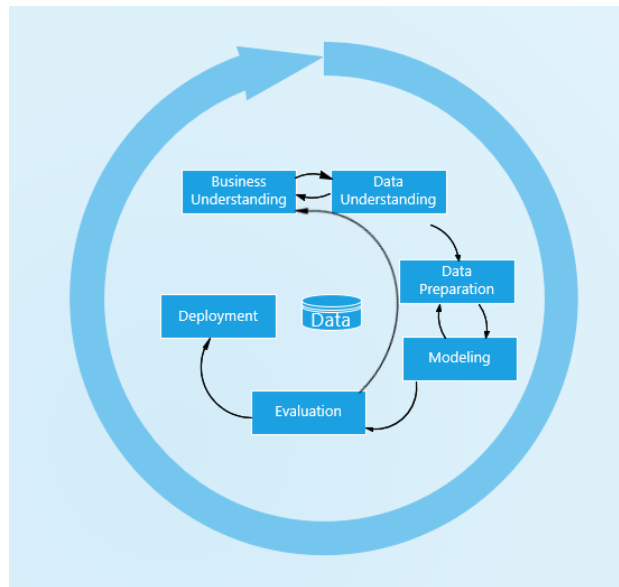


## Bab III

### METODOLOGI PENELITIAN



Gambar 3.1 Proses Metodologi Penelitian

#### 3.1 Pemahaman Bisnis

Label kandungan gizi yang tertera pada produk pangan berperan penting dalam kegiatan berbelanja. Pembeli dapat mengetahui informasi gizi pada barang yang ingin mereka beli, dan penjual pun dapat memanfaatkan informasi tersebut untuk mengategorikan produk yang dijual. Meskipun memiliki kelebihan tersebut, berbagai aspek kemasan dari produk pangan tersebut dapat membuat label kandungan gizi menjadi sulit untuk dibaca. Selain itu, hal ini menyebabkan pembeli perlu menggunakan sedikit waktu untuk mencari posisi label kandungan gizi yang tertera pada produk yang menarik perhatian mereka. Oleh karena itu, penelitian ini dilakukan untuk membuat sebuah aplikasi yang dapat membantu meningkatkan kenyamanan dan efektivitas ketika berbelanja.

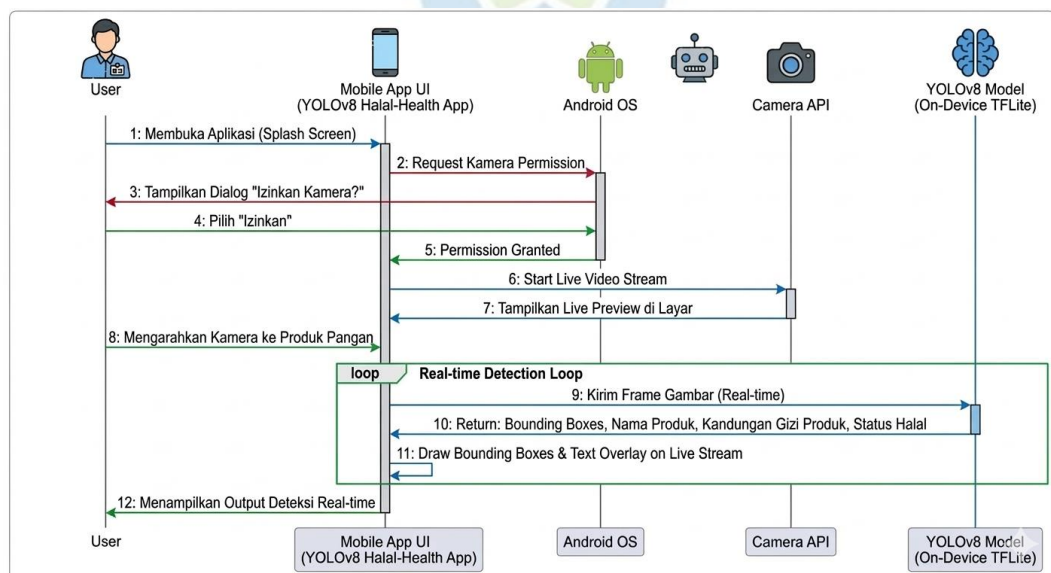
Aplikasi yang dikembangkan pada penelitian ini menggunakan citra sebagai input, dan mengintegrasikan *Augmented Reality* pada bagian outputnya. Algoritma YOLO digunakan pada model ini dikarenakan algoritma tersebut memiliki kinerja yang baik untuk mendeteksi objek. *Augmented Reality* digunakan agar aplikasi dapat “menambahkan” informasi digital ke lingkungan nyata secara *realtime*, sehingga dapat memberikan pengalaman memilih makanan yang lebih nyaman.

Tujuan dari pembuatan aplikasi deteksi produk pangan pada penelitian ini yaitu memberikan pengalaman memilih makanan bergizi yang lebih nyaman dan mudah. Pengembangan aplikasi ini diharapkan dapat memberikan dorongan kepada pengguna agar mengkonsumsi makanan yang bergizi dan hidup sehat. Aplikasi ini juga dapat berpotensi menggantikan label harga pada pusat perbelanjaan yang tertarik untuk digitalisasi proses perbelanjaan jika dikembangkan untuk penggunaan tersebut.

Untuk mengembangkan aplikasi yang dapat memenuhi tujuan tersebut, pemahaman yang baik akan kebutuhan aplikasi sangat diperlukan. Agar dapat mengidentifikasi kebutuhan secara rinci, analisis kebutuhan aplikasi perlu dilakukan sebelum memulai pengembangan aplikasi. Hal ini dilakukan agar aplikasi dapat berjalan sesuai dengan kebutuhan dan memiliki kinerja yang baik.

### 3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional yang diperlukan oleh aplikasi deteksi produk pangan berbasis *Augmented Reality* dapat divisualisasikan menggunakan UML seperti pada gambar .



Gambar 3.2 UML Aplikasi

UML gambar 3.1 memberikan penjelasan mengenai alur aplikasi yang akan dibuat. Aplikasi akan meminta permission kamera karena kamera merupakan bagian penting dari fungsi deteksi produk secara real-time. Setelah mendapatkan

permission, user dapat menggunakan aplikasi untuk mendeteksi produk yang ingin di cek, lalu aplikasi akan menggunakan input tersebut untuk melalui proses deteksi objek, dan memberikan output berupa bounding boxes dan informasi produk secara real-time.

### 3.1.2 Kebutuhan Non Fungsional

Kebutuhan perangkat lunak yang akan digunakan untuk proses pengembangan aplikasi dan pengujian aplikasi dapat dilihat pada tabel 3.1 dan 3.2.

Tabel 3.1 Kebutuhan Perangkat Pemrograman

Jenis Perangkat	Nama Perangkat	
Perangkat Keras	Prosesor	Ryzen 7 3750H
	VGA	Nvidia GTX 1660 TI
	RAM	16 GB
	SSD	512 GB
Perangkat Lunak	Sistem Operasi	Windows 11 Home Edition
	Bahasa Pemrograman	Python, Kotlin
	<i>Code Editor</i>	Google Colab, Android Studio Code

Tabel 3.2 Kebutuhan Perangkat Aplikasi

Jenis Perangkat	Nama Perangkat	
Perangkat Keras	CPU	Helio G100 Ultimate
	Front Camera	50M OIS + 8M
	RAM	8 GB
	Storage	256 GB
Perangkat Lunak	Sistem Operasi	Android Version 15

### 3.2 Pemahaman Data

Pemahaman data merupakan tahap pengumpulan data citra yang akan diproses menjadi dataset yang akan digunakan pada proses pelatihan dan pengujian model. Data yang dikumpulkan merupakan data citra yang berisi

produk pangan olahan. Dataset yang akan digunakan merupakan data primer yang didapatkan dan diproses sendiri oleh penulis, sehingga dataset tersebut belum pernah dipublikasikan sebelumnya.

Data citra yang akan digunakan terdiri dari 11 kelas, yang terdiri dari 5 jenis mie instant, 4 jenis minuman bubuk instan, dan 2 makanan ringan. Masing-masing kelas terdiri dari 5 data citra, sehingga menghasilkan 55 data citra mentah. Produk mie instan terdiri dari varian extra hot, thai, sambal belacan, dan Korean spicy. Produk minuman bubuk instan terdiri dari chocolatos, creamy latte, cappuccino, dan mocafrio. Produk makanan ringan terdiri dari malkist dan wafello. Jenis kelas tersebut dipilih berdasarkan karakteristik kemasan, seperti kemiripan antara mie instant, perbedaan kemasan untuk setiap jenis produk pangan olahan, dan ukuran dari produk pangan olahan yang dipilih. Pemilihan ini dilakukan untuk mendapatkan keberagaman karakteristik pada data citra, sehingga dapat menghasilkan model data yang baik.

Pengambilan data citra dilakukan melalui proses pemotretan secara manual dengan menggunakan berbagai macam latar belakang agar model yang dilatih dapat terhindar dari *overfitting background* dan bias. Selain itu, penggunaan latar belakang yang beragam juga dapat meningkatkan ketangguhan dan generalisasi model yang dihasilkan. Setelah melalui proses pengambilan data citra, setiap kelas terdiri dari 5 data citra.

### **3.3 Persiapan Data**

Dataset yang telah disiapkan masih berupa dataset mentah, yang harus melalui beberapa persiapan sebelum dapat digunakan untuk training model. Pada tahapan ini, penggunaan platform *Roboflow* dapat membantu mempermudah dan mempercepat alur kerja persiapan data.

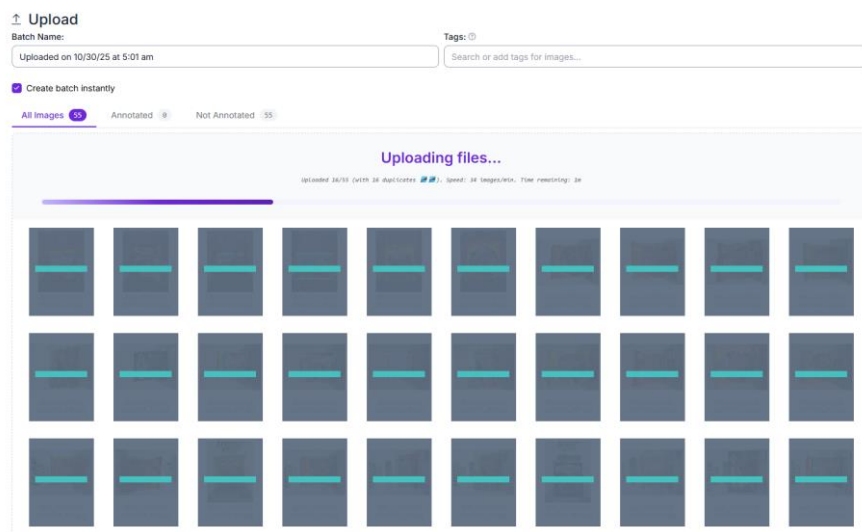
#### **3.3.1 Anotasi Dataset**

Dataset yang akan digunakan perlu diberikan anotasi yang berupa *bounding box* pada setiap objek yang terdapat pada data citra. Penggunaan platform *Roboflow* mempermudah proses anotasi, sehingga dapat dilakukan

dengan mudah dan dalam waktu singkat. Objek yang akan di anotasi dapat dikelompokkan menjadi 11 kelas produk pangan olahan, yaitu :

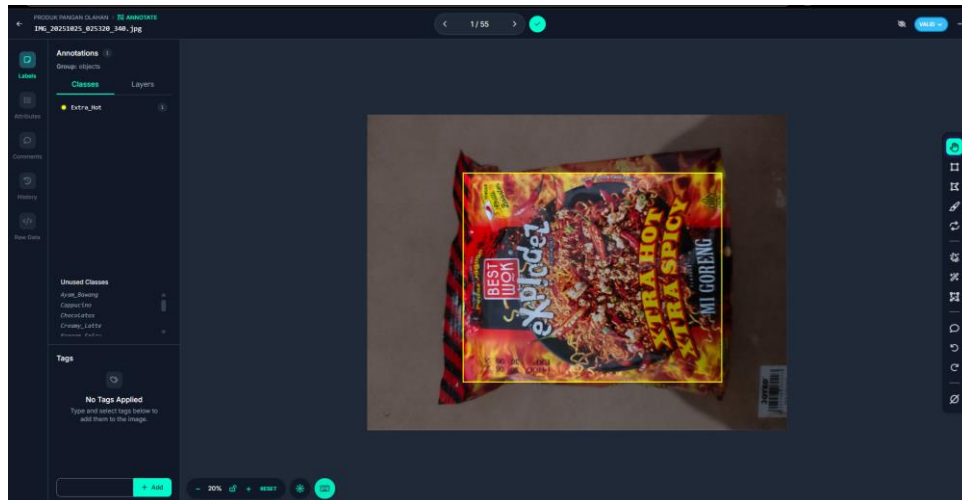
- 1) Ayam Bawang
- 2) Cappucino
- 3) Chocolatos
- 4) Creamy Latte
- 5) Extra Hot
- 6) Korean Spicy
- 7) Malkist
- 8) Mocafrio
- 9) Sambal Belacan
- 10) Thai Chili
- 11) Wafello

Tahap pertama yang perlu dilakukan adalah mengunggah dataset pada *Roboflow* seperti pada gambar 3.1



Gambar 3.3 Proses Unggah Data

Jika proses pengunggahan telah selesai, maka tahap anotasi bisa dimulai. Anotasi data dilakukan dengan memberikan *bounding box* sekaligus pembagian kelas objek yang ditunjukkan pada gambar 3.2.



Gambar 3.4 Tahap Anotasi Data

### 3.3.2 Pembagian Data

Data citra yang telah diberikan anotasi perlu dibagi menjadi tiga set: data latih (82%), data validasi (9%), data uji (9%). Pembagian ini dilakukan agar setiap kelas memiliki data validasi dan data tes. Pembagian ini dilakukan agar model dapat diuji menggunakan data yang tidak digunakan saat pelatihan model.

### 3.3.3 *Pre-processing* dan *Augmentasi*

Dataset yang telah diberi anotasi masih belum siap digunakan untuk pelatihan model YOLOv8n. Dataset tersebut perlu melalui tahap *pre-processing* dan *augmentasi*. Tahap tersebut dilakukan untuk meningkatkan ketahanan dan generalisasi model pelatihan.

*Pre-processing* yang akan digunakan pada dataset terdiri dari dua tahap. Tahap pertama adalah *auto-orient* yang berfungsi untuk memastikan data citra tersebut memiliki orientasi yang benar. Tahap kedua adalah *resize* yang berfungsi untuk mengubah ukuran data citra menjadi 640x640 piksel agar sesuai dengan standar input YOLOv8n. *Roboflow* memberikan opsi tersebut seperti yang ditunjukkan pada gambar 3.3.

3

### Preprocessing

Decrease training time and increase performance by applying image transformations to all images in this dataset.

<b>Auto-Orient</b>	<a href="#">Edit</a>	×
<b>Resize</b> Stretch to 640×640	<a href="#">Edit</a>	×
<b>+ Add Preprocessing Step</b>		

**Continue**

Gambar 3.5 Tahap *Pre-Processing*

Setelah *pre-processing*, data citra akan melalui tahapan augmentasi untuk memperbanyak variasi data tanpa perlu mengumpulkan gambar tambahan secara manual. Teknik augmentasi yang akan diterapkan meliputi beberapa teknik transformasi yang ditunjukkan pada gambar 3.4, diantaranya adalah:

4

### Augmentation

What can augmentation do?

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

<b>Flip</b> Horizontal, Vertical	<a href="#">Edit</a>	×
<b>90° Rotate</b> Clockwise, Counter-Clockwise, Upside Down	<a href="#">Edit</a>	×
<b>Crop</b> 0% Minimum Zoom, 20% Maximum Zoom	<a href="#">Edit</a>	×
<b>Rotation</b> Between -10° and +10°	<a href="#">Edit</a>	×
<b>Shear</b> ±10° Horizontal, ±10° Vertical	<a href="#">Edit</a>	×
<b>Grayscale</b> Apply to 15% of images	<a href="#">Edit</a>	×
<b>Brightness</b> Between -15% and +15%	<a href="#">Edit</a>	×
<b>Blur</b> Up to 1px	<a href="#">Edit</a>	×
<b>Noise</b> Up to 0.3% of pixels	<a href="#">Edit</a>	×
<b>+ Add Augmentation Step</b>		
<b>Use Previous Augmentations</b> Use augmentations from a previous version.		

**Continue**

Gambar 3.6 Tahap Augmentasi Data

- *Flip*: Data citra akan dibalik secara horizontal atau vertical, bertujuan sebagai simulasi kondisi objek dari perspektif lain.
- Rotasi 90 Derajat: data citra akan dirotasi 90 derajat sesuai dengan arah jarum jam, berlawanan jarum jam, atau dibalik secara vertical. Bertujuan untuk memberikan perspektif lain pada objek.
- *Crop*: Data citra akan di crop sebanyak 0 hingga 20 persen. Hal ini bertujuan sebagai simulasi jarak antara kamera dengan objek.
- Rotasi -10 Hingga 10 Derajat dan *Shear*: data citra akan dirotasi atau diberikan *shear* secara acak antara -10 hingga 10 derajat, bertujuan untuk memberikan variasi distorsi perspektif.
- *Grayscale*: data citra akan diberikan efek *grayscale*, bertujuan untuk melatih model tidak terlalu bergantung pada warna, tetapi pada fitur lainnya.
- *Brightness*: data citra akan diberikan tingkat kecerahan diantara -15 hingga 15%, bertujuan agar model dapat mengenali objek dengan tingkat kecerahan yang berbeda.
- *Noise* dan *Blur*: data citra akan diberikan efek *salt and pepper noise* dan juga *blur*. Penambahan *noise* bertujuan agar model lebih tahan terhadap cacat visual. *Blur* dapat membantu model untuk lebih tahan terhadap fokus kamera.

Proses augmentasi ini menghasilkan tiga versi berbeda dari setiap data citra pada dataset tersebut, sehingga akan menghasilkan model data yang lebih tangguh.

### 3.4 Pemodelan

Setelah dataset siap, tahap selanjutnya yang perlu dilakukan adalah melakukan pemodelan. Pemodelan memiliki beberapa hal penting yang perlu diperhatikan, agar dapat menghasilkan model yang baik.

#### 3.4.1 Pemilihan Model

YOLOv8n adalah sebuah model yang memiliki keseimbangan antara akurasi dan kecepatan pengambilan kesimpulan [28]. Hal ini merupakan faktor penting untuk sebuah aplikasi *mobile* yang menerima *input* secara *real-time*. YOLOv8n (nano) dipilih karena memiliki arsitektur yang ringan, cocok untuk

aplikasi *mobile*. Selain itu, YOLOv8n juga merupakan sebuah model yang telah dilatih menggunakan dataset COCO, sehingga cocok untuk pelatihan model dengan jumlah data citra yang sedikit.

### 3.4.2 Transfer Learning

Proses pelatihan model data yang dilakukan dari awal, memungkinkan model memiliki tingkat akurasi yang rendah dan tingkat ketahanan kurang baik. Akan tetapi, *Transfer Learning* mempermudah proses pelatihan tanpa perlunya dataset yang besar. Oleh karena itu, penggunaan YOLOv8n yang telah dilatih sebelumnya menggunakan COCO dataset dapat membantu menghasilkan model dengan performa yang baik [29].

### 3.4.3 Pelatihan Model

Pelatihan model menggunakan *Google Colaboratory* mempermudah proses instalasi maupun proses pelatihan model. *Google Colaboratory* dapat diakses secara langsung menggunakan *browser* pilihan, sehingga tidak memerlukan proses instalasi aplikasi tambahan selain *browser* tersebut. *Library* yang penting dalam proses training ini adalah *ultralytics*. *Library* ini digunakan untuk melakukan pelatihan dan *fine-tuning* model. Sebelum memulai pelatihan model, perlu dilakukan pengunduhan dataset yang sebelumnya telah disiapkan pada platform *Roboflow*. *Roboflow* memiliki fitur pengunduhan dataset dalam bentuk zip secara langsung, atau bisa menggunakan kode pengunduhan. Karena pelatihan model menggunakan *Google Colaboratory*, penggunaan kode pengunduhan merupakan pilihan yang akan digunakan. Implementasi penggunaan kode pengunduhan ditunjukkan pada gambar 3.5.



pelatihan ini adalah *learning rate*, *batch size*, dan *epoch*. Nilai parameter yang digunakan ditentukan pada Tabel 3.3.

Tabel 3.3 Nilai Parameter

Nomor Pelatihan	Learning Rate	Batch Size	Epoch
1	0.0001	16	20
2	0.0001	16	40
3	0.0001	16	60
4	0.0001	16	80
5	0.0001	16	100

Setelah melatih model sesuai dengan parameter pada Tabel 3.3, pengujian model dilakukan secara otomatis menggunakan fitur yang telah disediakan oleh *ultralitics*. Pengujian dilakukan pada data uji yang berjumlah 11 citra. Kode yang digunakan untuk pengujian otomatis model diimplementasikan pada Gambar 3.7.

1	<code>!yolo task=detect mode=predict model='runs/detect/train/weights/best.pt' source='/content/Produk-Pangan-Olahan-5/test/images'</code>
---	--

Gambar 3.9 Pengujian Otomatis Model

Hasil pengujian model ditunjukkan pada Gambar 3.8.



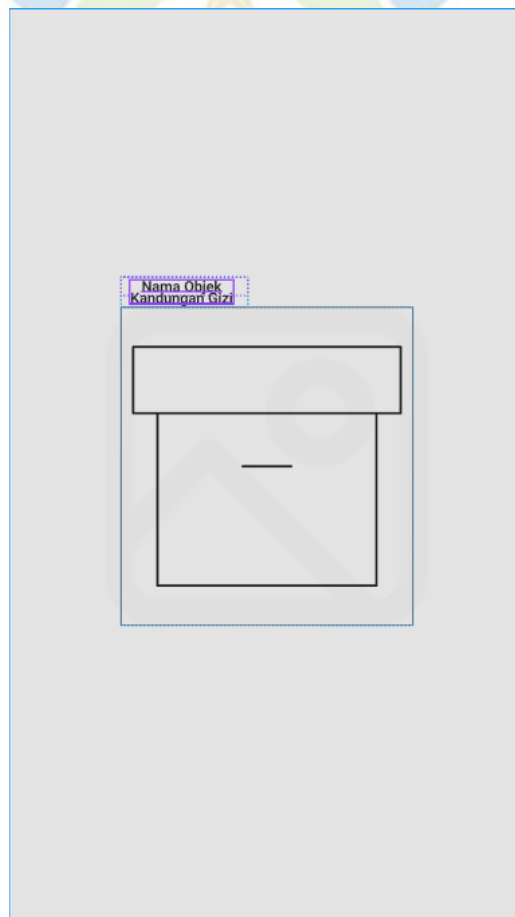
Gambar 3.10 Hasil Pengujian Model

### 3.6 Penyebaran

Model terbaik yang dihasilkan oleh pelatihan model pada tahap sebelumnya akan digunakan untuk mengembangkan aplikasi berbasis android. Pengembangan ini bertujuan agar model dapat lebih mudah untuk digunakan.

#### 3.6.1 Desain Antarmuka Halaman Utama

Aplikasi ini memiliki desain antarmuka yang simple, sehingga mempercepat akses pengguna untuk memulai prediksi produk yang diinginkan. Halaman utama terdiri dari tiga komponen, yaitu pratinjau kamera dan *header*, dan *bounding box*. *Header* berisikan informasi produk pangan seperti nama dan kandungan gizi. Pratinjau kamera berfungsi sebagai komponen yang menampilkan pratinjau kamera dan hasil prediksi. *Bounding Box* berfungsi untuk menandai objek yang berhasil diprediksi serta menjadi patokan bagi *header*. Desain antarmuka halaman utama ditunjukkan pada Gambar 3.9.



Gambar 3.11 Desain Antarmuka Halaman Utama