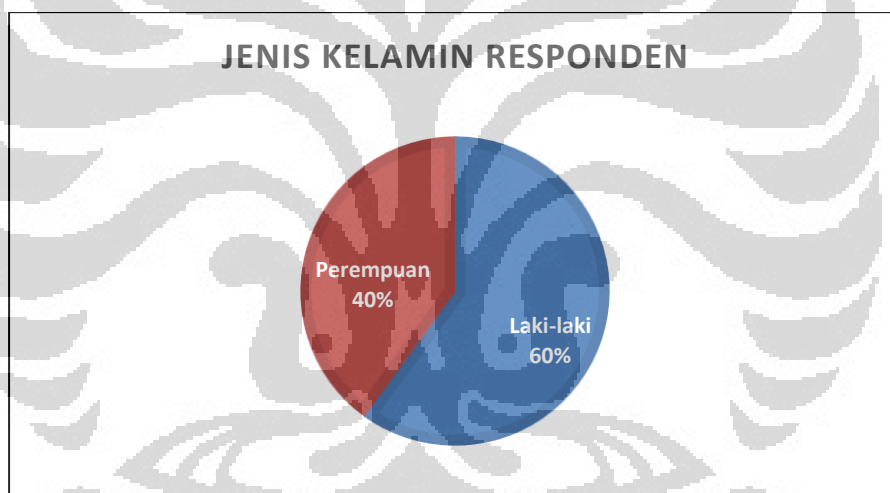
Sebelum menjawab kuisiener berikut, saya harap Anda dapat menonton video singkat berikut mengenai cara berfungsinya aplikasi. Video tersebut dapat dilihat pada tautan berikut maupun pada video yang berada di bawah.  
[https://www.canva.com/design/DAFBgSjLgsk/CkAgrixkxSlmF1Zc-FnWRQ/view?utm\\_content=DAFBgSjLgsk&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAFBgSjLgsk/CkAgrixkxSlmF1Zc-FnWRQ/view?utm_content=DAFBgSjLgsk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

yusuf.abraham@ui.ac.id [Switch account](#)

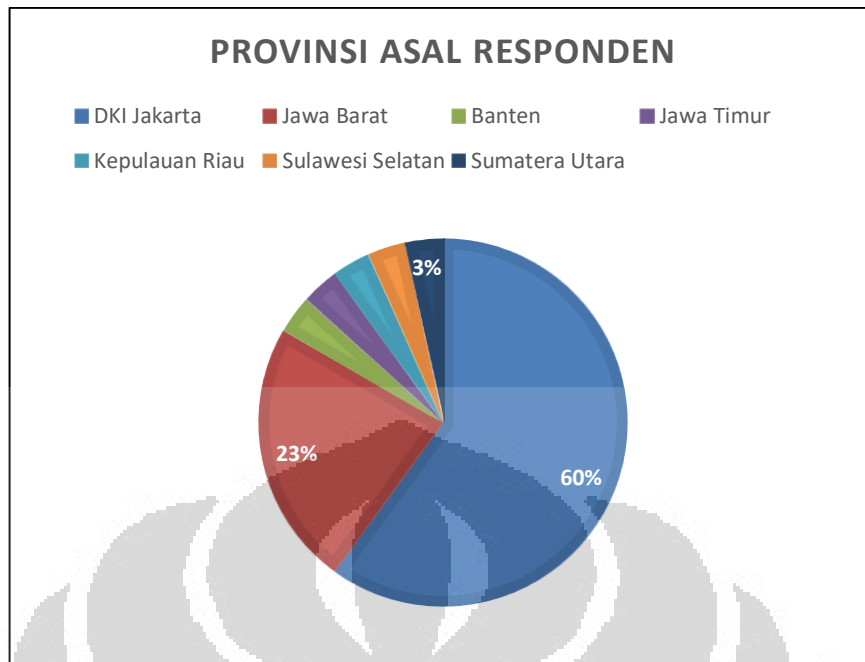
\* Required

Email \*

Gambar 4.14. Tampilan Kuisiener UEQ yang Disebarkan Melalui Google Forms

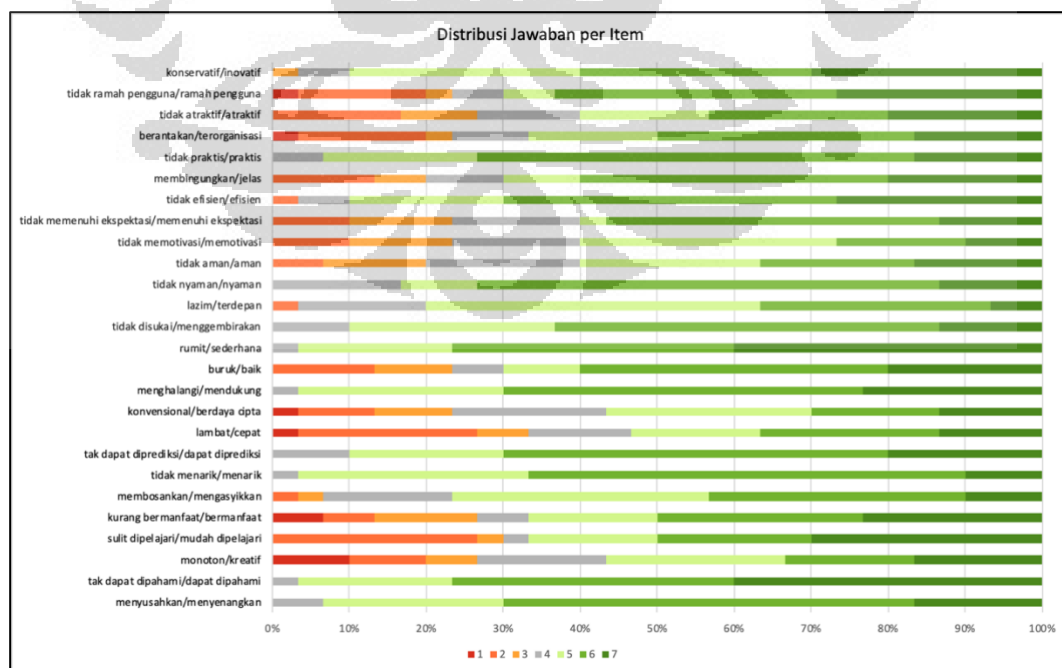


Gambar 4.15. Bagan Persebaran Jenis Kelamin Responden User Experience Questionnaire (UEQ)



Gambar 4.16. Bagan Persebaran Daerah Provinsi Asal Responden User Experience Questionnaire (UEQ)

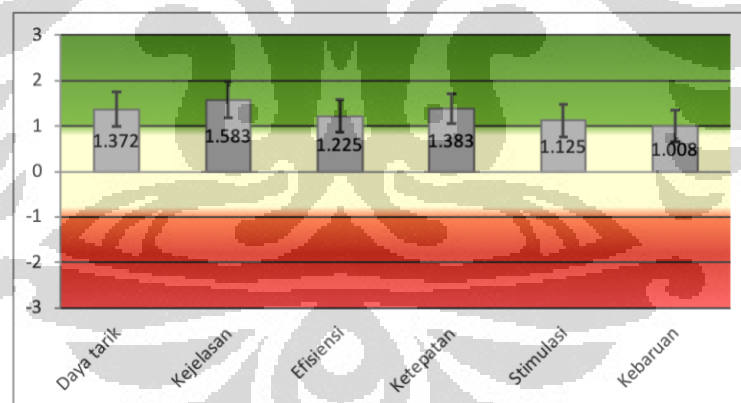
Dalam Gambar 4.16 terlihat persebaran distribusi jawaban per item dari hasil kuisioner yang disebarluaskan. Persebaran tersebut terlihat terdapat pada 26 jenis item berbeda yang masing-masing menjadi pertanyaan di dalam kuisioner tersebut. Untuk dapat melihat gambaran pertanyaan tersebut ke dalam kategori-kategori yang lebih jelas dapat mengacu pada Gambar 4.17.



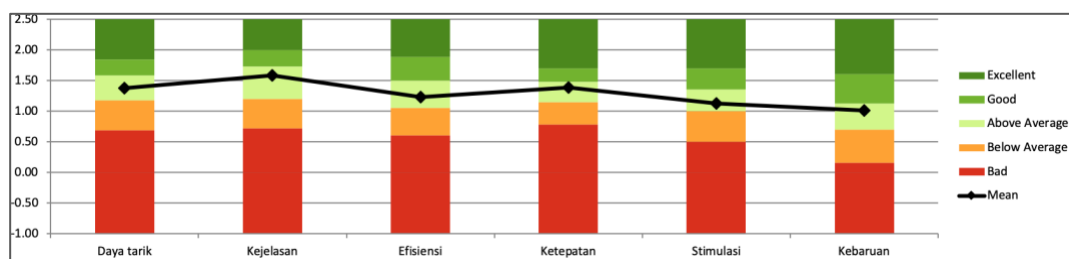
Gambar 4.17. Distribusi Jawaban Hasil Kuisioner UEQ Untuk Tiap Item

Hasil keseluruhan dari UEQ untuk aplikasi Phoodto terlihat di dalam Gambar 4.17 dan diagram untuk benchmark UEQ desain aplikasi Phoodto dapat terlihat pada Gambar 4.18. Terlihat hasil tertinggi menurut hasil kuisisioner bahwa faktor kejelasan memiliki nilai tertinggi dengan nilai 1,583 di antara poin penilaian lainnya. Ini dapat disebabkan karena faktor *user interface* dari aplikasi yang bersifat *to the point* atau langsung ke bagian fungsionalitas aplikasi. Pada peringkat kedua, terdapat poin penilaian ketepatan, yang memiliki nilai 1,383. Ini diikuti dengan daya tarik pada peringkat ketiga dengan nilai 1,372. Berikutnya terdapat efisiensi dengan nilai 1,225 dan stimulasi dengan nilai 1,125. Terakhir terdapat kebaruan dengan nilai 1,008.

Kebaruan memiliki nilai 1,008 kemungkinan karena sudah terdapat teknologi klasifikasi gambar dan deteksi objek yang ada pada aplikasi-aplikasi yang lain dan keterbatasan pemberian informasi makanan terhadap deskripsi saja. Namun, dengan pengembangan fitur lebih lanjut, aplikasi Phoodto dapat menyediakan informasi yang lebih lanjut sehingga bisa memberikan unsur kebaruan yang lebih pada penelitian lebih lanjut.



Gambar 4.18. Rata-rata Nilai UEQ dari Desain Aplikasi Phoodto



Gambar 4.19. Diagram Benchmark UEQ untuk Aplikasi Phoodto

Berdasarkan diagram *benchmark* dari UEQ, aplikasi Phoodto tergolong dalam kategori “*above average*”. Ini berarti bahwa hasil penilaian untuk aplikasi Phoodto masuk ke dalam 50% hasil terbaik dari dataset benchmark. Sehingga, menurut UEQ, kualitas dari aplikasi Phoodto secara keseluruhan memiliki kualitas yang cukup tinggi. Meskipun demikian, tidak ada dari 6 kategori tersebut yang masuk ke dalam rentang “*good*” maupun “*excellent*”, yang masing-masing masuk ke dalam 25% dan 10% hasil terbaik. Sehingga, pengembangan aplikasi Phoodto masih memiliki ruang untuk perbaikan dan pengembangan lebih lanjut.

#### **4.6. Dampak Terhadap Lingkungan Hidup dan Masyarakat**

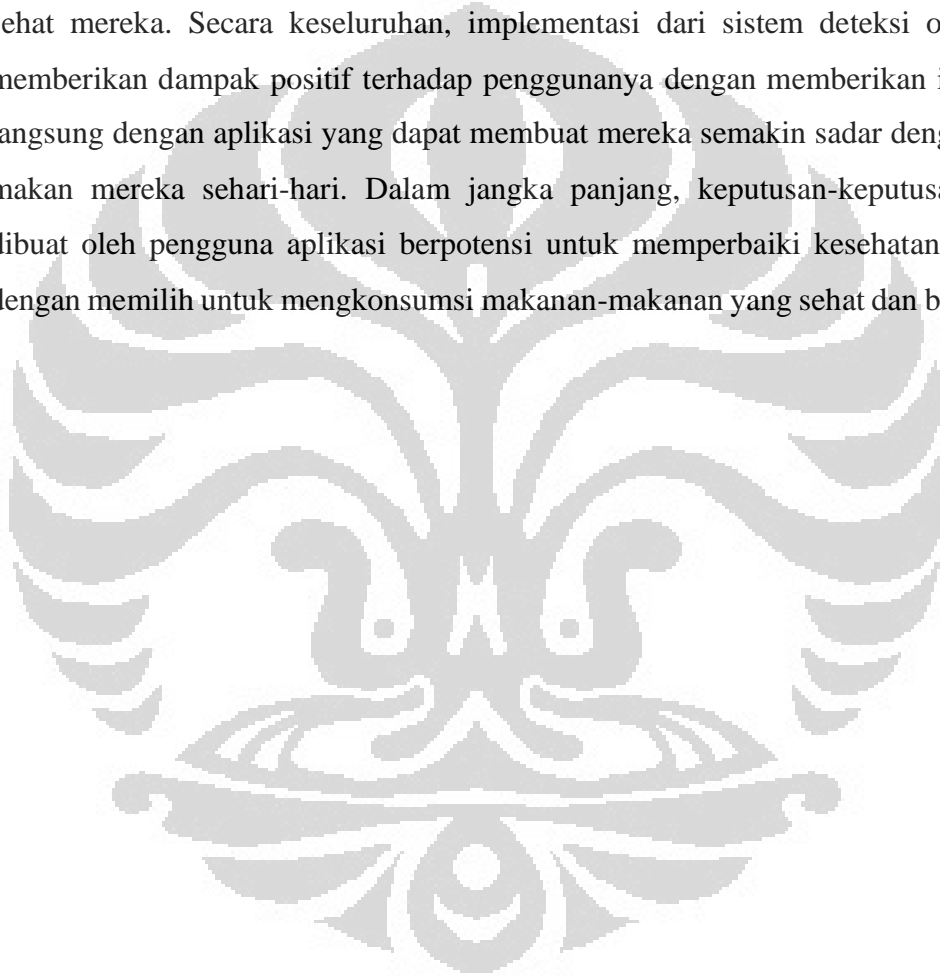
Kesehatan merupakan sesuatu yang sangat penting untuk menjaga keberlangsungan kehidupan. Untuk dapat menunjang kesehatan yang optimal, pilihan makanan yang baik untuk tubuh haruslah menjadi suatu hal yang diperhatikan secara cermat. Kerap kali, karena kesibukan sehari-hari, orang-orang tidak bisa memberikan fokus penuh terhadap apa saja yang dikonsumsi, sehingga ini dapat membangun kebiasaan-kebiasaan buruk dalam keseharian pola makan mereka.

Perancangan sistem deteksi objek makanan yang diimplementasikan di dalam aplikasi mobile Flutter ini mampu memiliki pengaruh terhadap masyarakat melalui pola hidup mereka. Selain dapat memberikan sebuah metode deteksi objek makanan secara langsung, penggunaan aplikasi juga akan memperoleh informasi mengenai makanan yang dikonsumsi tersebut. Manfaat-manfaat yang dapat diberikan oleh aplikasi ini adalah membantu pengguna mengenal makanan yang dikonsumsi, membantu pengguna dengan keterbatasan makanan karena kondisi kesehatan ataupun alergi makanan, dan memberikan edukasi singkat mengenai makanan yang dikonsumsi.

Pada saat ini aplikasi ini memiliki sebuah keterbatasan karena jumlah objek makanan yang dapat dideteksi hanya berjumlah sebanyak 40 kategori. Ini dipengaruhi karena faktor keterbatasan waktu dalam pengumpulan data dan juga usaha pelabelan data yang dilakukan secara mandiri. Namun, ini tidak menutup

kemungkinan bahwa dengan perluasan dataset implementasi semacam ini dapat memiliki dampak yang lebih besar kepada masyarakat secara luas.

Melalui riset ini, rancang bangun sistem deteksi objek makanan dalam bentuk aplikasi dapat menjadi bukti konkret bahwa aplikasi ini dapat memberikan manfaat bagi orang lain. Potensi penggunaan aplikasi ini tidak hanya untuk orang-orang yang memiliki keterbatasan saja, namun juga bagi orang-orang yang ingin mengetahui pola makannya lebih lanjut sehingga dapat meningkatkan pola makan sehat mereka. Secara keseluruhan, implementasi dari sistem deteksi objek ini memberikan dampak positif terhadap penggunaanya dengan memberikan interaksi langsung dengan aplikasi yang dapat membuat mereka semakin sadar dengan pola makan mereka sehari-hari. Dalam jangka panjang, keputusan-keputusan yang dibuat oleh pengguna aplikasi berpotensi untuk memperbaiki kesehatan mereka dengan memilih untuk mengonsumsi makanan-makanan yang sehat dan bernutrisi.



## **BAB 5**

### **KESIMPULAN**

#### **5.1. Kesimpulan**

Berdasarkan pemaparan laporan ini terdapat beberapa hal yang dapat disimpulkan mengenai pengembangan pendeteksi makanan dengan deteksi objek YOLO, yakni:

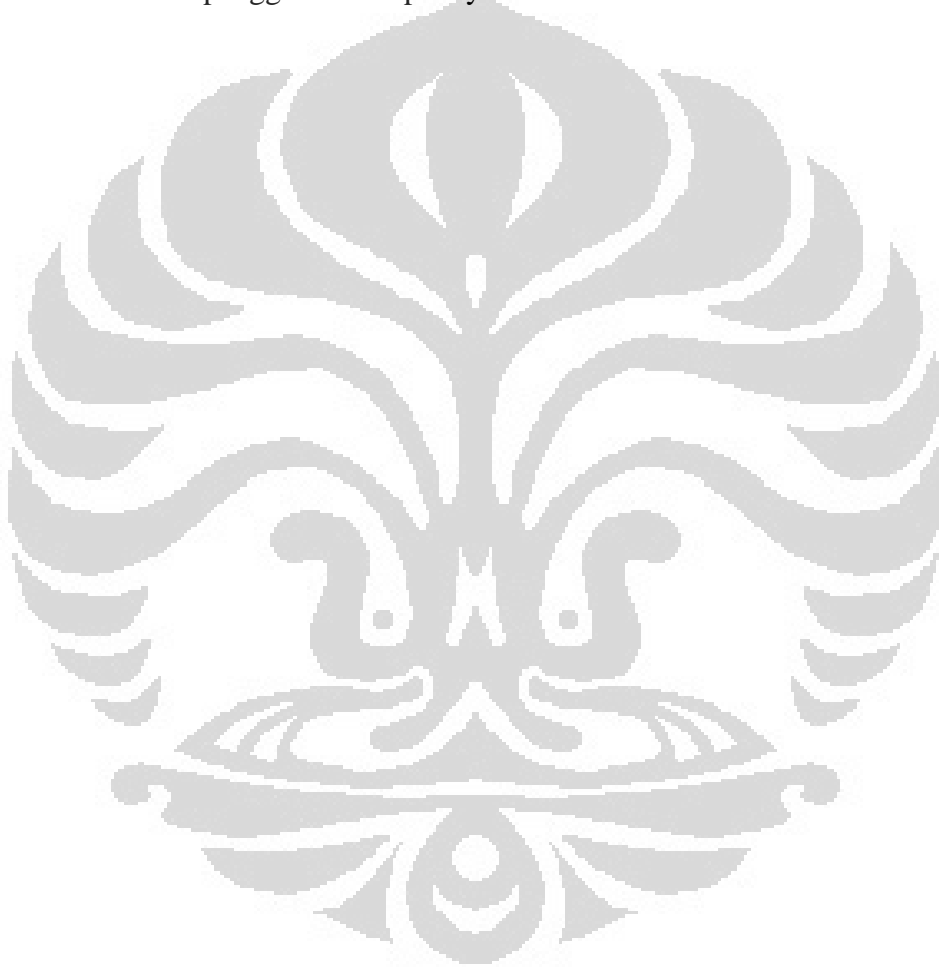
1. Perancangan model deteksi objek menggunakan algoritma YOLOv5 berhasil dilakukan dan dilatih pada dataset yang dirancang secara mandiri dengan total 5574 gambar mencakup 40 kategori.
2. Pengimplementasian sistem deteksi objek makanan berhasil dilakukan pada *cloud* dengan menggunakan Python dan hasil deteksi dapat ditampilkan kepada pengguna aplikasi yang dirancang dengan menggunakan Flutter.
3. Pengujian performa pada model sebelum augmentasi mencapai rata-rata hasil akhir Mean Average Precision (mAP) sebesar 0,85 dengan Precision 0,8 dan Recall 0,8.
4. Pengujian performa model akhir setelah dilatih pada dataset teraugmentasi mencapai nilai Mean Average Precision (mAP) sebesar 0,943 dengan nilai Precision 0,889 dan Recall 0,904.
5. Perancangan dan pengembangan aplikasi *frontend* dengan Flutter berhasil dilakukan dengan hasil survei kuisisioner UEQ mengindikasikan bahwa aplikasi Phoodto yang dikembangkan memiliki hasil 50% teratas, sehingga pengalaman menggunakan aplikasi tergolong di atas rata-rata.

#### **5.2. Saran**

Setelah perancangan dan implementasi sistem deteksi objek makanan, terdapat beberapa saran yang dapat diberikan untuk penelitian ke depannya, yaitu:

1. Meningkatkan jumlah kategori dan jumlah data dari dataset yang digunakan untuk melatih sistem deteksi objek makanan, sehingga jenis makanan yang dapat terdeteksi berjumlah lebih banyak dengan akurasi yang lebih tinggi.

2. Menambah deskripsi makanan yang terdeteksi sehingga pengguna aplikasi tidak hanya terbatas pada definisi dan gambaran makanan saja namun pengguna aplikasi mendapatkan ilmu mengenai kandungan nutrisi makanan yang terdeteksi.
3. Memperluas fitur aplikasi sehingga dapat mengintegrasikan sistem deteksi objek dengan melacak keseharian pola makanan pengguna aplikasi. Informasi tersebut dapat dicatat di dalam aplikasi dan digunakan untuk referensi pengguna ke depannya.



## DAFTAR ACUAN

- [1] K. Ha, J. R. Sakaki, and O. K. Chun, “Nutrient Adequacy Is Associated with Reduced Mortality in US Adults,” *The Journal of Nutrition*, Aug. 2021, doi: 10.1093/jn/nxab240.
- [2] D. D. Headey and H. H. Alderman, “The Relative Caloric Prices of Healthy and Unhealthy Foods Differ Systematically across Income Levels and Continents,” *Journal of Nutrition*, vol. 149, no. 11, pp. 2020–2033, Nov. 2019, doi: 10.1093/jn/nxz158.
- [3] J. Sun, K. Radecka, and Z. Zilic, “FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks,” Sep. 2019, [Online]. Available: <http://arxiv.org/abs/1909.05994>
- [4] C. Kiourt, G. Pavlidis, and S. Markantonatou, “Deep Learning Approaches in Food Recognition,” 2020, pp. 83–108. doi: 10.1007/978-3-030-49724-8\_4.
- [5] J. Gu *et al.*, “Recent Advances in Convolutional Neural Networks,” Dec. 2015, [Online]. Available: <http://arxiv.org/abs/1512.07108>
- [6] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4. Springer Verlag, pp. 611–629, Aug. 01, 2018. doi: 10.1007/s13244-018-0639-9.
- [7] P. Malhotra and E. Garg, “Object Detection Techniques: A Comparison,” in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Jul. 2020, pp. 1–4. doi: 10.1109/ICSSS49621.2020.9202254.
- [8] S. Sahil *et al.*, “A Survey of Modern Deep Learning based Object Detection Models.”
- [9] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” May 2014, [Online]. Available: <http://arxiv.org/abs/1405.0312>



- [10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Jun. 2015, [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [12] L. Jing and Y. Tian, “Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey,” Feb. 2019, [Online]. Available: <http://arxiv.org/abs/1902.06162>
- [13] U. Nepal and H. Eslamiat, “Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs,” *Sensors*, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020464.
- [14] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A forest fire detection system based on ensemble learning,” *Forests*, vol. 12, no. 2, pp. 1–17, Feb. 2021, doi: 10.3390/f12020217.
- [15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” Aug. 2016, [Online]. Available: <http://arxiv.org/abs/1608.06993>
- [16] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” Nov. 2019, [Online]. Available: <http://arxiv.org/abs/1911.11929>
- [17] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation.”
- [18] M. Tan, R. Pang, and Q. v. Le, “EfficientDet: Scalable and Efficient Object Detection,” Nov. 2019, [Online]. Available: <http://arxiv.org/abs/1911.09070>
- [19] “Precision-Recall,” Dec. 15, 2021. [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_precision\\_recall.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html) (accessed Dec. 15, 2021).

- [20] H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th edition. 2019.
- [21] B. Laugwitz, T. Held, and M. Schrepp, “Construction and Evaluation of a User Experience Questionnaire,” 2008, pp. 63–76. doi: 10.1007/978-3-540-89350-9\_6.
- [22] N. Yuniar, “Kemendikbudristek beri pendanaan untuk tim terbaik Bangkit 2021,” *Antara News*, Sep. 08, 2021.
- [23] Q. Cai, J. Li, H. Li, and Y. Weng, “BTBUFood-60: Dataset for Object Detection in Food Field,” in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb. 2019, pp. 1–4. doi: 10.1109/BIGCOMP.2019.8678916.
- [24] A. Ramesh, A. Sivakumar, and S. Sherly Angel, “Real-time Food-Object Detection and Localization for Indian Cuisines using Deep Neural Networks,” Dec. 2020. doi: 10.1109/ICMLANT50963.2020.9355987.
- [25] C. Kumar B., R. Punitha, and Mohana, “YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications,” in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Aug. 2020, pp. 1316–1321. doi: 10.1109/ICSSIT48917.2020.9214094.