

**RANCANG BANGUN APLIKASI DETEKSI JAMUR PADA ROTI  
BERBASIS MOBILE**

**TUGAS AKHIR**

**Diajukan oleh:**

**MUHAMMAD GHAZI SIDDIQ**

**NIM. 190705044**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2023 M / 1445 H**

**LEMBAR PERSETUJUAN**

**RANCANG BANGUN APLIKASI DETEKSI JAMUR PADA ROTI  
BERBASIS MOBILE**

**TUGAS AKHIR**

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)  
dalam Ilmu Teknologi Informasi

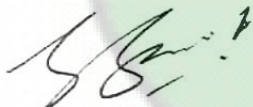
**Oleh:**

**MUHAMMAD GHAZI SIDDIQ  
NIM. 190705044**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui untuk Dimunajasyahkan Oleh:

**Pembimbing I,**



**Khairan AR, M.Kom  
NIP. 198607042014031001**

**Pembimbing II,**



**Rahmat Musfikar, M.Kom  
NIP. 198909132020121015**

Mengetahui

Ketua Program Studi Teknologi Informasi



**Ima Dwitawati, M.B.A  
NIP. 198210132014032002**

**LEMBAR PENGESAHAN**

**RANCANG BANGUN APLIKASI DETEKSI JAMUR PADA ROTI BERBASIS  
MOBILE**

**TUGAS AKHIR**

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan

.....

Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)  
Dalam Prodi Teknologi Informasi

Pada Hari/Tanggal : Selasa, 7 November 2023  
22 Rabi'ul Akhir 1445H

di Darussalam, Banda Aceh  
Panitia Ujian Munaqasah Tugas Akhir

Ketua,



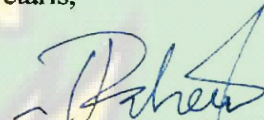
Khairan AR, M.Kom  
NIP. 198607042014031001

Penguji I,



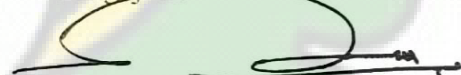
Hendri Ahmadian, S.Si., M.I.M  
NIP. 198301042014031002

Sekretaris,



Rahmat Musfikar, M.Kom  
NIP. 198909132020121015

Penguji II,



Firmansyah, S.Kom., M.T  
NIP. 198704212015031002

Mengetahui:

Dekan Fakultas Sains dan Teknologi  
UIN Ar-Raniry Banda Aceh,



Dr. Ir. Muhammad Dirhamsyah, M.T., IPU  
NIP. 196210021988111001

## LEMBAR PERNYATAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Muhammad Ghazi Siddiq  
Nim : 190705044  
Program studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul : Rancang bangun aplikasi deteksi jamur pada roti berbasis mobile

Dengan ini menyatakan bahwa dalam penulisan skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh,

Yang Menyatakan,



Muhammad Ghazi Siddiq

## ABSTRAK

Nama : Muhammad Ghazi Siddiq  
NIM : 190705044  
Program studi : Teknologi Informasi  
Judul : Rancang bangun aplikasi deteksi jamur pada roti berbasis mobile  
Tanggal Sidang : Selasa, 7 November 2023  
Jumlah Halaman : 48  
Pembimbing I : Khairan AR, M.Kom.  
Pembimbing II : Rahmat Musfikar, M.Kom  
Kata Kunci : Jamur, Deteksi Jamur, Mobile, YOLO, Tensorflow

Jamur dapat muncul karena beberapa faktor seperti kondisi lingkungan yang kurang baik, sanitasi yang buruk, kualitas bahan baku yang rendah, dan lain-lain. Hal ini menyebabkan jamur menjadi masalah yang serius dalam industri roti. Selain menurunkan kualitas roti, jamur juga dapat menimbulkan risiko kesehatan bagi konsumen. Dalam era teknologi informasi yang terus berkembang pesat, penggunaan teknologi mobile dan *Artificial Intelligence* (AI) dapat memberikan solusi yang inovatif dalam mendeteksi jamur pada roti secara otomatis dan cepat. Dengan memanfaatkan kemampuan pengolahan citra dan algoritma YOLO, aplikasi deteksi jamur pada roti berbasis mobile dapat memberikan hasil yang akurat dan dapat diandalkan. Tugas akhir ini merincikan pengembangan aplikasi berbasis mobile yang diberi nama "*Mold Detector*" (mold.apk), yang dikembangkan dengan menggunakan Java Android Project. Aplikasi ini memiliki kemampuan untuk mendeteksi keberadaan jamur pada permukaan roti dengan memanfaatkan kamera ponsel. Pengembangan aplikasi ini didasarkan pada pemanfaatan library TensorFlow Lite dan YOLOv5 untuk mengembangkan model deteksi objeknya. Kesimpulan dari rancang bangun ini adalah berhasilnya pengembangan aplikasi pendeteksi jamur berbasis mobile. Aplikasi ini menggunakan algoritma YOLO yang telah diadaptasi ke dalam model TFLite. Hasil uji coba menunjukkan tingkat keberhasilan deteksi jamur rata-rata di atas 50%, menegaskan potensi aplikasi ini untuk membantu pengguna mengidentifikasi jenis-jenis jamur secara lebih efektif dan efisien melalui perangkat mobile.

**Kata Kunci** Jamur, Deteksi Jamur, Mobile, YOLO, Tensorflow



## KATA PENGANTAR

Segala puji dan syukur kami haturkan kepada Allah Subhanahu Wata'ala. Hanya kepada-Nya kami memohon pertolongan. Kami bersyukur atas segala bantuan, kasih sayang, dan rahmat-Nya yang telah memungkinkan penulis menyelesaikan tugas akhir berjudul "**Pengembangan Aplikasi Pendeteksi Jamur pada Roti Berbasis Mobile**". Kami juga senantiasa mengirimkan shalawat dan salam kepada Rasulullah Shallallahu 'Alaihi Wasallam, yang menjadi sumber inspirasi dan panutan bagi seluruh umat manusia.

Tugas akhir ini disusun sebagai bagian dari persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry. Selain itu, tugas akhir ini juga merupakan hasil dari penerapan ilmu yang telah dipelajari selama kuliah di Program Studi Teknologi Informasi.

Penulis sadar bahwa tugas akhir ini masih memiliki kekurangan. Oleh karena itu, penulis berharap dapat terus belajar dan mengembangkan pemahaman kami dalam menerapkan ilmu yang telah kami peroleh. Penyusunan tugas akhir ini tidak terlepas dari bimbingan, masukan, dan arahan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Mama dan Papa yang selalu memberikan doa, dukungan, dan motivasi dalam menyelesaikan tugas akhir ini.
2. Bapak Dr. Ir. Muhammad Dirhamsyah, M.T., IPU. Selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh.
3. Ibu Ima Dwitawati, M.B.A. selaku Ketua Prodi Teknologi Informasi Fakultas Sains dan Teknolgi Universitas Islam Negeri Ar-Raniry Banda Aceh.
4. Bapak Khairan AR, M.Kom. selaku pembimbing I dan Bapak Rahmat Musfikar, M.Kom. selaku pembimbing II yang telah memberikan arahan dan bimbingandalam penyusunan tugas akhir ini.

5. Ibu Cut Ida Rahmadiana, S.Si selaku Staff Prodi Teknologi Informasi yang senantiasa membantu penulis dalam pemberkasan administrasi.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah membekali dengan ilmu pengetahuan di bidang Teknologi Informasi.
7. Ucapan terima kasih juga kepada sahabat-sahabat tercinta angkatan 2019.
8. Terimakasih juga penulis haturkan untuk semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat penulis sebutkansatu persatu.

Akhir kata, penulis sadar bahwa kesempurnaan tidak ada, dan penulis memohon maaf atas kesalahan yang mungkin penulis buat dalam penyusunan tugas akhir ini. Penulis berharap tugas akhir ini dapat bermanfaat bagi pembaca dan menjadi referensi yang berguna untuk pengembangan yang lebih baik. Kebenaran berasal dari Allah dan kesalahan berasal dari diri penulis. Semoga Allah SWT senantiasa melimpahkan rahmat dan ridho-Nya kepada kita semua.

Banda Aceh, 10 Juli 2022  
Penulis,

Muhammad Ghazi Siddiq



## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>DAFTAR SINGKATAN.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
I.1. Latar belakang.....	1
I.2. Rumusan Masalah .....	2
I.3. Tujuan .....	2
I.4. Manfaat .....	2
I.5. Batasan Masalah.....	3
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>4</b>
II.1. Pengenalan Jamur Pada Roti .....	4
II.2. Metode Deteksi Jamur Pada Roti .....	4
II.3. Teknologi Pengolahan Citra Untuk Deteksi Jamur Pada Roti ..	5
II.4. YOLO .....	5
II.5. Aplikasi Mobile .....	8
II.6. Tensorflow.....	8
II.7. Dataset .....	8
II.8. Metode Agile .....	9
II.9. Penelitian Terdahulu.....	11
<b>BAB III METODE PERANCANGAN PROYEK.....</b>	<b>14</b>
III.1. Pendekatan Metode .....	14
III.2. Alur Kerja.....	14
III.2.1. Analisis.....	15
III.2.1.1. Analisis Permasalahan.....	15
III.2.1.2. Deskripsi Umum Sistem.....	16



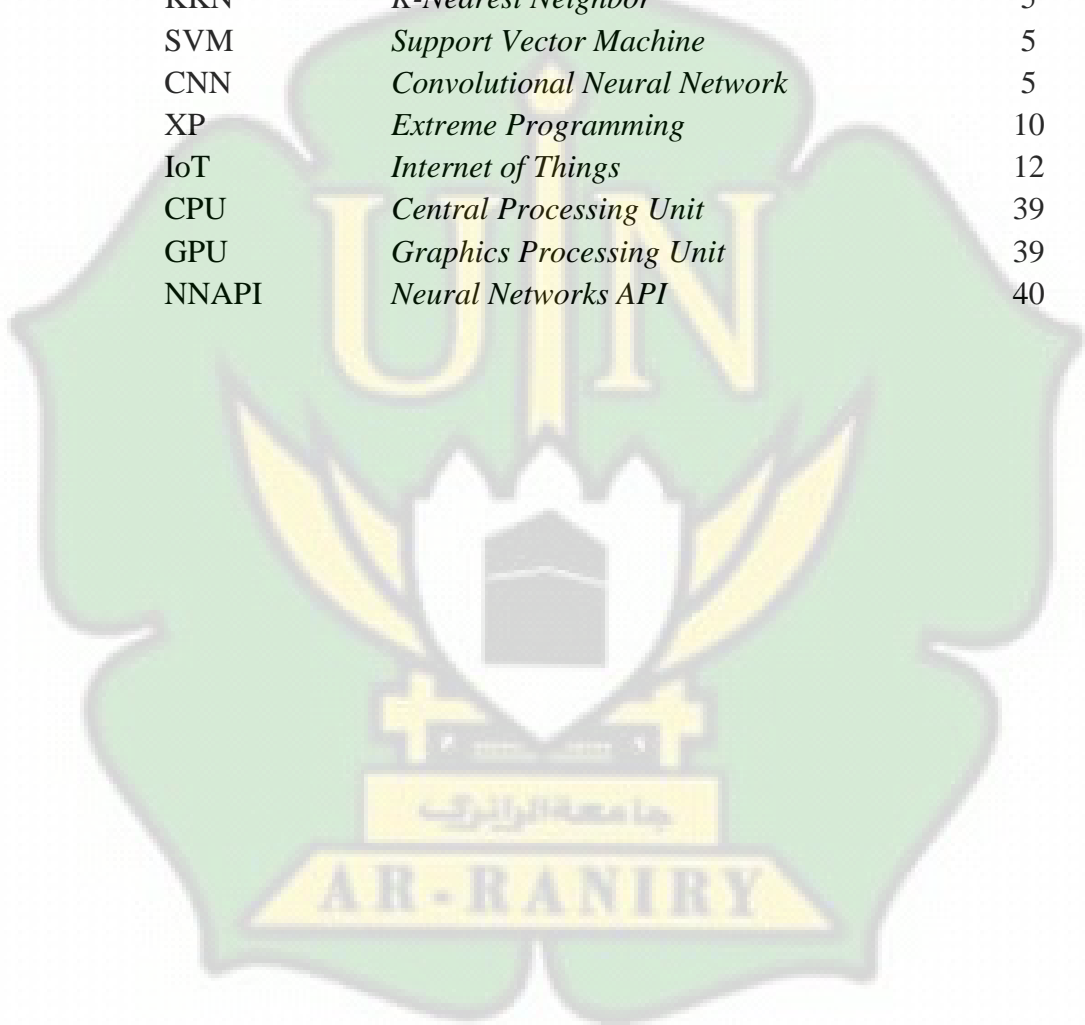
III.2.1.3.	Spesifikasi Kebutuhan Perangkat Lunak.....	17
III.2.1.3.1.	kebutuhan fungsional.....	18
III.2.1.3.2.	Kebutuhan Non-Fungsional.....	18
III.2.2.	Desain.....	18
III.2.2.1.	Desain Sistem .....	18
III.2.2.2.	Desain antarmuka pengguna.....	23
III.2.2.2.1.	Desain Antarmuka Menu Utama .....	23
III.2.2.2.2.	Desain Antarmuka Camera.....	24
III.2.2.3.	Desain Class Diagram .....	26
III.2.3.	Implementasi.....	26
III.2.3.1.	Spesifikasi Perangkat Keras Dan Perangkat Lunak .....	27
III.2.3.2.	Alur Penerapan Desain System .....	28
III.2.4.	Uji Coba .....	30
III.2.5.	Pengembangan dan Perbaikan .....	31
<b>BAB IV</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>32</b>
IV.1.	Hasil.....	32
IV.2.	Pembahasan .....	36
IV.3.	Spesifikasi .....	37
IV.3.1.	Menjalankan Melalui Android .....	37
IV.4.	Pengujian deteksi jamur pada roti .....	38
IV.5.	Kelebihan dan Kekurangan .....	42
IV.5.1.	Kelebihan .....	42
IV.5.2.	Kekurangan .....	42
<b>BAB V</b>	<b>PENUTUP.....</b>	<b>44</b>
V.1.	Kesimpulan .....	44
V.2.	Saran.....	44
<b>DAFTAR PUSTAKA .....</b>		<b>46</b>
<b>LAMPIRAN.....</b>		<b>48</b>

## DAFTAR GAMBAR

Gambar II.1 Proses kerja YOLO.....	7
Gambar III.1 Alur Kerja.....	14
Gambar III.2 Deskripsi umum sistem .....	16
Gambar III.3 Use Case Diagram Desain Sistem .....	19
Gambar III.4 Activity Diagram Threads .....	21
Gambar III.5 Activity Diagram Devices .....	21
Gambar III.6 Activity Diagram Identifikasi Jamur Pada Roti .....	22
Gambar III.7 Sequence Diagram Threads .....	23
Gambar III.8 Sequence Diagram Devices .....	23
Gambar III.9 Sequence Diagram Identifikasi Jamur .....	24
Gambar III.10 Desain Antarmuka Menu Utama.....	25
Gambar III.11 Desain Antarmuka Camera .....	26
Gambar III.12 Desain Class Diagram .....	27
Gambar III.13 Alur Penerapan Desain Sistem .....	29
Gambar III.14 Hasil Uji Coba.....	32
Gambar IV.1 Ikon Aplikasi.....	33
Gambar IV.2 Splashscreen.....	34
Gambar IV.3 Menu .....	34
Gambar IV.4 Halaman Camera.....	35
Gambar IV.5 Halaman Bottom Sheet .....	35
Gambar IV.6 Hasil Deteksi .....	36

## DAFTAR SINGKATAN

SINGKATAN	Nama	Halaman
AI	<i>Artificial Intelligence</i>	1
YOLO	<i>You Only Look Once</i>	1
PCR	<i>polymerase chain reaction</i>	4
LAMP	<i>loop-mediated isothermal amplification</i>	4
KKN	<i>K-Nearest Neighbor</i>	5
SVM	<i>Support Vector Machine</i>	5
CNN	<i>Convolutional Neural Network</i>	5
XP	<i>Extreme Programming</i>	10
IoT	<i>Internet of Things</i>	12
CPU	<i>Central Processing Unit</i>	39
GPU	<i>Graphics Processing Unit</i>	39
NNAPI	<i>Neural Networks API</i>	40



## DAFTAR TABEL

Tabel II.1 Kelebihan dan Kekurangan YOLO .....	6
Tabel III.1 Deskripsi Umum Sistem .....	17
Tabel III.2 Deskripsi Use Case Diagram Desain Sistem.....	19
Tabel III.3 Alur Kejadian Identifikasi Jamur Pada Roti .....	19
Tabel III.4 Alur Kejadian Memilih Threads .....	20
Tabel III.5 Alur Kejadian Memilih Jenis Device .....	20
Tabel III.6 Atribut Desain Antarmuka Menu Utama .....	25
Tabel III.7 Atribut Desain Antarmuka Camera.....	26
Tabel III.8 Spesifikasi Perangkat Keras Dan Perangkat Lunak .....	28
Tabel IV.1 List Keterangan Deteksi.....	33
Tabel IV.2 Komponen Aplikasi .....	33
Tabel IV.3 Spesifikasi Smartphone.....	38
Tabel IV.4 Pengujian Device .....	39
Tabel IV.5 Pengujian Deteksi Jamur Pada Foto (.Jpeg Dan Sejenisnya)	41
Tabel IV.6 Pengujian Deteksi Jamur Pada Objek Nyata (Roti).....	42

## DAFTAR LAMPIRAN

Lampiran 1 Implementasi YOLOv5 Ke tflite .....	48
Lampiran 2 image yang digunakan untuk dataset .....	49
Lampiran 3 kode aplikasi .....	50





# BAB I

## PENDAHULUAN

### I.1. Latar belakang

Produksi roti merupakan salah satu industri yang mengalami pertumbuhan yang cepat dan menjadi salah satu makanan pokok yang digemari oleh banyak orang. Namun, dalam proses produksi roti, terdapat masalah yang sering dihadapi yaitu munculnya jamur. Jamur dapat menurunkan kualitas roti dan menimbulkan kerugian bagi industri roti (Senthilkumar, M., & Arulmozhivarman, P. 2018).

Jamur dapat muncul karena beberapa faktor seperti kondisi lingkungan yang kurang baik, sanitasi yang buruk, kualitas bahan baku yang rendah, dan lain-lain (Siregar, S. A., Simanjuntak, D. D., & Lestari, A. A. 2018). Hal ini menyebabkan jamur menjadi masalah yang serius dalam industri roti. Selain menurunkan kualitas roti, jamur juga dapat menimbulkan risiko kesehatan bagi konsumen.

Untuk mengatasi masalah ini, diperlukan sistem pendeteksi jamur pada roti yang efektif dan efisien. Saat ini, metode pendeteksian jamur pada roti masih dilakukan secara manual seperti LAMP dan PCR yang memerlukan waktu yang lama dan subjektif. Metode ini juga memerlukan biaya yang cukup tinggi dan menyebabkan produksi yang lambat.

Dalam era teknologi informasi yang terus berkembang pesat, penggunaan teknologi mobile dan *Artificial Intelligence* (AI) dapat memberikan solusi yang inovatif dalam mendeteksi jamur pada roti secara otomatis dan cepat. Dengan memanfaatkan kemampuan pengolahan citra dan algoritma YOLO (*You Only Look Once*), aplikasi deteksi jamur pada roti berbasis mobile dapat memberikan hasil yang akurat dan dapat diandalkan.

Dalam konteks ini, penelitian yang terkait dengan deteksi jamur bertujuan mengembangkan sebuah aplikasi yang memungkinkan pengguna dengan mudah mendeteksi keberadaan jamur pada roti melalui perangkat mobile mereka. Aplikasi

ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan efisiensi dan kualitas kontrol kualitas pada roti.

Diharapkan bahwa dengan adanya aplikasi ini, industri roti dapat mengurangi risiko kontaminasi jamur, meningkatkan keamanan pangan, serta meningkatkan kepercayaan konsumen terhadap produk roti yang dihasilkan. Selain itu, penelitian ini juga dapat memberikan kontribusi dalam pengembangan teknologi deteksi jamur yang lebih baik dan dapat diterapkan dalam industri makanan secara luas.

### **I.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah disampaikan sebelumnya, rumusan masalah dari tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara merancang dan membangun aplikasi deteksi jamur pada roti berbasis mobile ?
2. Bagaimana akurasi nilai deteksi dari aplikasi deteksi jamur berbasis mobile ?

### **I.3. Tujuan**

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya, tujuan dari tugas akhir ini adalah sebagai berikut:

1. Untuk Membangun aplikasi deteksi jamur pada roti berbasis mobile.
2. Untuk mengetahui akurasi nilai deteksi dari aplikasi deteksi jamur pada roti berbasis mobile.

### **I.4. Manfaat**

Manfaat dari tugas akhir ini adalah:

1. Memberikan solusi deteksi jamur pada roti yang lebih cepat dan akurat, sehingga dapat membantu produsen roti dalam mengontrol kualitas dan keamanan produk mereka.
2. Meningkatkan kualitas dan keamanan roti yang dihasilkan.  
Menambah pengetahuan dan pemahaman mengenai penggunaan teknologi computer vision dan machine learning dalam bidang industri makanan.

### **I.5. Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah:

1. Aplikasi tersebut hanya memiliki opsi real-time detection dan label “mold”.
2. Metode yang digunakan dalam projek ini adalah YOLO (*You Only Look Once*).



## BAB II

### KAJIAN PUSTAKA

#### II.1. Pengenalan Jamur Pada Roti

Jamur pada roti adalah kondisi di mana roti mengandung pertumbuhan jamur yang dapat menurunkan kualitas roti dan menimbulkan risiko kesehatan bagi konsumen. Jamur dapat muncul karena beberapa faktor seperti kondisi lingkungan yang kurang baik, sanitasi yang buruk, kualitas bahan baku yang rendah, dan lain-lain (Beuchat, L.R., and Pitt, J.I. 1999).

Beberapa jenis jamur yang dapat muncul pada roti antara lain *Saccharomyces cerevisiae*, *Aspergillus flavus*, dan *Penicillium spp.* Jamur ini dapat menyebabkan perubahan warna, bau, rasa, dan tekstur pada roti. Jika dibiarkan, jamur dapat menyebar dan menyebabkan kerugian yang cukup besar bagi industri roti (Pitt, J.I. and Hocking, A.D. 2009). Oleh karena itu, pendeteksian jamur pada roti sangat penting untuk memastikan kualitas dan keamanan produk yang dihasilkan.

#### II.2. Metode Deteksi Jamur Pada Roti

Beberapa metode yang digunakan untuk mendeteksi jamur pada roti antara lain meliputi metode PCR (*polymerase chain reaction*), metode LAMP (*loop-mediated isothermal amplification*), dan teknologi biosensor. Menurut Ana Hidalgo DKK (2018) bahwa metode PCR dapat digunakan untuk deteksi jamur pada roti dengan sensitivitas dan spesifisitas yang tinggi, namun, Metode PCR hanya dapat mendeteksi target spesifik yang telah diidentifikasi sebelumnya dan memerlukan peralatan khusus dan penggunaan bahan kimia yang mahal. Dan menurut Ahmed Al-Hatmi DKK (2018) bahwa metode LAMP dapat digunakan untuk deteksi jamur pada roti dengan sensitivitas dan spesifisitas yang tinggi dalam waktu yang singkat, namun prosedur amplifikasi LAMP memerlukan mesin inkubasi khusus yang mahal, yang dapat menjadi kendala dalam penggunaannya pada skala besar atau di laboratorium yang terbatas. Kemudian menurut Hadi Azimi DKK (2021) bahwa teknologi biosensor dapat digunakan untuk deteksi jamur pada produk roti dengan

sensitivitas dan spesifisitas yang tinggi dalam waktu yang singkat, namun, sama seperti metode sebelumnya biaya yang sangat mahal dan Ketergantungan pada sensor yang hanya dapat mendeteksi target spesifik yang telah diidentifikasi sebelumnya.

Maka dari itu pada penulis akan menggunakan YOLO (*You Only Look Once*) sebagai metode dekteksi jamur pada roti yang lebih efisien, tanpa biaya yang mahal, dan juga waktu yang sangat cepat.

### **II.3. Teknologi Pengolahan Citra Untuk Deteksi Jamur Pada Roti**

Teknologi pengolahan citra merupakan teknologi yang dapat digunakan untuk mendeteksi adanya jamur pada roti dengan cara menganalisis citra roti secara otomatis (Mahanta, A., & Suman, S. K. 2021). Beberapa algoritma yang digunakan dalam teknologi pengolahan citra untuk deteksi jamur pada roti antara lain meliputi algoritma *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), dan *Convolutional Neural Network* (CNN) (Reddy, M. M., & Sanjeevi, S. 2020). Namun, 3 algoritma tersebut dapat memerlukan waktu pelatihan yang lama dan membutuhkan sumber daya komputasi yang tinggi untuk menangani data yang sangat besar dan kompleks. Oleh sebab itu, algoritma yang paling efisien dan akurat dalam mendeteksi jamur pada roti adalah algoritma YOLO (*You Only Look Once*) (Redmon, J., & Farhadi, A. 2018).

### **II.4. YOLO**

YOLO (*You Only Look Once*) adalah algoritma object detection yang digunakan untuk mendeteksi objek dalam sebuah gambar atau video. Algoritma ini dikembangkan oleh Joseph Redmon dan Ali Farhadi pada tahun 2016 dan diperkenalkan dalam paper "*You Only Look Once: Unified, Real-Time Object Detection*" yang diterbitkan pada konferensi computer vision, CVPR (Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016).

YOLO menggunakan sebuah neural network yang dapat mengelompokkan objek dalam sebuah gambar menjadi berbagai kelas dan menentukan koordinat dari objek tersebut dalam satu proses deteksi yang hanya melakukan "*one look*" saja (Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016). Hal ini membuat algoritma YOLO sangat cepat dibandingkan metode object detection lainnya.



berikut ini adalah kelebihan dan kekurangan pada yolo (Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. 2020) :

Tabel II.1 Kelebihan dan Kekurangan

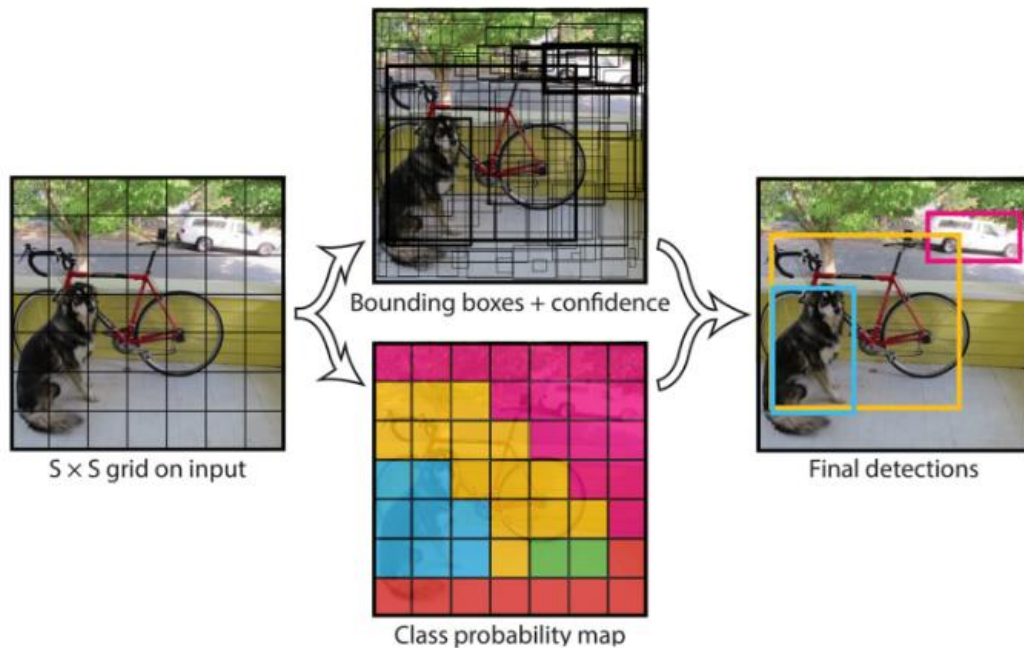
NO	Kelebihan YOLO	Kekurangan YOLO
1	Kecepatan: YOLO memproses gambar secara cepat dengan menghitung probabilitas dan kotak pembatas secara bersamaan. Hal ini membuat YOLO lebih cepat daripada algoritma deteksi objek lainnya.	Tidak mampu mendeteksi objek yang sangat kecil: YOLO kurang efektif dalam mendeteksi objek kecil seperti huruf atau angka kecil pada gambar.
2	Akurasi: YOLO mampu memberikan hasil deteksi objek dengan akurasi yang cukup baik, terutama untuk objek yang relatif besar.	Deteksi objek yang saling tumpang tindih: YOLO cenderung kurang efektif dalam mendeteksi objek yang saling tumpang tindih atau berdekatan
3	Mendukung deteksi objek real-time: Kecepatan dan akurasi YOLO memungkinkan untuk diimplementasikan pada aplikasi real-time.	

YOLO (*You Only Look Once*) adalah salah satu teknik deteksi objek di bidang computer vision yang sangat populer. Berikut adalah rangkaian cara kerja YOLO secara umum:

1. Pertama-tama, gambar masukan akan diubah menjadi grid atau jaringan piksel kecil yang disebut dengan "*grid cells*".
2. Setiap grid cell akan memprediksi beberapa bounding box, atau kotak pembatas yang mengelilingi objek yang ada pada grid tersebut. Tiap bounding box akan memiliki beberapa atribut, seperti koordinat x, y,

lebar, tinggi, dan *confidence score*, yang menunjukkan seberapa yakin model bahwa objek benar-benar ada pada bounding box tersebut.

3. Setelah bounding box diprediksi, model akan menghitung probabilitas untuk setiap kelas objek yang mungkin, seperti mobil, pesawat, atau orang. Probabilitas ini akan dihitung berdasarkan bagaimana fitur-fitur visual pada *bounding box* tersebut cocok dengan fitur-fitur visual dari objek-objek pada data latih.
4. Akhirnya, model akan mengambil *bounding box* dengan *confidence score* tertinggi dan mengasosiasikannya dengan kelas objek dengan probabilitas tertinggi.



Gambar II.1 Proses kerja YOLO

Dalam prosesnya, YOLO menggunakan arsitektur *convolutional neural network* (CNN) yang sangat kompleks dan membutuhkan komputasi yang tinggi. Namun, karena kemampuannya yang sangat akurat dan cepat, YOLO banyak digunakan di berbagai aplikasi deteksi objek pada *real-time*, seperti deteksi kecelakaan lalu lintas dan pengawasan keamanan (Wasril, Abi Rachman, dkk. 2019).

## II.5. Aplikasi Mobile

Aplikasi mobile saat ini telah menjadi bagian penting dalam kehidupan sehari-hari. Aplikasi mobile dapat digunakan untuk berbagai keperluan seperti hiburan, pendidikan, dan bisnis. Penggunaan aplikasi mobile semakin meningkat dengan pesatnya perkembangan teknologi smartphone.

Android adalah sistem operasi mobile yang paling banyak digunakan di seluruh dunia. Android menyediakan platform yang kuat dan fleksibel untuk pengembangan aplikasi mobile. Android juga menyediakan berbagai tools dan framework untuk memudahkan pengembangan aplikasi seperti Android Studio dan Android SDK. Penelitian oleh Akbar et al. (2019) membahas pengembangan aplikasi pemesanan makanan berbasis Android menggunakan framework Android. Hasil penelitian menunjukkan bahwa penggunaan framework Android dapat mempercepat proses pengembangan aplikasi (W. A. M. A. Wijesinghe dan R. N. Munasinghe. 2019).

## II.6. Tensorflow

Tensorflow adalah sebuah *Library Machine Learning* yang bersifat open source, dikembangkan oleh Google pada tahun 2015. Tensorflow menggunakan data grafik (graph) untuk membangun modelnya. Data grafik ini terdiri dari sejumlah node yang mewakili operasi matematika, dan node-node ini terhubung melalui yang disebut sebagai tensor.

Model yang dihasilkan oleh Tensorflow bisa diekspor dalam format Flatbuffer (.tflite) dan dijalankan pada perangkat mobile dengan bantuan Tensorflow Lite. Tensorflow Lite, yang diperkenalkan oleh Google pada tahun 2017, memiliki kinerja yang lebih baik dan ukuran biner yang lebih kecil karena optimisasi kernel, aktivasi yang telah disatukan (*pre-fused activations*), serta jumlah dependensi yang lebih minim (Alsing, Oscar. 2018).

## II.7. Dataset

Dataset adalah kumpulan data yang digunakan dalam proses pembelajaran mesin (*machine learning*) atau analisis data. Dataset terdiri dari beberapa baris yang mewakili sebuah objek atau entitas, dan setiap baris terdiri dari beberapa kolom

yang mewakili fitur atau atribut dari objek tersebut (Provost, F., & Fawcett, T. 2013).

Dataset dapat digunakan dalam berbagai macam proses analisis data, seperti klasifikasi, regresi, dan clustering. Dalam proses pembelajaran mesin, dataset digunakan sebagai input untuk mengevaluasi dan menyempurnakan model (Géron, A. 2019).

Ada dua jenis dataset yang umum digunakan dalam proses pembelajaran mesin, yaitu dataset latih (training dataset) dan dataset uji (testing dataset). Dataset latih digunakan untuk melatih model, sedangkan dataset uji digunakan untuk mengevaluasi kinerja model yang telah dilatih.

Dataset yang digunakan dalam proses pembelajaran mesin harus mencakup representasi yang valid dari domain yang dianalisis dan harus diperoleh dari sumber yang dapat dipercaya. Juga penting untuk mengevaluasi dan membersihkan dataset sebelum digunakan dalam proses pembelajaran mesin untuk menghindari bias dan kesalahan dalam model yang dihasilkan (Kelleher, J. D., Tierney, B., & Tierney, B. 2018).

Maka dari itu dataset yang digunakan adalah dataset yang di buat secara manual oleh penulis berupa kurang lebih 300 gambar jamur pada roti.

## **II.8. Metode Agile**

Metode Agile adalah suatu pendekatan pengembangan perangkat lunak yang fleksibel dan adaptif, yang berfokus pada pengiriman produk yang bernilai kepada pengguna akhir dengan cara yang efisien dan efektif. Metode Agile menekankan pada kerja sama dan komunikasi antara tim pengembang dan pemangku kepentingan, serta memprioritaskan pengiriman produk secara berulang dan bertahap dalam waktu yang singkat (Cohn, M. 2005).

Pendekatan Agile didasarkan pada empat nilai inti yaitu:

1. Individu dan interaksi lebih diutamakan daripada proses dan alat.
2. Perangkat lunak yang berfungsi lebih diutamakan daripada dokumen yang lengkap.

3. Kerja sama dengan pelanggan lebih diutamakan daripada negosiasi kontrak.
4. Respons terhadap perubahan lebih diutamakan daripada mengikuti rencana yang telah ditetapkan.

Metode Agile memiliki beberapa kerangka kerja seperti Scrum, Kanban, dan XP (*Extreme Programming*). Setiap kerangka kerja memiliki aturan, praktik, dan prosesnya sendiri yang disesuaikan dengan kebutuhan pengembangan perangkat lunak yang berbeda. Berikut adalah langkah-langkah umum untuk melakukan metode Agile:

1. Analisis Kebutuhan

Analisis Kebutuhan dalam metode Agile adalah proses yang penting dalam pengembangan aplikasi atau produk. Ini melibatkan pemahaman dan dokumentasi kebutuhan bisnis serta kebutuhan pengguna yang akan diimplementasikan dalam produk yang sedang dikembangkan.

2. Desain Sistem

Desain sistem pada metode Agile adalah proses iteratif yang terjadi secara berulang selama seluruh siklus pengembangan aplikasi. Tujuannya adalah untuk menciptakan struktur dan rincian teknis dari sistem dengan berfokus pada pengiriman nilai yang tinggi kepada pelanggan. Desain sistem dalam metode Agile sering kali terkait dengan setiap sprint pengembangan, dan keputusan desain dapat berubah seiring berjalannya waktu untuk mengakomodasi perubahan kebutuhan yang mungkin terjadi.

3. Implementasi

Implementasi dalam metode Agile merujuk pada tahap di mana fitur atau fungsionalitas yang telah dirancang dan dikembangkan akan diintegrasikan ke dalam produk atau aplikasi yang sedang dibuat. Pendekatan Agile menekankan pada pengiriman hasil kerja secara berkala dan inkremental, sehingga implementasi berfokus pada menggabungkan fitur-fitur ini dalam iterasi berulang yang disebut "sprint".



#### 4. Uji Coba

Uji coba pada metode Agile adalah tahap penting dalam pengembangan aplikasi di mana fitur-fitur yang telah diimplementasikan diuji untuk memastikan bahwa mereka berfungsi dengan benar dan sesuai dengan kebutuhan pengguna. Uji coba dalam metode Agile dilakukan secara berulang setiap kali fitur-fitur baru ditambahkan ke dalam produk selama siklus pengembangan yang disebut "sprint". Tujuan utama dari uji coba adalah untuk meningkatkan kualitas produk, mendeteksi masalah sejak dini, dan memberikan umpan balik cepat kepada tim pengembang.

#### 5. Pengembangan dan Perbaikan

Pengembangan dan Perbaikan dalam metode Agile adalah proses berkelanjutan untuk menciptakan, meningkatkan, dan mengoptimalkan produk atau aplikasi seiring berjalannya waktu. Pendekatan ini menerapkan siklus pengembangan iteratif dan inkremental, di mana produk terus dikembangkan dalam iterasi pendek yang disebut "sprint". Tujuan utama dari Pengembangan dan Perbaikan dalam metode Agile adalah untuk memberikan nilai yang tinggi kepada pengguna dan pemangku kepentingan secara cepat dan adaptif.

Metode Agile memungkinkan fleksibilitas dalam pengembangan aplikasi, memungkinkan tim untuk beradaptasi dengan perubahan kebutuhan dan menghasilkan produk berkualitas tinggi yang lebih sesuai dengan kebutuhan pengguna. Penting untuk mengadopsi prinsip-prinsip dan praktik Agile dengan baik serta mempertahankan komunikasi yang terbuka antara tim dan pemangku kepentingan selama seluruh proses pengembangan aplikasi.

### **II.9. Penelitian Terdahulu**

Penelitian terdahulu mengenai aplikasi pendeteksi jamur pada roti berbasis mobile telah menunjukkan bahwa aplikasi ini dapat digunakan untuk mendeteksi dan mengklasifikasikan jamur pada roti dengan tingkat akurasi yang cukup tinggi.

Pada penelitian yang dilakukan oleh Pangestu, A. S., & Tanamal, R. pada tahun 2020 pada jurnal yang berjudul "Rancang Bangun Aplikasi Sistem Pakar Berbasis

Mobile Untuk Mendiagnosis Penyakit Kulit Pada Kucing Persia", menunjukkan bahwa aplikasi sistem pakar berbasis mobile yang dapat mendiagnosis penyakit kulit pada kucing Persia. Aplikasi ini dibuat menggunakan Thunkable tanpa coding, dengan menggunakan Decision Tree dan metode inferensi forward chaining. Data diperoleh melalui wawancara dengan pakar kesehatan hewan. Hasil penelitian menunjukkan bahwa aplikasi Meow'Diagnosis berguna dan berjalan dengan baik untuk mendiagnosis penyakit kulit kucing Persia.

Pada penelitian yang dilakukan oleh Rizal Effendi pada tahun 2020 pada jurnal yang berjudul "Rancang Bangun Aplikasi Augmented Reality Untuk Deteksi Pengenalan Tanaman Obat Berbasis Android", menunjukkan bahwa aplikasi ini menggunakan Unity untuk memproses semua keperluan AR dan menu antarmuka. Hasil implementasi dan pengujian menunjukkan aplikasi telah sesuai dengan perancangan yang direncanakan, dengan fitur-fiturnya berfungsi dengan baik. Jurnal ini membahas cara aplikasi menampilkan informasi tanaman berupa suara dan gambar pada kamera.

Sementara itu pada penelitian lain yang dilakukan Fadly Aditya Putra, D. dalam Jurnal yang berjudul "Rancang Bangun Aplikasi Mobile Pendeteksi Potensi Kanker" pada tahun 2022, menunjukkan bahwa hasil penelitian ini menghasilkan aplikasi mobile yang dapat memprediksi kanker payudara pada stadium awal. Aplikasi tersebut melakukan pemindaian atau input mammografi dengan fitur image processing dan sistem pakar berdasarkan jawaban pengguna terkait gejala. Pengujian menunjukkan akurasi, presisi, dan recall sistem pakar mencapai 100%. Semua fitur pada aplikasi berjalan dengan baik berdasarkan uji alpha.

Kemudian penelitian yang dilakukan oleh Puput Laksmiwati pada tahun 2023 pada jurnal yang berjudul "Rancang Bangun Aplikasi Mobile untuk Monitoring Kesehatan Berbasis *Internet of Things* (IoT)", menunjukkan bahwa aplikasi ini menggunakan teknologi IoT untuk mengumpulkan data kesehatan yang akurat dari perangkat kesehatan terhubung, seperti sensor suhu tubuh dan detektor detak jantung. Aplikasi ini juga menyediakan fitur untuk menyimpan riwayat kesehatan, membuat grafik perkembangan, dan mengatur pengingat. Keamanan data dan privasi pengguna menjadi perhatian utama, dengan enkripsi data dan implementasi

protokol keamanan yang ketat. Pengguna akan mendapatkan notifikasi jika ada kondisi kesehatan yang tidak normal.



## BAB III

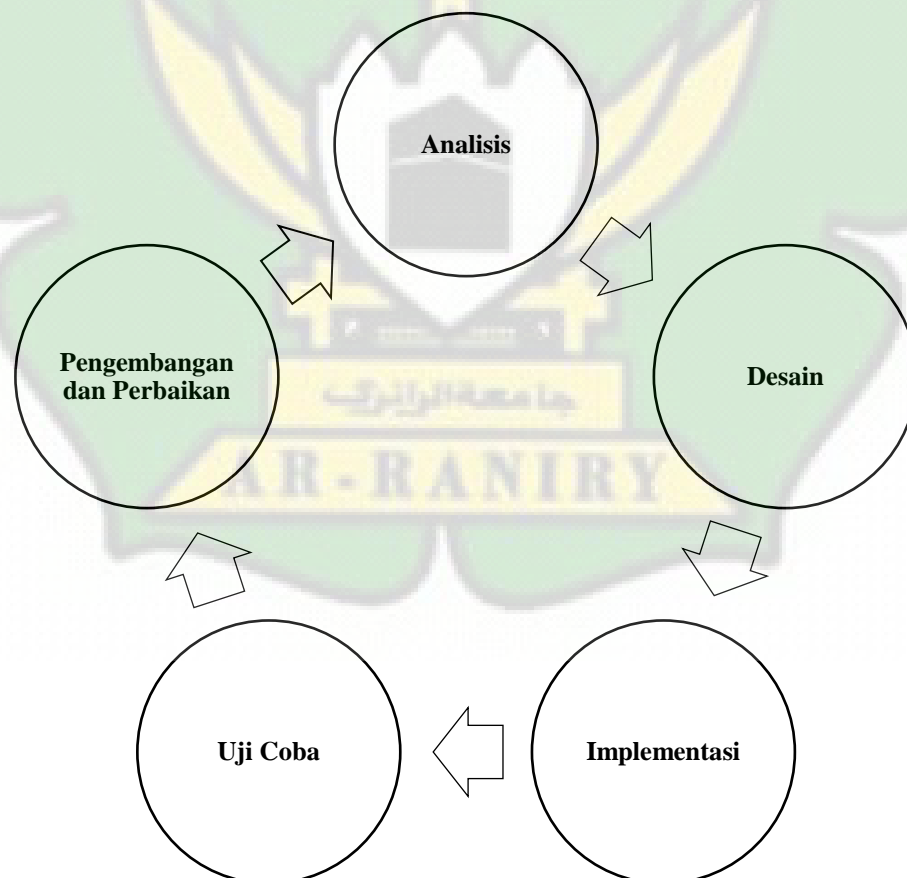
### METODE PERANCANGAN PROYEK

#### III.1. Pendekatan Metode

Untuk memastikan bahwa proyek ini dapat terlaksana sesuai dengan tujuan dan harapan, maka dalam penelitian ini akan menggunakan pendekatan metode Agile. Agile merupakan pendekatan iteratif dan inkremental untuk mengelola proyek software yang menekankan pada fleksibilitas dan respons terhadap perubahan.

#### III.2. Alur Kerja

Pada bab ini menjelaskan alur pembuatan aplikasi bisa dilihat pada gambar III.1 yang dimulai dari analisis sampai dengan pengembang dan perbaikan.



Gambar III.1 Alur Kerja

### **III.2.1. Analisis**

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

#### **III.2.1.1. Analisis Permasalahan**

Pada roti tersebut, sistem ini juga berperan sebagai bukti apakah roti tersebut aman untuk dikonsumsi atau tidak. Di beberapa tempat penjualan, terkadang tidak diketahui apakah roti yang telah dijual selama beberapa minggu sudah diketahui oleh pembeli, yang akhirnya dapat mengakibatkan pembeli mengalami masalah kesehatan seperti sakit perut. Oleh karena itu, salah satu metode untuk menjamin kesehatan konsumen adalah dengan menggunakan sistem deteksi jamur pada roti. Sistem ini dapat mengidentifikasi apakah roti masih dalam kondisi yang baik atau telah terkontaminasi oleh jamur. Dengan cara ini, sistem ini dapat membantu penjual roti untuk memastikan bahwa produk yang mereka tawarkan tidak mengandung jamur yang berpotensi membahayakan kesehatan. Selain itu, sistem ini juga dapat memberikan informasi kepada pembeli mengenai kondisi roti yang mereka beli, sehingga mereka dapat menghindari risiko penyakit seperti sakit perut yang mungkin disebabkan oleh mengonsumsi roti yang sudah berjamur.

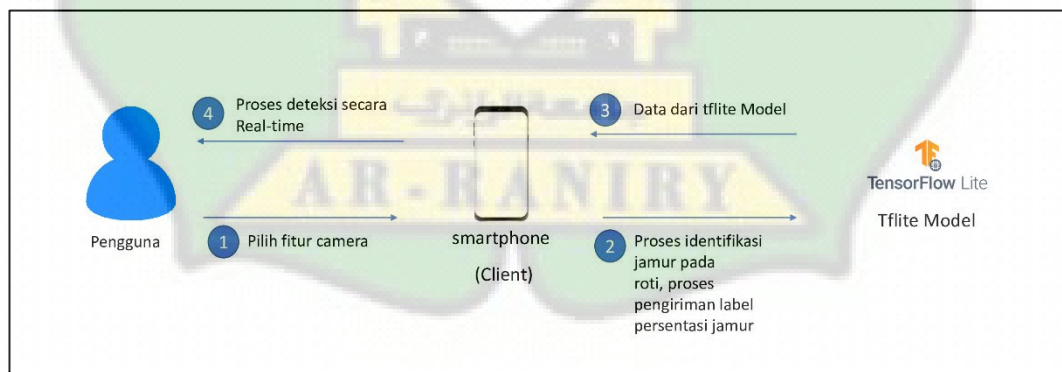
Perkembangan teknologi informasi di dunia semakin cepat, khususnya pada teknologi perangkat bergerak (mobile). Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak maupun web. Perkembangan ini juga mempengaruhi cara seseorang dalam mengetahui roti itu basi atau tidaknya kedalam sebuah sistem. Untuk mengetahui roti itu berjamur pada saat ini mungkin hanya digunakan di supermarket yang mana mengetahuinya pada saat scan detail expirednya, dan pada permukaan rotinya yang area nya sudah menghitam tetapi terkadang juga itu bukan lah jamur melainkan selai atau meses. Oleh karena itu, tugas akhir ini akan mengimplementasikan sebuah aplikasi berbasis mobile Implementasi pada aplikasi mobile diharapkan mampu menambah fleksibilitas dari sistem deteksi yang sudah ada saat ini. Aplikasi deteksi yang akan dibuat tidak dibatasi

ruang lingkup penggunaannya. Pengguna dari aplikasi ini adalah siapa pun yang ingin membeli atau menyantap roti.

Sistem deteksi pada aplikasi ini mengimplementasikan metode YOLO ke dalam aplikasi mobile. Aplikasi ini diimplementasikan algoritma-algoritma dari pustaka (library) Tensorflowlite dan pustaka (library) YOLOV5. Prosedur pada aplikasi ini adalah sebagai berikut: pertama, aplikasi ini akan mengambil gambar dari kamera. Kedua, aplikasi ini akan mengirim gambar ke pustaka Tensorflowlite untuk melakukan inferensi. Ketiga, aplikasi ini akan menerima hasil inferensi berupa koordinat dan label dari objek yang terdeteksi. Keempat, aplikasi ini akan menampilkan hasil deteksi pada layar dengan menambahkan kotak dan teks pada gambar.

Proses pemberian teks dan kotak ini dilakukan dengan melatih aplikasi sebelumnya. Tahap pelatihan meliputi pengenalan jamur yang kemudian hasil pengenalan ini diproses dengan menggunakan beberapa teknik pemrosesan citra agar dapat diterima dengan baik oleh aplikasi, kemudian mengajarkan sistem mengenai identitas jamur yang terdeteksi dengan menggunakan algoritma pelatihan.

### III.2.1.2. Deskripsi Umum Sistem



Gambar III.2 Deskripsi umum sistem

Gambar III.2 menggambarkan struktur perangkat lunak yang akan dikonstruksi. Gambar ini menunjukkan bahwa perhatian utama difokuskan pada perangkat lunak yang sedang dikembangkan. Sistem yang sedang dibangun mencakup tahap deteksi jamur yang dilakukan menggunakan



kamera pada perangkat client/smartphone, proses identifikasi jamur yang berlangsung di sisi client/smartphone, dan pengiriman label persentase hasil identifikasi ke model tflite. Dalam gambar ini, tahapan input, proses, dan output dalam konteks penyelesaian tugas akhir ini ditunjukkan dengan jelas. Rangkaian tahapnya adalah sebagai berikut.

1. Tahap input aplikasi mobile sistem deteksi ditunjukkan pada nomor 1. Input aplikasi yang diberikan adalah pemilihan fitur kamera dengan pencocokan jamur pada roti, pemilihan fitur threads, dan pemilihan fitur devices.
2. Tahap proses ditunjukkan pada nomor 2, 3, dan 4. Pada tahap ini, aplikasi mobile sistem deteksi jamur melakukan proses identifikasi jamur dari client ke tflite model, proses training data set jamur pada roti, proses deteksi, serta proses pengiriman label persentasi berupa teks dan border box.
3. Tahap output aplikasi mobile system deteksi jamur ditunjukkan pada nomor 2. Output dari aplikasi mobile system deteksi jamur ini adalah pengiriman label persentasi jamur pada tflite model.

### **III.2.1.3. Spesifikasi Kebutuhan Perangkat Lunak**

Sejalan dengan lingkup perangkat lunak yang telah diuraikan dalam bagian deskripsi sistem secara umum, diperlukan pembuatan spesifikasi perangkat lunak agar dapat menghadirkan solusi bagi permasalahan yang diberikan dan beroperasi dengan optimal dalam memenuhi kebutuhan yang ada. Dengan adanya spesifikasi ini, diharapkan dapat memastikan kesesuaian dengan kebutuhan para pengguna. Spesifikasi kebutuhan perangkat lunak mengacu pada penjelasan tertulis mengenai kebutuhan sistem yang diinginkan oleh pengguna atau klien. Dalam kerangka tugas akhir ini, spesifikasi kebutuhan perangkat lunak terbagi menjadi dua kategori, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

### III.2.1.3.1. kebutuhan fungsional

Tabel III.1 Deskripsi Umum

No	Kebutuhan Fungsional	Deskripsi
1	Mendeteksi pada objek jamur pada roti	Melakukan deteksi pada objek jamur pada roti
2	Menthreads pada objek jamur pada roti	Melakukan pemilihan threads 1-8
3	Mendevices pada objek jamur pada roti	Melakukan pemilihan devices (GPU,CPU,dan NNAPI)

### III.2.1.3.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem adalah sebagai berikut.

#### 1. Kebutuhan Performa

Perangkat lunak akan berjalan dengan performa terbaik apabila dijalankan di atas spesifikasi perangkat minimal.

### III.2.2. Desain

Desain dalam subbab ini membahas perancangan dari aplikasi tugas akhir bernama mold detector. Subbab ini terdiri dari desain sistem, desain antarmuka pengguna, dan desain class diagram.

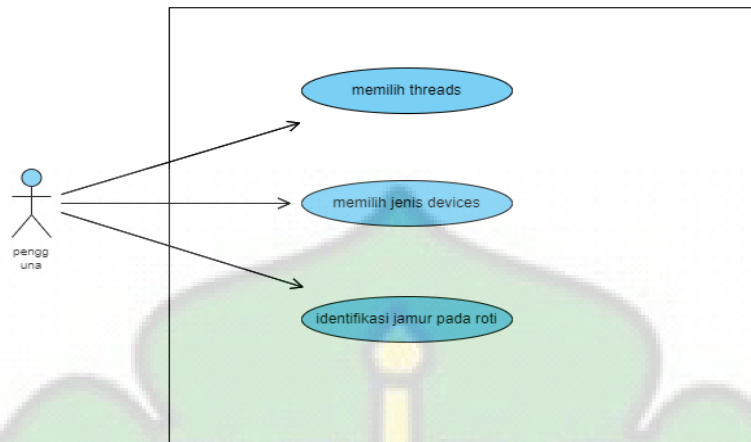
#### III.2.2.1. Desain Sistem

Desain system untuk tugas akhir ini terdiri dari menggunakan notasi UML diantaranya use case diagram, activity diagram, class diagram, dan sequence diagram.

#### 1. Use case diagram

Berdasarkan kebutuhan fungsional, system membutuhkan 3 kasus penggunaan . use case diagram bisa dilihat pada gambar III.3 beserta

spesifikasinya pada tabel III.2 dan alur kejadiannya pada table III.3 s/d III.6.



Gambar III.3 Use Case Diagram Desain Sistem

Tabel III.2 Deskripsi Use Case Diagram Desain Sistem

No	Nama use case	Deskripsi
1	Pengguna	Orang yang berinteraksi dengan sistem
2	Memilih threads	Pengguna menggunakan pemilihan threads 1 - 8 untuk mempercepat kinerja system deteksi.
3	Memilih jenis device	Pengguna menggunakan device (CPU, GPU ,dan NNAPI) untuk mempercepat inferensi.
4	Identifikasi jamur pada roti	Melakukan pendeteksian objek (jamur) pada permukaan roti secara real time

Tabel III.3 Alur Kejadian Identifikasi Jamur Pada Roti

Actor	Sistem
1. memilih pilihan “camera”	
	2. menampilkan halaman “kamera”
3. mengarahkan smarphone pada objek	

	4. menampilkan hasil deteksi objek berupa border box dan label percentase
--	---

Tabel III.4 Alur Kejadian Memilih Threads

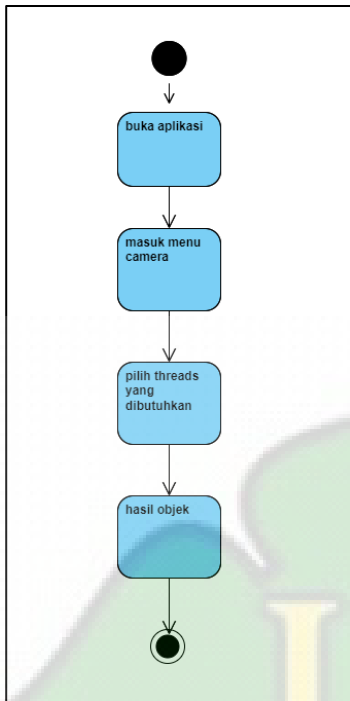
Actor	Sistem
1. memilih pilihan “bottom sheet button” pada bagian bawah halaman kamera	
	2. menampilkan “menampilkan tampilan tambahan yang muncul dari bagian bawah layer (threads)”
3. memilih threads 1-8	
	4. meningkatkan waktu eksekusi pada inferensi model TFLite

Tabel III.5 Alur Kejadian Memilih Jenis Device

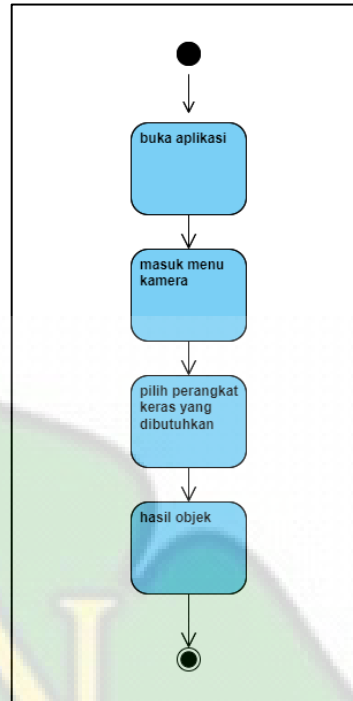
Actor	Sistem
1. memilih pilihan “bottom sheet button” pada bagian bawah halaman kamera	
	2. menampilkan “menampilkan tampilan tambahan yang muncul dari bagian bawah layer (devices)”
3. memilih devices	
	4. meningkatkan keefisiensi pada eksekusi inferansi model TFLite

## 2. Activity diagram

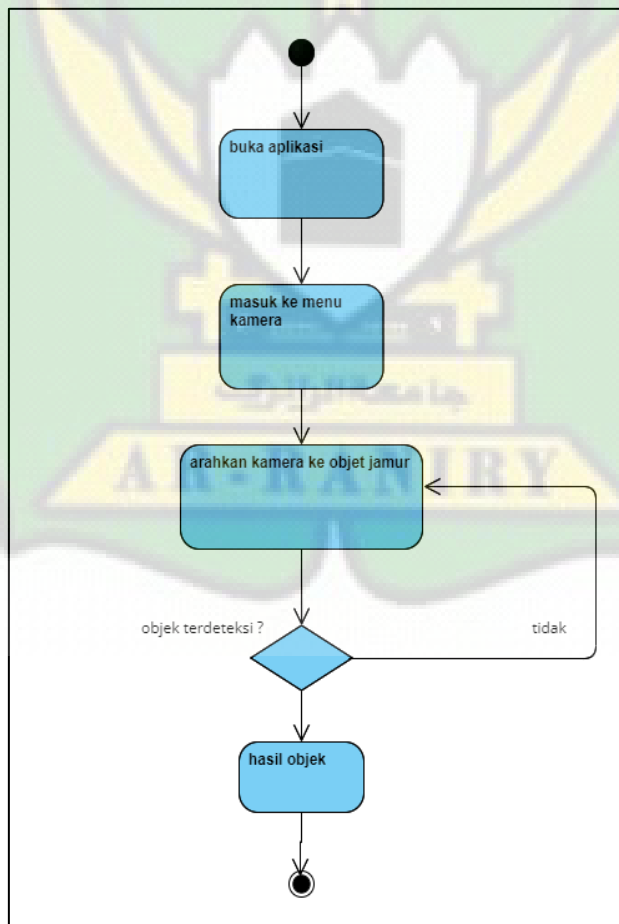
Berdasarkan use case diagram diatas terdiri dari 3 aktivitas yang dilakukan system maka terbentuk 3 activity diagram yaitu memilih threads (gambar 3.4), memilih device (gambar 3.5), dan identifikasi jamur pada roti (gambar 3.6).



Gambar III.4 Activity Diagram Threads



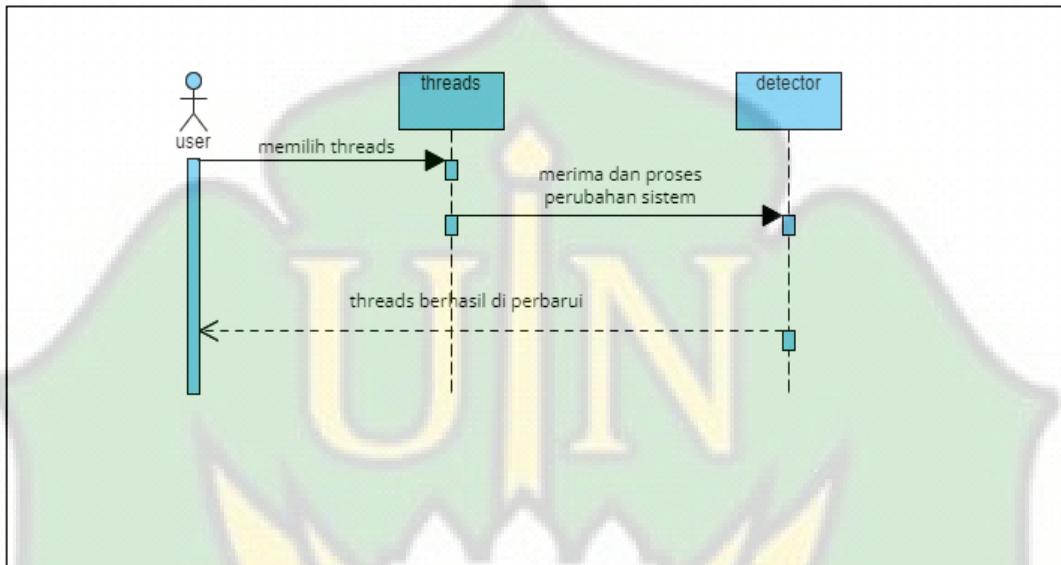
Gambar III.5 Activity Diagram Devices



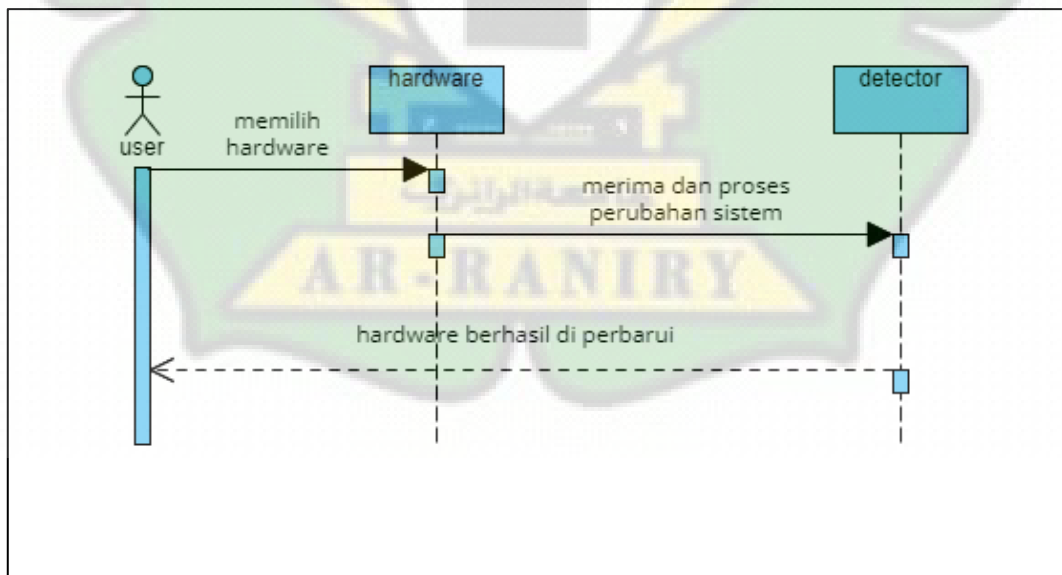
Gambar III.6 Activity Diagram Identifikasi Jamur Pada

### 3. Sequence diagram

Berdasarkan use case diagram diatas terdiri dari 3 aktivitas yang dilakukan system maka terbentuk 3 sequence diagram yaitu memilih threads (gambar III.7), memilih device (gambar III.8), dan identifikasi jamur pada roti (gambar III.9).

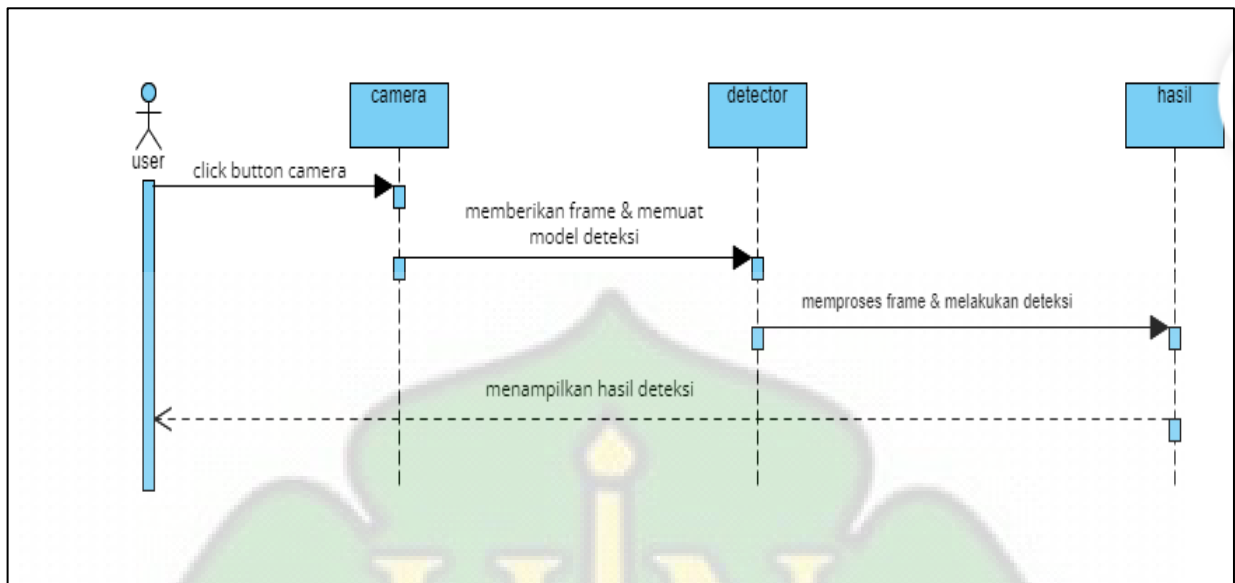


Gambar III.7 Sequence Diagram Threads



Gambar III.8 Sequence Diagram Devices





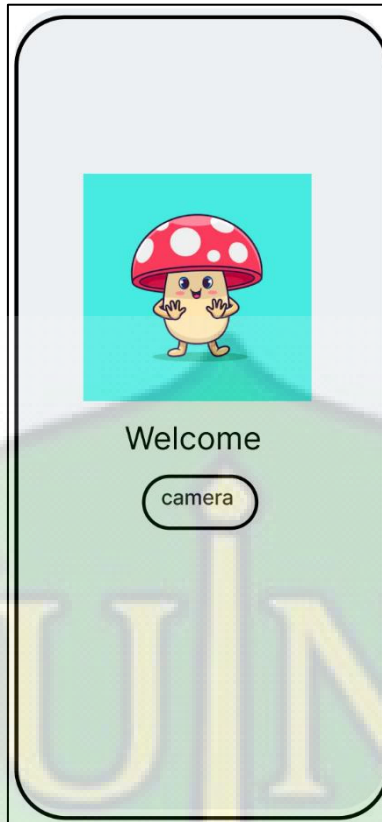
Gambar III.9 Sequence Diagram Identikasi Jamur

### III.2.2.2. Desain antarmuka pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan dan tampilan yang menarik bagi penggunanya. Sesuai dengan gambaran umum perangkat lunak yang dikerjakan dalam aplikasi mold detector, antarmuka pengguna hanya dirancang untuk satu hak akses pengguna, yaitu semua orang yang menggunakan aplikasi ini. terdapat halaman menu utama, dan halaman camera.

#### III.2.2.2.1. Desain Antarmuka Menu Utama

Halaman ini akan ditampilkan ketika baru memasuki aplikasi. Desain dapat dilihat pada gambar III.10. dan atribut antarmuka dapat dilihat pada tabel III.6.



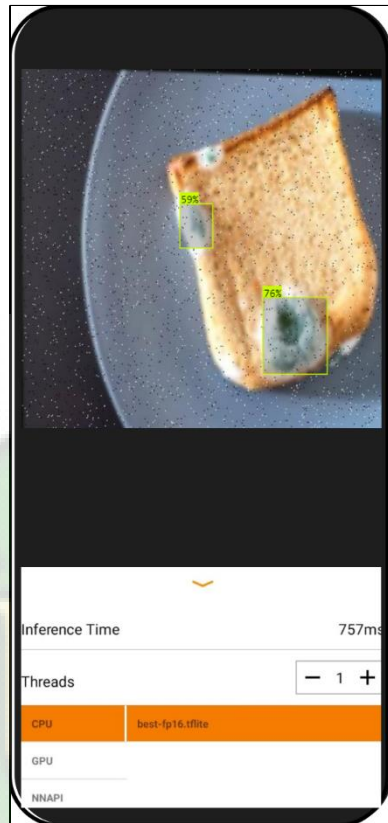
Gambar III.10 Desain Antarmuka Menu Utama

Tabel III.6 Atribut Desain Antarmuka Menu Utama

No	Nama atribut	Jenis atribut	fungsi	Masukan/keluaran
1	Ikon aplikasi	Imageview	Tampilan gambar ikon aplikasi	Image (.jpg/.png)
2	Main menu	textview	Tampilan untuk text	String
3	Button_camera	button	Tombol untuk menuju halaman camera	Buttonclick

#### III.2.2.2.2. Desain Antarmuka Camera

Halaman ini ditampilkan pada aplikasi ketika pengguna memilih pilihan camera pada tampilan antarmuka menu utama. Rancangan halaman dapat dilihat pada Gambar III.11. Atribut antarmuka dapat dilihat pada Tabel III.7.



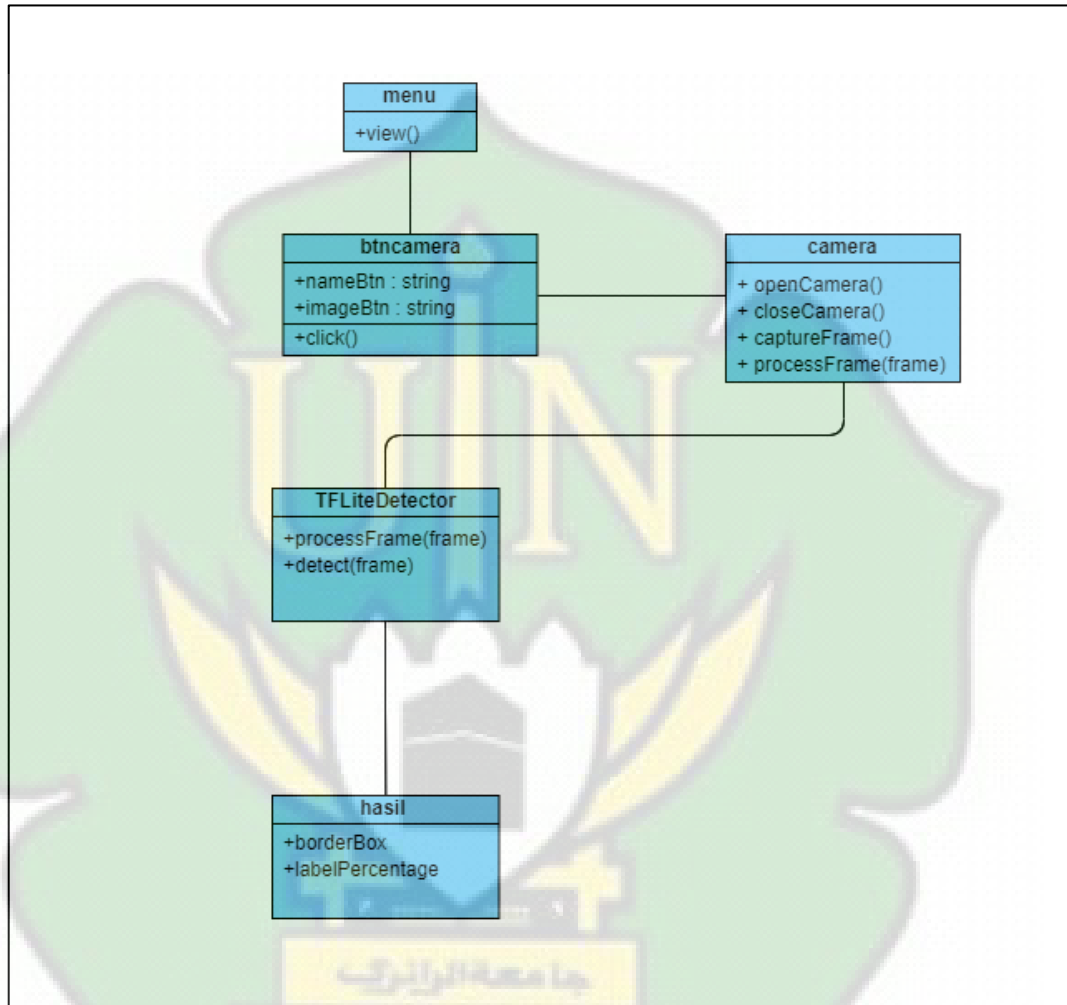
Gambar III.11 Desain Antarmuka Camera

Tabel 3.7 Atribut Desain Antarmuka Camera

No	Nama atribut	Jenis atribut	fungsi	Masukan/keluaran
1	kamera	CustomCameraView	Informasi persentase jamur pada roti	String
2	Interface time	textview	Informasi waktu interface	String
3	threads	button	Tombol untuk memilih threads 1-9	ButtonClick
4	CPU	button	Tombol untuk memilih devices CPU	ButtonClick
5	GPU	button	Tombol untuk memilih devices GPU	ButtonClick
6	NNAPI	button	Tombol untuk memilih devices nn	ButtonClick

### III.2.2.3. Desain Class Diagram

Perancangan class diagram dapat dilihat pada Gambar 3.12. Pada class diagram ini akan ditunjukkan kelas-kelas beserta beberapa atribut dan fungsi penting yang akan digunakan dalam aplikasi mold detector.



Gambar III.12 Desain Class Diagram

### III.2.3. Implementasi

Dalam tahap ini, penulis menerapkan desain yang terstruktur dan mengimplementasikan algoritma YOLO (You Only Look Once) untuk melakukan pendeteksian jamur pada permukaan roti. Algoritma YOLO telah terbukti efektif dalam mendeteksi objek dengan akurasi tinggi dan waktu inferensi yang cepat. Penerapan algoritma ini akan melibatkan tahap prapemrosesan, di mana gambar permukaan roti akan diubah ukurannya dan dinormalisasi agar sesuai dengan persyaratan algoritma. Selanjutnya, model

YOLO yang telah dilatih akan diintegrasikan ke dalam sistem dan dijalankan pada gambar roti yang telah diproses. Hasil dari pendeteksian jamur akan dihasilkan dalam bentuk kotak pembatas di sekitar area yang mengandung jamur. Proses ini akan diperinci dan diuji untuk memastikan akurasi dan efisiensi deteksi yang optimal pada aplikasi pendeteksian jamur pada roti kami. Pada bagian ini akan ditampilkan spesifikasi perangkat keras dan perangkat lunak, dan juga alur penerapan desain system.

### III.2.3.1. Spesifikasi Perangkat Keras Dan Perangkat Lunak

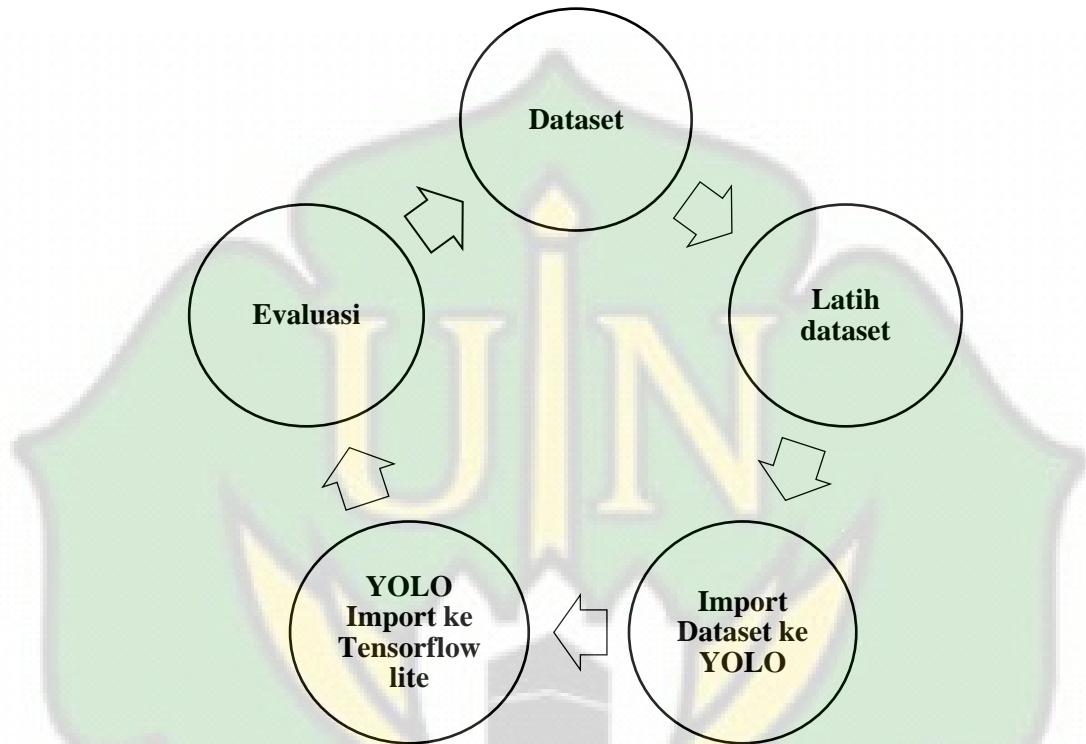
Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak ini dijelaskan pada Tabel III.8.

Tabel III.8 Spesifikasi Perangkat Keras Dan Perangkat Lunak

Perangkat keras	komputer	Lenovo thinkpad x270
	prosesor	Intel(R) Core (TM) i5-6300U CPU @ 2.40GHz (4 CPUs), ~2.5GHz
	Memori primer	8 GB
	Memori sekunder	237 GB
Perangkat lunak	Sistem operasi	Windows 10 Pro 64-bit
	Perangkat lunak	Android studio, visual paradigm, google collabs, roboflow, tensorflow lite, Microsoft word 2019

### III.2.3.2. Alur Penerapan Desain System

Proses implementasi perangkat lunak memaparkan langkah-langkah penerapan desain sistem yang tergambar dengan rinci pada Gambar III.13.



Gambar 3.13 Alur Penerapan Desain Sistem

#### 1. Dataset

Pertama download dataset yang berlabel “breadmold” yang berisi tentang jamur (berupa foto dan labelnya) yang bisa di download di link dibawah ini:

<https://universe.roboflow.com/muhammad-ghazi-siddiq-a6hso/bread-mold-yb186/dataset/10>

#### 2. Latih dataset

Setelah melakukan download dataset, maka data set tersebut harus dilatih agar bisa menjadi sebuah model untuk deteksi jamur yang bisa dilakukan di roboflow atau google collabs.



### 3. Import Dataset ke YOLO

Setelah dataset tadi dilatih, maka dilakukan import pada dataset tersebut ke direktori YOLO dan kemudian dilatih lagi dengan perintah:

```
“!python train.py --img 416 --batch 10 --epochs 250 --data data.yaml --weights yolov5s.pt --cache”
```

Keterangan :

1. python train.py: Memulai proses pelatihan dengan menjalankan skrip train.py dalam lingkungan Python.
2. --img 416: Menentukan ukuran gambar input yang akan digunakan selama pelatihan. Dalam hal ini, ukuran gambar input adalah 416x416 piksel.
3. --batch 10: Menentukan ukuran batch data yang digunakan selama pelatihan. Pada setiap iterasi, sebanyak 10 sampel data akan digunakan untuk memperbarui bobot model.
4. --epochs 250: Menentukan jumlah epochs atau iterasi yang akan dilakukan selama pelatihan. Dalam kasus ini, pelatihan akan dilakukan sebanyak 250 epochs.
5. --data data.yaml: Mengacu pada file data.yaml yang berisi konfigurasi dataset dan struktur model yang akan digunakan selama pelatihan.
6. --weights yolov5s.pt: Menentukan berat awal (pretrained weights) yang akan digunakan sebagai titik awal pelatihan. Dalam hal ini, model akan dimulai dengan berat awal dari model YOLOv5 versi kecil (yolov5s.pt).
7. --cache: Mengaktifkan caching untuk mempercepat proses baca/tulis data selama pelatihan dengan menyimpan data dalam cache.

Setelah menunggu beberapa saat, maka akan keluar file yang bernama “best.pt”.

#### 4. YOLO Import ke Tensorflow lite

Kemudian file best.pt di import dalam bentuk .tflite, yang bertujuan untuk diekspor ke direktori tensorflow lite dengan perintah :

```
“!python export.py --weights runs/train/exp/weights/best.pt --include tflite --img 416”
```

Keterangan :

1. python export.py: Memulai proses ekspor model dengan menjalankan skrip export.py dalam lingkungan Python.
2. --weights runs/train/exp/weights/best.pt: Menentukan jalur (path) ke berat terbaik (best.pt) dari model yang telah dilatih. Model inilah yang akan diekspor ke format TensorFlow Lite.
3. --include tflite: Menentukan bahwa model akan diekspor ke format TensorFlow Lite (TFLite), sehingga dapat digunakan di aplikasi mobile dan perangkat dengan konsumsi daya yang lebih rendah.
4. --img 416: Menentukan ukuran gambar input yang akan digunakan saat model dijalankan dalam format TFLite. Dalam hal ini, ukuran gambar input adalah 416x416 piksel.

Setelah menunggu beberapa saat, maka akan keluar file yang bernama “best-fp16.tflite”.

#### 5. Evaluasi

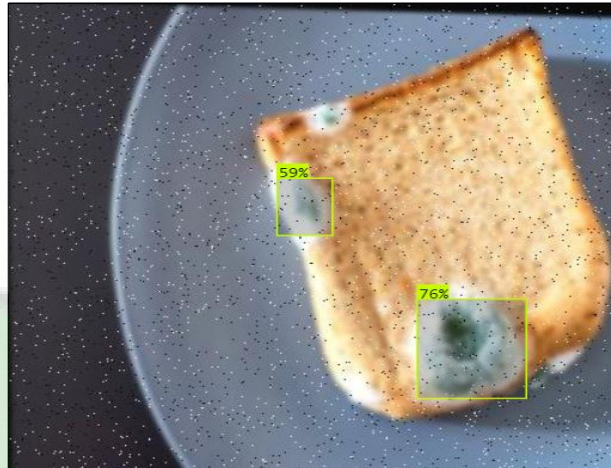
Sebelum dilakukan sebuah evaluasi mulai dari dataset hingga ke tensorflow lite untuk memperkecil miss ataupun error pada system. Jika tidak ada error, maka sudah bisa memasukkan best-fp16.tflite ke direktori \android\app\src\main\assets. Yang mana akan di lakukan export model tflite tersebut ke versi mobile pada Android Studio.

#### III.2.4. Uji Coba

Melakukan uji coba sistem untuk memastikan bahwa sistem dapat berfungsi dengan baik dan sesuai dengan kebutuhan. Dalam tahap uji coba ini

penulis menggunakan google collab untuk menguji model sistem yang sudah dikonversi pada implementasi diatas.

Berikut ini adalah contoh hasil uji coba pada model detector yang sudah dilatih:



Gambar III.14 Hasil Uji Coba

Dari gambar diatas, terlihat ada dua objek jamur yang terdeteksi dari uji coba model dataset yang sudah dilatih yang tingkat akurasinya mencapai 79 % dan 59 %. Model ini sudah bisa di terapkan untuk dikonversi ke tensorflow lite. Dan juga model detector ini sudah bisa di terapkan pada aplikasi mold detector.

### **III.2.5. Pengembangan dan Perbaikan**

Melakukan perbaikan dan pengembangan sistem berdasarkan hasil uji coba adalah salah satu tahap penting dalam siklus hidup pengembangan perangkat lunak. Tahap ini bertujuan untuk meningkatkan kualitas, kinerja, dan fungsionalitas sistem sesuai dengan kebutuhan pengguna.

## BAB IV

### HASIL DAN PEMBAHASAN

#### IV.1. Hasil

Tugas akhir ini merincikan pengembangan aplikasi berbasis mobile yang diberi nama "Mold Detector" (mold.apk), yang dikembangkan dengan menggunakan Java Android Project. Aplikasi ini memiliki kemampuan untuk mendeteksi keberadaan jamur pada permukaan roti dengan memanfaatkan kamera ponsel. Pengembangan aplikasi ini didasarkan pada pemanfaatan library TensorFlow Lite dan YOLOv5 untuk mengembangkan model deteksi objeknya.

Dengan mengintegrasikan YOLOv5, aplikasi ini mampu mendeteksi jamur secara real-time dengan tingkat akurasi yang tinggi. Hasil dari pengembangan aplikasi ini mencakup berbagai komponen (Tabel IV.2), di antaranya:

1. Tampilan ikon (Gambar IV.1) yang menggambarkan jamur dalam bentuk gambar kartun.
2. Splash screen (Gambar IV.2) yang muncul saat pertama kali membuka aplikasi.
3. Menu utama (Gambar IV.3) yang terdiri dari judul "Main Menu" dan tombol "Camera".
4. Halaman kamera (Gambar IV.4) yang aktif saat tombol "Camera" ditekan, berisi tampilan kamera ponsel serta tombol bottom sheet.
5. Halaman bottom sheet (Gambar IV.5) yang menyediakan pilihan perangkat (CPU, GPU, dan NNAPI), model yang digunakan (best-fp.tflite), dan informasi waktu inferensi.

6. Hasil deteksi (Gambar IV.6) yang menampilkan border box dan label hasil deteksi jamur. Untuk menentukan rata-rata hasil deteksi, misalnya dengan 3 objek terdeteksi seperti contoh 80,75, dan 83%, kemudian dapat dijumlahkan lalu membaginya dengan total objek yang terdeteksi. list keterangan nilai deteksi pada tabel IV.1.


Tabel IV.1 List Keterangan Deteksi

Daftar nilai	Keterangan
0-20%	Sangat Buruk
21-40%	Buruk
41-80%	Baik
81-100%	Sangat Baik

Nilai deteksi diukur dalam rentang 0 hingga 100%. Sebagai contoh, jika terdapat 3 objek yang terdeteksi dengan nilai masing-masing 80, 75, dan 83%, maka rata-rata nilai deteksinya adalah  $(80 + 75 + 83) / 3 = 79.33\%$ .

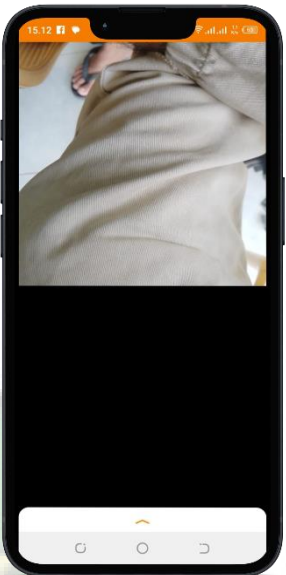
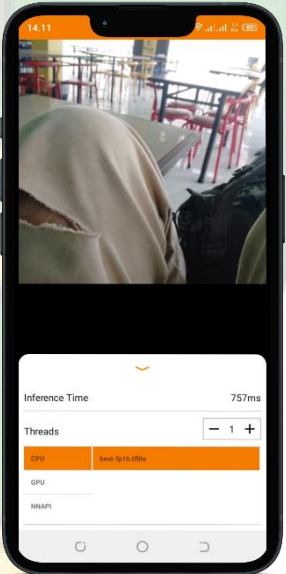
Dengan demikian, laporan ini menjelaskan pengembangan aplikasi "Mold Detector" yang memiliki kemampuan untuk mendeteksi jamur pada roti menggunakan teknologi terkini dalam pengolahan citra dan kecerdasan buatan.


Tabel IV.2 Komponen Aplikasi

No	Nama	Gambar
1.	Tampilan ikon	 <p>Gambar IV.1 Ikon Aplikasi</p>

<p>2.</p>	<p>Splash screen</p>	 <p>Gambar IV.2 Splashscreen</p>
<p>3.</p>	<p>Menu utama</p>	 <p>Gambar IV.3 Menu</p>



<p>4.</p>	<p>Halaman kamera</p>	 <p>Gambar IV.4 Halaman Camera</p>
<p>5.</p>	<p>Halaman bottom sheet</p>	 <p>Gambar IV.5 Halaman Bottom</p>

6.	Hasil deteksi	 <p>Gambar IV.6 Hasil Deteksi</p>
----	---------------	--

## IV.2. Pembahasan

Dalam hal ini Penulis akan menjabarkan tentang langkah-langkah implementasi yang dilakukan dalam menyelesaikan Aplikasi mold detector berdasarkan teori – teori yang sudah dipelajari oleh Penulis adalah sebagai berikut :

### 1. Perancangan Desain Sistem

Arsitektur sistem aplikasi mold detector berbasis mobile ini adalah sebuah aplikasi handphone android dengan menggunakan framework tensorflow lite sebagai untuk menjalankan model machine learning pada perangkat. Penulis juga membuat use case diagram, sequence diagram, diagram activity, class diagram, model yolov5 yang sudah dikonversi ke tensorflow lite (.tflite) untuk program aplikasi mold detector ini sebagai objek penelitian. Hal ini bertujuan untuk memperjelas arah dari program yang dibuat agar dapat lebih dimengerti oleh pembaca, membuat pembaca

mengerti menggunakan program ini dan menjelaskan kegunaan dari program ini.

## 2. Penyediaan kebutuhan aplikasi

Pada tahapan ini Penulis menyediakan kebutuhan yang dibutuhkan dalam perancangan aplikasi mold detector. Mulai dari perangkat keras (hardware) komputer dan handphone android hingga perangkat lunak (software) sistem operasi, bahasa pemrograman, dataset, dan framework.

## 3. Penulisan Coding Program ke Komputer

Penulis mengetikkan instruksi - instruksi (coding) rancangan sistem ke komputer sesuai dengan pembahasan pemrograman yang digunakan. Coding yang dituliskan didapat dari buku – buku dan coding yang terdapat di internet dengan memodifikasi dari coding yang ada.

## 4. Pengujian Sistem

Dilakukan untuk mengetahui apakah pekerjaan pemrograman telah dilakukan secara benar sehingga bisa menghasilkan fungsi – fungsi yang dikehendaki. Pengujian juga dimaksudkan untuk mengetahui keterbatasan dan kelemahan sistem yang dibuat agar dapat sebisa mungkin dilakukan penyempurnaan.

### **IV.3. Spesifikasi**

Aplikasi ini memanfaatkan library tensorflow lite untuk mengoperasikan model deteksi objek pada perangkat android. Model yang digunakan adalah YOLOv5, yang merupakan model berbasis neural network yang dirancang khusus untuk mendeteksi objek dalam gambar. Untuk menjalankan aplikasi ini, kita perlu mengikuti langkah-langkah berikut dibawah ini.

#### **IV.3.1. Menjalankan Melalui Android**

Untuk menjalankan aplikasi ini melalui android, dilakukan dengan cara :

1. Pastikan perangkat HP terhubung ke komputer melalui kabel USB dan dalam mode pengembang (Developer Mode). Aktifkan opsi pengembang dengan masuk ke Pengaturan, cari opsi "Tentang

Perangkat" atau "Informasi Perangkat", lalu ketuk nomor versi perangkat lunak beberapa kali hingga opsi pengembang aktif.

2. Pastikan juga bahwa driver USB yang sesuai telah diinstal pada komputer agar dapat mengenali perangkat HP.
3. Buka proyek Android Studio yang ingin di instal ke perangkat HP.
4. Di Android Studio, klik menu "Run" atau "Run App" (ikon segitiga hijau) atau tekan tombol Shift+F10 pada keyboard untuk memulai proses instalasi.
5. Android Studio akan memeriksa ketersediaan perangkat yang terhubung. Pilih perangkat HP dari daftar perangkat yang tersedia.
6. Jika perangkat tidak terdeteksi, pastikan bahwa USB Debugging (Pengelolaan USB) diaktifkan pada perangkat HP Anda melalui opsi pengembang. Setelah diaktifkan, Android Studio seharusnya dapat mendeteksi perangkat Anda.
7. Setelah memilih perangkat HP, Android Studio akan mengkompilasi dan menginstal aplikasi ke perangkat tersebut.
8. Tunggu hingga proses instalasi selesai. Setelah berhasil diinstal, aplikasi akan muncul di layar perangkat HP Anda dan siap untuk digunakan.

#### **IV.4. Pengujian deteksi jamur pada roti**

Pada bab ini, pengujian deteksi dijelaskan dengan memeriksa input ke sistem dan keluaran dari sistem. Bagian ini juga mencakup informasi tentang spesifikasi smartphone yang digunakan dalam pengujian (tabel IV.2), perangkat (tabel IV.3), pengujian deteksi jamur pada gambar dengan format jpg (tabel IV.4) dan sejenisnya, serta pengujian deteksi jamur menggunakan objek nyata yaitu roti (tabel IV.5).

##### **a. Spesifikasi smartphone**

Tabel IV.3 Spesifikasi Smartphone

Merek perangkat	Tecno Spark 8c
Sistem operasi	Android 11

Memori internal	64 GB
RAM	4 GB

b. **Perangkat**

Tabel IV.4 Pengujian Device

Perangkat	Inference time (ms)
CPU	750 - 770
GPU	250 - 280
NNAPI	756 -778

Pada device terdiri dari CPU, GPU dan NNAPI yang memiliki kelebihan dan kekurangan. berikut ini adalah kelebihan dan kekurangan dari device diatas:

1. CPU (*Central Processing Unit*):

Kelebihan:

1. Universal: Tersedia di semua perangkat, sehingga bisa digunakan di mana saja.
2. Fleksibilitas: Bisa mengatasi berbagai tugas dan model TFLite, terutama model kecil atau tugas sederhana.
3. Konsumsi Daya Rendah: Biasanya lebih efisien dalam penggunaan daya baterai dibandingkan GPU.

Kekurangan:

1. Kinerja Terbatas: Kinerjanya bisa terbatas terutama untuk model yang besar atau tugas kompleks.
2. Kecepatan Inferensi: Lebih lambat dibandingkan dengan GPU atau perangkat khusus untuk tugas ML.

2. GPU (*Graphics Processing Unit*):

Kelebihan:

1. Paralelisme: Mampu menangani tugas secara paralel, meningkatkan kecepatan inferensi, terutama pada tugas yang membutuhkan komputasi intensif.
2. Kinerja Tinggi: Memberikan kinerja inferensi yang lebih cepat, terutama pada model besar.

Kekurangan:

1. Penggunaan Daya Tinggi: Menggunakan lebih banyak daya dibandingkan dengan CPU.
2. Tidak Universal: Tidak semua perangkat memiliki GPU yang kuat atau bahkan GPU, sehingga tidak selalu tersedia.
3. NNAPI (*Neural Networks API*):

Kelebihan:

1. Optimasi Perangkat Keras: Didesain khusus untuk mempercepat inferensi jaringan saraf, memberikan kinerja yang lebih baik.
2. Optimasi Sumber Daya: Meminimalkan penggunaan daya, memori, dan sumber daya lainnya.

Kekurangan:

1. Ketergantungan Perangkat: Bergantung pada perangkat keras dan dukungan perangkat, mungkin tidak tersedia pada semua perangkat.
2. Kompleksitas Implementasi: Memerlukan penyesuaian khusus dalam kode aplikasi untuk optimalitas dan penerapan yang benar.

c. **Pengujian deteksi jamur pada foto (.jpeg dan sejenisnya)**

Tabel IV.5 Pengujian Deteksi Jamur Pada Foto (.Jpeg Dan Sejenisnya)



Pengujian	Threads	Jumlah objek terdeteksi	Rata-rata (%)	Keterangan
1	1	2	82,5	Sangat baik
2	2	3	84	Sangat baik
3	3	2	85	Sangat baik
4	4	2	82,5	Sangat baik
5	5	2	85	Sangat baik
6	6	2	86	Sangat baik
7	7	2	85	Sangat baik
8	8	2	87	Sangat baik

**d. Pengujian deteksi jamur pada objek nyata (roti)**

Tabel IV.6 Pengujian Deteksi Jamur Pada Objek Nyata (Roti)

Pengujian	Threads	Jumlah objek terdeteksi	Rata-rata (%)	Keterangan
1	1	2	80	Baik
2	2	3	79	Baik
3	3	4	82	Sangat Baik
4	4	4	81	Sangat Baik
5	5	5	85	Sangat Baik

6	6	5	85	Sangat Baik
7	7	5	85	Sangat Baik
8	8	5	87	Sangat Baik

Hasil deteksi di atas (tabel IV.5 dan IV.6) menunjukkan tingkat akurasi yang sangat tinggi. Namun, perlu diingat bahwa hasil tersebut dapat berfluktuasi karena beberapa faktor, seperti resolusi kamera smartphone yang dapat mempengaruhi hasil deteksi, terutama saat menggunakan kamera dengan resolusi yang lebih kecil. Selain itu, faktor cuaca juga dapat mempengaruhi hasil deteksi; misalnya, deteksi mungkin berbeda saat cuaca cerah dibandingkan dengan kondisi gelap. Serta, kualitas objek yang terdeteksi, seperti gambar yang buram, juga dapat memengaruhi hasil deteksi. Oleh karena itu, hasil deteksi pada sistem aplikasi ini tidak selalu konsisten.

#### **IV.5. Kelebihan dan Kekurangan**

Program aplikasi mold detector ini memiliki kelebihan dan kekurangan pada implementasinya di lingkungan user. Kelebihan dan kekurangan pada aplikasi dijelaskan di bawah ini.

##### **IV.5.1. Kelebihan**

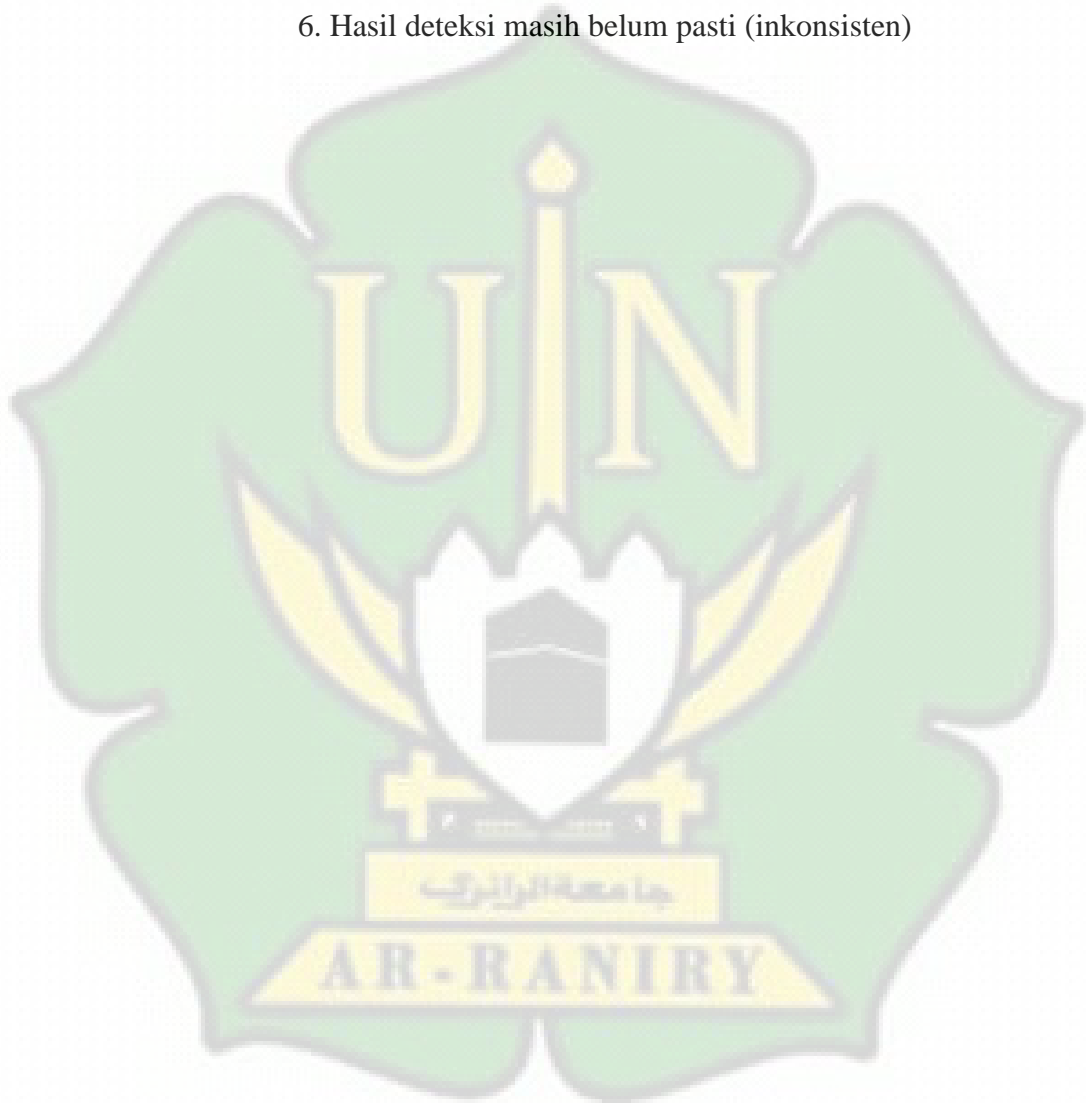
Kelebihan pada perancangan Aplikasi mold detector ini diantaranya yaitu:

1. Tampilan menarik.
2. Mudah digunakan.
3. Bisa digunakan pada spesifikasi hardware dan software yang rendah hingga tinggi.

##### **IV.5.2. Kekurangan**

Kekurangan atau kelemahan pada perancangan Aplikasi mold detector ini diantaranya yaitu:

1. Program aplikasi ini hanya bisa dijalankan pada sistem operasi android.
2. Tidak dilengkapi dengan suara.
3. Hanya memiliki 1 menu
4. Fitur yang terbatas
5. Dataset hanya 500 images
6. Hasil deteksi masih belum pasti (inkonsisten)



## **BAB V**

### **PENUTUP**

#### **V.1. Kesimpulan**

Kesimpulan dari rancang bangun ini adalah berhasilnya pengembangan aplikasi pendeteksi jamur berbasis mobile. Aplikasi ini menggunakan algoritma YOLO yang telah diadaptasi ke dalam model TFLite. Hasil uji coba menunjukkan tingkat keberhasilan deteksi jamur rata-rata di atas 50%, menegaskan potensi aplikasi ini untuk membantu pengguna mengidentifikasi jenis-jenis jamur secara lebih efektif dan efisien melalui perangkat mobile. Namun, perlu dicatat bahwa aplikasi ini saat ini hanya dapat mendeteksi jamur pada roti. Dalam pengembangan selanjutnya, diharapkan aplikasi dapat diperluas fungsinya untuk mendeteksi jamur pada berbagai jenis makanan lainnya.

#### **V.2. Saran**

Berikut adalah beberapa saran untuk pengembangan lebih lanjut tugas akhir ini:

##### **1. Penambahan Fitur:**

Saran pertama adalah menambahkan lebih banyak fitur ke aplikasi. Misalnya, integrasikan fitur yang memungkinkan pengguna untuk membagikan hasil identifikasi jamur ke media sosial atau menyimpan riwayat identifikasi.

##### **2. Perbaikan Antarmuka Pengguna (UI/UX):**

Selanjutnya, perlu dilakukan peningkatan pada antarmuka pengguna (UI/UX) aplikasi. Rancang ulang tata letak, warna, dan elemen desain agar aplikasi lebih mudah digunakan dan menarik bagi pengguna.

##### **3. Pengembangan Dataset:**

Salah satu langkah penting adalah memperluas dan memperbarui dataset yang digunakan. Perbanyak sampel gambar jamur dengan variasi jenis dan karakteristik untuk meningkatkan akurasi dan keandalan model deteksi jamur.

#### 4. Pengoptimalan Algoritma Deteksi:

Lakukan pengoptimalan lebih lanjut pada algoritma deteksi YOLO untuk meningkatkan akurasi dan kecepatan deteksi. Uji coba berbagai parameter dan teknik yang berbeda untuk mendapatkan hasil yang lebih baik.

#### 5. Integrasi Teknologi Terbaru:

Teliti dan pertimbangkan untuk mengintegrasikan teknologi terbaru yang dapat meningkatkan kinerja aplikasi, seperti pemrosesan citra canggih atau integrasi dengan asisten suara.

#### 6. Uji Coba dan Evaluasi Lanjutan:

Lakukan uji coba lebih lanjut dengan pengguna untuk mendapatkan umpan balik dan melakukan evaluasi performa aplikasi. Gunakan umpan balik ini untuk melakukan perbaikan dan peningkatan lebih lanjut.

#### 7. Optimalisasi Kinerja Aplikasi:

Lakukan optimalisasi kinerja aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan lancar dan responsif pada berbagai perangkat mobile.

## DAFTAR PUSTAKA

- Redmon, J. & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
- Bochkovski, A. (2020). YOLOv5. GitHub. <https://github.com/ultralytics/yolov5>
- Joseph, R. & Annamalai, A. (2020). YOLOv5: The Next Evolution of Object Detection. <https://towardsdatascience.com/yolov5-the-next-evolution-of-object-detection-22eb7f5e6b>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- Pitt, J. I., & Hocking, A. D. (2009). Fungi and food spoilage. Springer Science & Business Media.
- Beuchat, L.R., and Pitt, J.I. 1999. Detection and enumeration of spoilage fungi in foods. In Methods in Biotechnology, Vol. 13: Foodborne Pathogens: Microbiology and Molecular Biology, ed. R.C. McKellar, 99-114. Humana Press, Totowa, NJ.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc.
- Kelleher, J. D., Tierney, B., & Tierney, B. (2018). Data science an introduction. CRC press.
- Loudwyck, E., & Kurniawati, Y. E. (2022). Pendeteksian Sepeda Motor Yang Melintasi Trotoar Menggunakan Algoritma YOLOv3. KALBISIANA Jurnal Sains, Bisnis Dan ..., 8(1), 391–399.
- W. A. M. A. Wijesinghe dan R. N. Munasinghe, "A comparative study of cross-platform mobile application development frameworks for Android and iOS platforms," International Journal of Advanced Computer Science and Applications, vol. 10, no. 6, 2019.
- Wasril, A. R., Ghozali, M. S., & Mustafa, M. B. (2019). Pembuatan Pendeteksi Obyek Dengan Metode You Only Look Once (Yolo) Untuk Automated Teller Machine (Atm). *Majalah Ilmiah UNIKOM*, 17(1), 69–76. <https://doi.org/10.34010/miu.v17i1.2240>
- Hidalgo, A., Andrade, M.J., Guijarro, B., Álvarez-Ortega, S., García-Hierro, S., Alonso-Calleja, C., & Capita, R. (2018). Development of a PCR assay for the detection of fungal spoilage in bread. LWT - Food Science and Technology.
- Azimi, H., Ehsani, A., Shakeri, A., & Shishebor, M. (2021). Development of a biosensor for the detection of fungal contamination in bread. Journal of Food Protection.



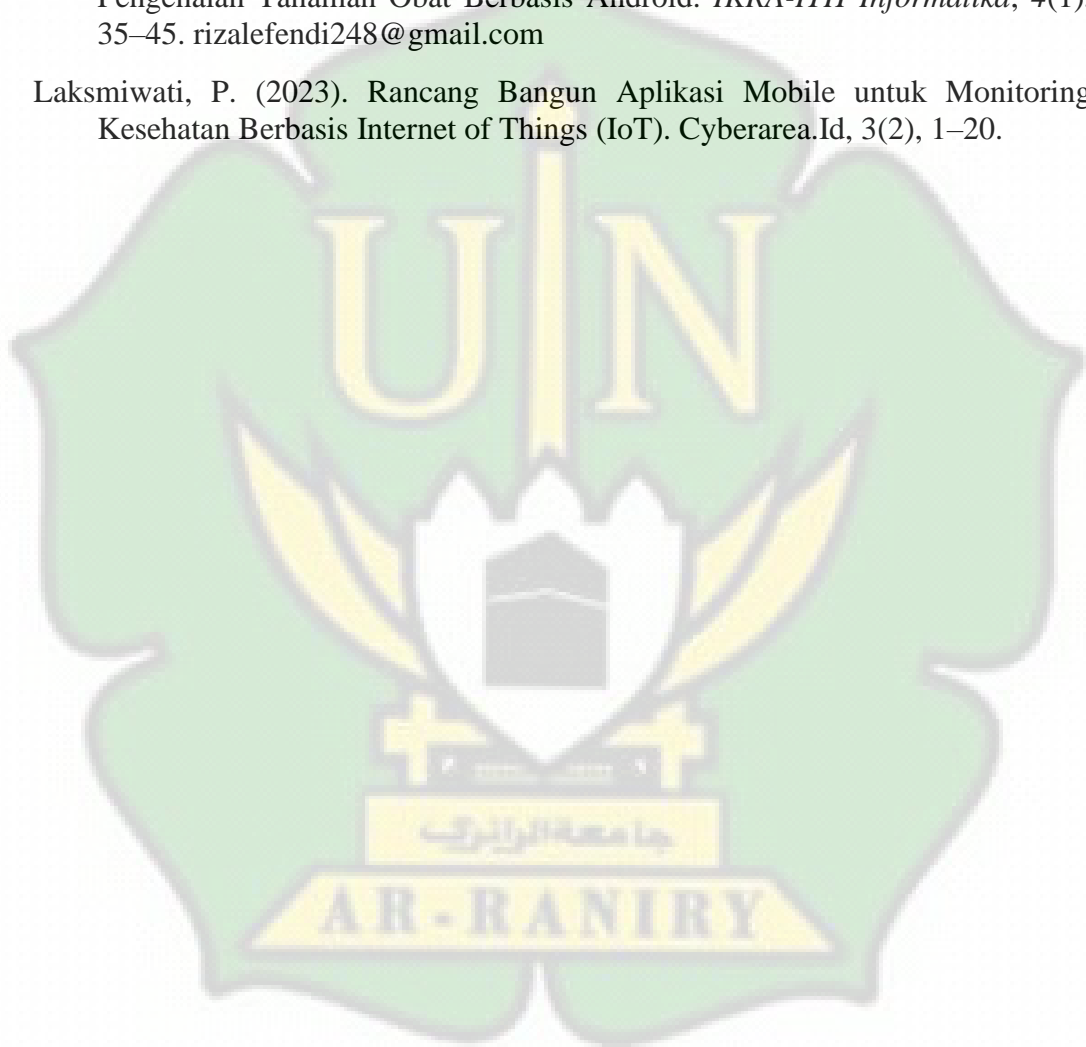
Alsing, Oscar. "Mobile Object Detection using TensorFlow Lite and Transfer Learning." 2018.

Fadly Aditya Putra, D. (2022). Rancang Bangun Aplikasi Mobile Pendeteksi Potensi Kanker. 2(3).

Pangestu, A. S., & Tanamal, R. (2020). Rancang Bangun Aplikasi Sistem Pakar Berbasis Mobile Untuk Mendiagnosis Penyakit Kulit Pada Kucing Persia. *Teknika*, 9(2), 81–87. <https://doi.org/10.34148/teknika.v9i2.279>

Rizal, E. (2020). Rancang Bangun Aplikasi Augmented Reality Untuk Deteksi Pengenalan Tanaman Obat Berbasis Android. *IKRA-ITH Informatika*, 4(1), 35–45. [rizalefendi248@gmail.com](mailto:rizalefendi248@gmail.com)

Laksmiwati, P. (2023). Rancang Bangun Aplikasi Mobile untuk Monitoring Kesehatan Berbasis Internet of Things (IoT). *Cyberarea.Id*, 3(2), 1–20.



# LAMPIRAN

## Lampiran 1 Implementasi YOLOv5 Ke tflite

1. Kode untuk menginstall yolov5 pada google colab.  
git clone <https://github.com/ultralytics/yolov5> # clone

```
%cd yolov5
```

```
%pip install -qr requirements.txt # install
```

```
import torch
```

```
import utils
```

```
display = utils.notebook_init()
```

2. Gambar hasil konversi dataset ke yolov5 pada google colab.

```
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
100/104 0.533G 0.03619 0.03307 0 76 416: 100% 105/105 [00:10<00:00, 9.73it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.76it/s]
all 18 95 0.617 0.537 0.435 0.16

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
101/104 0.533G 0.03625 0.03417 0 44 416: 100% 105/105 [00:11<00:00, 8.76it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.86it/s]
all 18 95 0.622 0.526 0.436 0.158

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
102/104 0.533G 0.03647 0.03274 0 16 416: 100% 105/105 [00:11<00:00, 9.51it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.01it/s]
all 18 95 0.627 0.526 0.443 0.165

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
103/104 0.533G 0.03615 0.03332 0 55 416: 100% 105/105 [00:11<00:00, 9.46it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 16.98it/s]
all 18 95 0.628 0.526 0.436 0.167

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
104/104 0.533G 0.03637 0.03369 0 37 416: 100% 105/105 [00:11<00:00, 9.37it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.69it/s]
all 18 95 0.644 0.547 0.471 0.17

105 epochs completed in 0.340 hours.
Optimizer stripped from runs/train/exp8/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp8/weights/best.pt, 14.3MB

Validating runs/train/exp8/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:01<00:00, 2.86it/s]
all 18 95 0.742 0.474 0.528 0.216

Results saved to runs/train/exp8
```

3. Gambar hasil konversi yolov5 ke model tflite pada google colab.

```
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
100/104 0.533G 0.03619 0.03307 0 76 416: 100% 105/105 [00:10<00:00, 9.73it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.76it/s]
all 18 95 0.617 0.537 0.435 0.16

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
101/104 0.533G 0.03625 0.03417 0 44 416: 100% 105/105 [00:11<00:00, 8.76it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.86it/s]
all 18 95 0.622 0.526 0.436 0.158

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
102/104 0.533G 0.03647 0.03274 0 16 416: 100% 105/105 [00:11<00:00, 9.51it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.01it/s]
all 18 95 0.627 0.526 0.443 0.165

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
103/104 0.533G 0.03615 0.03332 0 55 416: 100% 105/105 [00:11<00:00, 9.46it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 16.98it/s]
all 18 95 0.628 0.526 0.436 0.167

Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
104/104 0.533G 0.03637 0.03369 0 37 416: 100% 105/105 [00:11<00:00, 9.37it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:00<00:00, 17.69it/s]
all 18 95 0.644 0.547 0.471 0.17

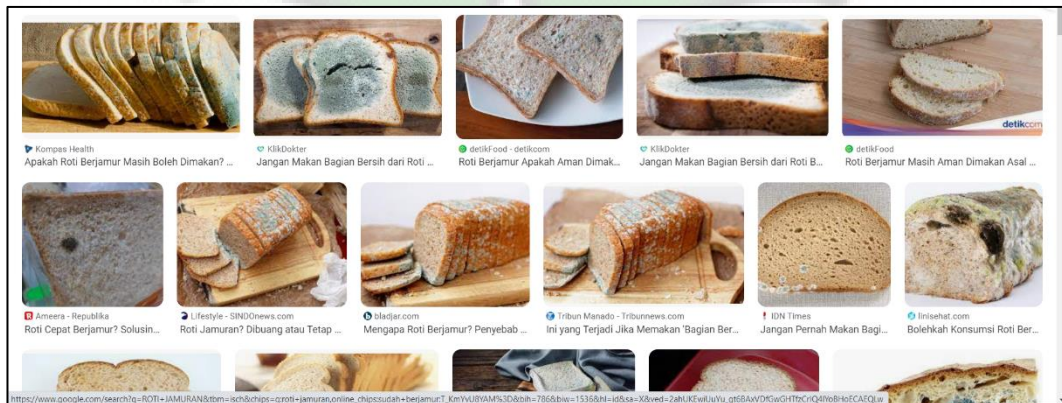
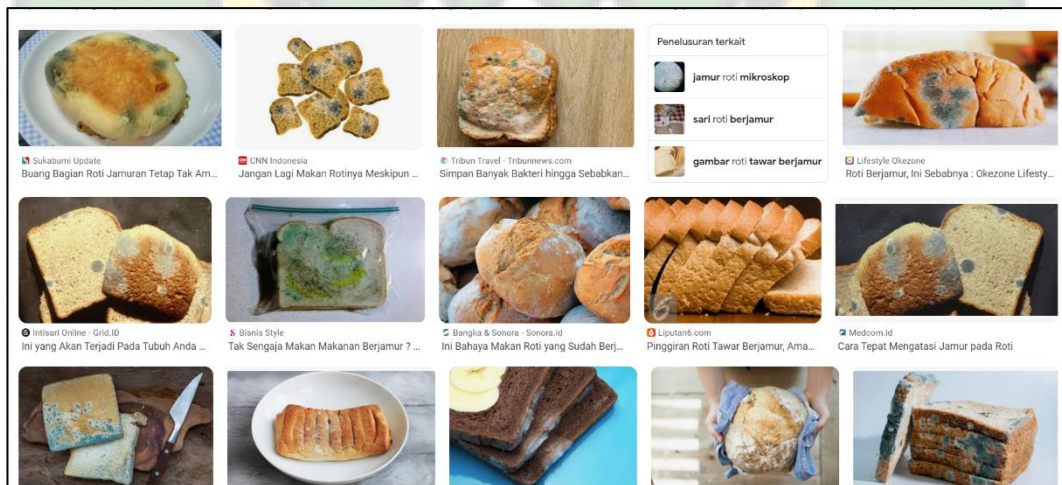
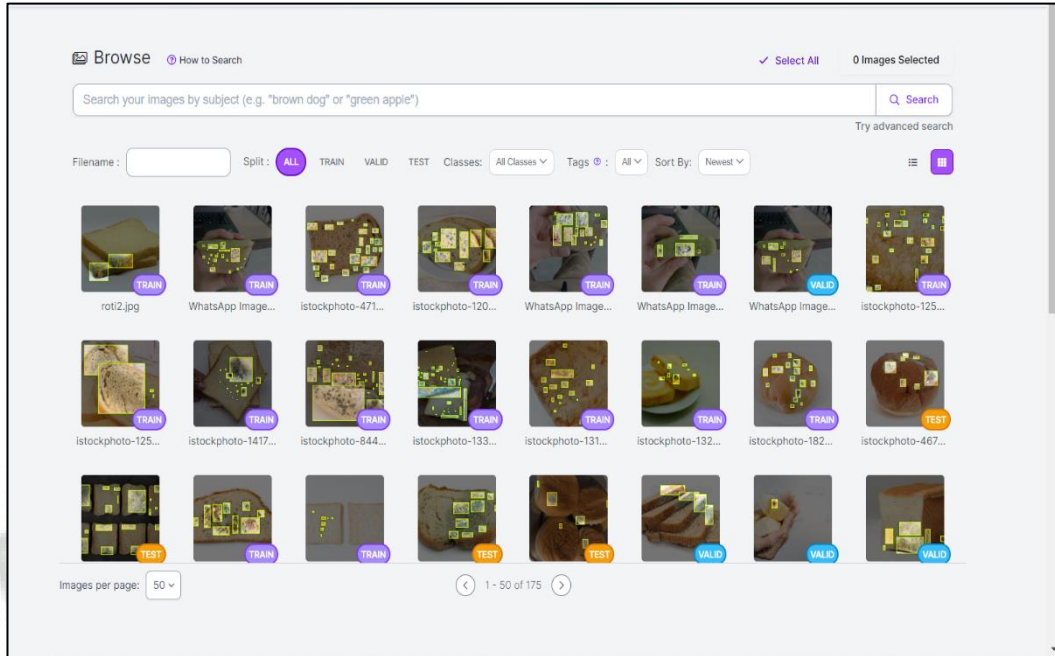
105 epochs completed in 0.340 hours.
Optimizer stripped from runs/train/exp8/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp8/weights/best.pt, 14.3MB

Validating runs/train/exp8/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
Class Images Instances P R mAP50 mAP50-95: 100% 3/3 [00:01<00:00, 2.86it/s]
all 18 95 0.742 0.474 0.528 0.216

Results saved to runs/train/exp8
```

## Lampiran 2 image yang digunakan untuk dataset

1. Kumpulan gambar roti berlabel “jamur” yang sudah dimasukkan pada web roboflow.



## Lampiran 3 kode aplikasi

### 1. Kode AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="org.tensorflow.lite.examples.detection">

    <uses-sdk />

    <uses-feature
        android:glEsVersion="0x00030001"
        android:required="true" />

    <uses-permission
android:name="android.permission.CAMERA" />

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature
android:name="android.hardware.camera.autofocus" />

    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_type"
        android:label="@string/tfe_od_app_name"
        android:roundIcon="@mipmap/ic_type_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".HomeActivity"
            android:exported="false">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity android:name=".DetectorActivity" />
        <activity android:name=".Splashscreen">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

## 2. Kode splashscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/orange_primary_dark"
tools:context=".Splashscreen">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="225dp"
        android:layout_height="0dp"
        android:layout_marginTop="213dp"
        android:layout_marginBottom="29dp"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/type" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="39dp"
        android:layout_marginBottom="275dp"
        android:text="Mold Detector APP"
        android:textSize="17sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="@+id/imageView4"
        app:layout_constraintTop_toBottomOf="@+id/imageView4" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## 3. Kode main activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/white"
android:orientation="vertical"
tools:context=".HomeActivity">

    <Button
        android:id="@+id/cameraButton"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:drawableLeft="@drawable/camera"
        android:text="Camera"

        android:textColor="@color/design_default_color_error"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3"
    />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="295dp"
        android:layout_height="160dp"
        android:layout_marginStart="74dp"
        android:layout_marginTop="41dp"
        android:layout_marginEnd="74dp"
        android:layout_marginBottom="18dp"
        app:layout_constraintBottom_toTopOf="@+id/textView3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/chou" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="80dp"
        android:layout_marginBottom="55dp"
        android:text="Main Menu"
        android:textSize="25dp"

        app:layout_constraintBottom_toTopOf="@+id/cameraButton"
        app:layout_constraintEnd_toEndOf="@+id/imageView3"

        app:layout_constraintTop_toBottomOf="@+id/imageView3" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### 4. Kode camera\_activity.xml

```

<?xml version="1.0" encoding="utf-8"?><!--
-->

<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00000000">

    <RelativeLayout

```



```

        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@android:color/black"
        android:orientation="vertical">

        <FrameLayout
            android:id="@+id/container"
            android:layout_width="match_parent"
            android:layout_height="match_parent"

tools:context="org.tensorflow.demo.CameraActivity" />

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:layout_alignParentTop="true"
            android:background="@color/semi_transparent">

        </androidx.appcompat.widget.Toolbar>

    </RelativeLayout>

    <include
        android:id="@+id/bottom_sheet_layout"
        layout="@layout/tfe_od_layout_bottom_sheet"
        />
</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Keterangan:

Kode di atas hanya menunjukkan tampilan antarmukanya dan belum mencakup kode untuk aktivitas aplikasi. Untuk informasi lebih lengkap, dapat mengunjungi tautan

[https://drive.google.com/drive/folders/1Bh2oivFhQ356bWLR48HDhjZ\\_shdiaNwE?usp=drive\\_link](https://drive.google.com/drive/folders/1Bh2oivFhQ356bWLR48HDhjZ_shdiaNwE?usp=drive_link)

Untuk dataset lengkap bisa mengunjungi tautan

<https://universe.roboflow.com/muhammad-ghazi-siddiq-a6hso/bread-mold-yb186/dataset/10>