

# ANDROID APPLICATION PROTOTYPE FOR DETECTING MOULD ON BREAD USING MACHINE LEARNING

## PROTOTYPE APLIKASI ANDROID UNTUK MENDETEKSI JAMUR PADA ROTI MENGGUNAKAN MACHINE LEARNING

Daniel Haganta Barus<sup>1</sup>, Clara Stephanie Bernadeth Piay<sup>2</sup>, Frissy Siregar<sup>3</sup>, Evta Indra<sup>4</sup>  
<sup>1,2,3,4</sup> Universitas Prima Indonesia, Jl. Sampul No.3, Kota Medan, Sumatera Utara, Indonesia  
Email: danielhaganta27@gmail.com<sup>1</sup>, clarapiay02@gmail.com<sup>2</sup>, frissysiregar@gmail.com<sup>3</sup>,  
evtaindra@unprimdn.ac.id<sup>4</sup>

**Abstract** - Mould contamination in bread poses a serious health risk if not detected early, especially in the food industry, which still relies heavily on manual visual inspection. This study aims to develop a prototype Android application capable of automatically detecting mould on bread using a machine learning approach based on the MobileNetV2 architecture. The classification model was trained on a dataset of 666 bread images, consisting of 533 training and 133 validation samples. Training was carried out over 37 epochs using data augmentation techniques and a learning rate of 0.0001. The results demonstrated consistent accuracy improvements and loss reductions without signs of overfitting. The model achieved 94% testing accuracy, with a precision, recall, and F1-score of 0.94 for both "Mouldy" and "Non-Mouldy" classes. The confusion matrix showed 125 correct predictions out of 133 test images. This research addresses the gap in lightweight and practical solutions for mobile-based mould detection. Unlike previous studies that used heavier models such as VGG16 or ResNet, this study shows that MobileNetV2 can achieve high performance with lower computational demands, making it suitable for real-world Android applications. The trained model was integrated into a simple Android interface, allowing users to upload images and instantly receive classification results. For future improvement, this prototype can be enhanced by incorporating object detection or image segmentation techniques such as YOLOv5 or U-Net to enable not only classification but also the localisation of mould areas in real-time.

**Keywords:** Android application, bread classification, machine learning (MobileNetV2), mould detection.

**Abstrak** - Kontaminasi jamur pada roti dapat menimbulkan risiko kesehatan yang serius apabila tidak terdeteksi sejak dini, terutama dalam industri makanan yang masih mengandalkan inspeksi visual secara manual. Penelitian ini bertujuan untuk mengembangkan prototipe aplikasi Android yang mampu mendeteksi keberadaan jamur pada roti secara otomatis menggunakan metode machine learning berbasis arsitektur MobileNetV2. Model dilatih menggunakan 666 gambar roti, terdiri dari 533 data pelatihan dan 133 data validasi. Proses pelatihan berlangsung selama 37 epoch dengan penerapan teknik augmentasi data serta pengaturan learning rate sebesar 0.0001. Hasil pelatihan menunjukkan peningkatan akurasi dan penurunan nilai loss yang konsisten tanpa indikasi overfitting. Model menghasilkan akurasi pengujian sebesar 94%, dengan nilai precision, recall, dan f1-score sebesar 0,94 untuk kedua kelas Berjamur dan Tidak Berjamur. Confusion matrix menunjukkan 125 prediksi yang benar dari 133 gambar uji. Penelitian ini mengisi celah riset terkait terbatasnya solusi ringan dan praktis untuk deteksi jamur yang dapat langsung diterapkan pada perangkat mobile. Berbeda dengan studi sebelumnya yang menggunakan model berat seperti VGG16 atau ResNet, penelitian ini membuktikan MobileNetV2 dapat memberikan performa tinggi dengan efisiensi komputasi yang lebih baik, sehingga cocok untuk implementasi di aplikasi Android. Model yang dikembangkan telah diintegrasikan ke dalam antarmuka aplikasi Android sederhana yang memungkinkan pengguna mengunggah gambar dan mendapatkan hasil klasifikasi secara langsung. Ke depannya, prototipe ini masih dapat ditingkatkan dengan mengintegrasikan teknik object detection seperti YOLOv5 atau U-Net agar sistem tidak hanya melakukan klasifikasi gambar secara menyeluruh, tetapi juga mampu mendeteksi dan menandai lokasi spesifik pertumbuhan jamur.

**Kata Kunci** - aplikasi android, klasifikasi roti, pembelajaran mesin mobilenetv2, deteksi jamur.

## I. PENDAHULUAN

Pertumbuhan jamur pada roti merupakan masalah serius dalam industri pangan karena dapat menurunkan kualitas produk dan membahayakan kesehatan konsumen. Jamur seperti *Penicillium*, *Aspergillus*, dan *Rhizopus* sering ditemukan pada roti yang disimpan terlalu lama atau dalam kondisi lembap[1]. Beberapa jenis jamur juga menghasilkan mikotoksin yang dapat menimbulkan gejala seperti demam, batuk, nyeri dada, hingga gangguan sistem pernapasan[2][3]. Meskipun pemeriksaan visual masih lazim digunakan, pendekatan ini rentan terhadap kesalahan manusia dan tidak efisien untuk skala produksi besar. Permasalahan utama dalam deteksi jamur pada roti saat ini adalah ketiadaan sistem otomatis yang cepat, akurat, dan dapat dijalankan pada perangkat ringan seperti smartphone. Produk yang telah terkontaminasi jamur berisiko tetap beredar di pasaran karena keterbatasan inspeksi manual. Oleh sebab itu, dibutuhkan solusi berbasis teknologi yang mampu melakukan deteksi dini secara otomatis[4]. Penelitian ini akan memberikan solusi bagaimana membangun sebuah sistem klasifikasi otomatis untuk mendeteksi jamur pada roti menggunakan model deep learning yang efisien dan cocok diterapkan di perangkat mobile[5]. Untuk menjawab hal tersebut, studi ini mengembangkan model klasifikasi gambar berbasis MobileNetV2, yang dikenal ringan namun tetap akurat, dan mengintegrasikannya ke dalam aplikasi Android yang dapat digunakan secara langsung oleh konsumen[6].

Beberapa penelitian sebelumnya telah memanfaatkan model CNN seperti VGG16, ResNet, dan Inception untuk deteksi jamur atau kontaminasi pangan, dan berhasil mencapai akurasi tinggi di atas 95% dalam lingkungan laboratorium. Namun, model-model tersebut cenderung berat dan membutuhkan sumber daya komputasi tinggi, sehingga kurang ideal untuk implementasi mobile. Saat ini, masih sangat terbatas penelitian yang mengeksplorasi penggunaan arsitektur lightweight seperti MobileNetV2 dalam aplikasi Android untuk deteksi jamur. Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengisi celah riset tersebut dengan membangun prototipe sistem deteksi jamur pada roti berbasis MobileNetV2 yang ringan, akurat, dan praktis. Sistem ini diharapkan mampu mendeteksi roti berjamur secara otomatis dan cepat dari perangkat mobile.

## II. SIGNIFIKASI STUDI

### A. Studi Literature

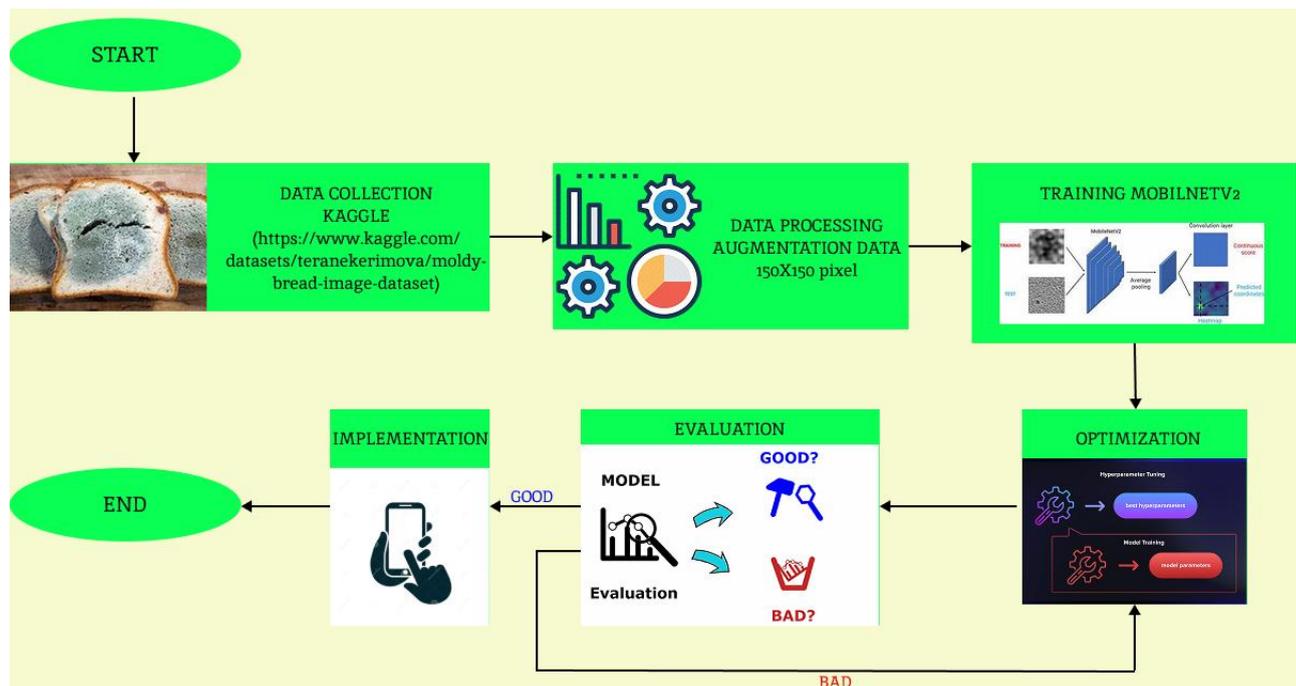
Deteksi jamur pada produk pangan, khususnya roti, memainkan peranan penting dalam menjamin keamanan konsumsi dan menjaga kualitas produk. Jamur seperti *Penicillium*, *Aspergillus*, dan *Rhizopus* tidak hanya merusak tampilan, rasa, dan tekstur roti, tetapi juga berpotensi menghasilkan mikotoksin berbahaya bagi kesehatan manusia. Dalam kondisi industri makanan saat ini yang masih banyak mengandalkan inspeksi visual manual, dibutuhkan sistem deteksi otomatis berbasis citra digital yang efisien dan andal[7]. Dalam beberapa tahun terakhir, metode deep learning, khususnya Convolutional Neural Network (CNN), telah menjadi pendekatan unggulan dalam pemrosesan citra. CNN memiliki kemampuan mengekstraksi fitur spasial seperti tekstur, warna, dan bentuk secara otomatis melalui kombinasi lapisan konvolusi dan pooling, menjadikannya sangat efektif dalam tugas klasifikasi visual, termasuk deteksi kontaminasi pangan[8][9].

Seiring meningkatnya kebutuhan akan solusi yang efisien untuk perangkat terbatas, arsitektur MobileNetV2 menjadi pilihan yang ideal. MobileNetV2 dirancang untuk efisiensi komputasi tinggi dengan tetap mempertahankan akurasi, melalui penggunaan depthwise separable convolution yang mengurangi jumlah parameter secara signifikan. Arsitektur ini sangat cocok untuk implementasi di perangkat mobile dan sistem embedded.[10].

Berbagai studi sebelumnya telah berhasil menerapkan MobileNetV2 dalam bidang klasifikasi citra seperti diagnosis penyakit tanaman, analisis citra medis, dan inspeksi mutu makanan [11]. Salah satu studi terkini oleh [12] menunjukkan bahwa MobileNetV2 mampu mencapai akurasi 99,48% untuk roti putih dan 99,72% untuk roti gandum dengan citra mikroskopik beresolusi tinggi. Studi tersebut dilakukan dalam lingkungan laboratorium yang terkontrol, dengan hasil sangat optimal tetapi belum siap langsung diadopsi ke perangkat konsumen. Sebagai pembandingan, penelitian ini menggunakan gambar roti dengan kondisi realistis dan pencahayaan beragam, serta berhasil mencapai akurasi 94%. Meski sedikit lebih rendah, keunggulan studi ini terletak pada kemampuannya diimplementasikan ke dalam aplikasi Android ringan yang dapat digunakan langsung oleh masyarakat luas, termasuk konsumen rumah tangga, pelaku UMKM roti, dan regulator pangan seperti BPOM atau lembaga sertifikasi mutu. Dengan demikian, penelitian ini tidak hanya berkontribusi pada aspek teknis klasifikasi citra, tetapi juga memberikan solusi praktis dan siap pakai untuk deteksi jamur. Sistem ini dapat mempercepat proses inspeksi visual, mengurangi potensi distribusi produk rusak ke konsumen, dan membantu UMKM menjaga kualitas produksi secara mandiri. Secara keseluruhan, penelitian ini mengembangkan model MobileNetV2 yang dilatih untuk membedakan antara gambar roti berjamur dan tidak berjamur, dengan tujuan akhir menciptakan sistem deteksi jamur yang cepat, akurat, efisien, dan mudah diakses oleh semua lapisan pemangku kepentingan dalam rantai produksi dan konsumsi roti [13].

*B. Metode Penelitian*

Penelitian ini menghadirkan sebuah solusi melalui penerapan metode Machine Learning dengan memanfaatkan model Convolutional Neural Network (CNN) untuk mendeteksi keberadaan jamur pada roti secara otomatis. Proses implementasi dilakukan menggunakan bahasa pemrograman Python, dengan tahapan yang dirancang sebagaimana ditunjukkan pada Gambar 1.

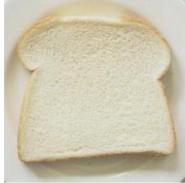


Gambar 1. Alur Penelitian

*1. Pengumpulan Data*

Data yang digunakan dalam penelitian ini untuk melatih model CNN berasal dari dataset berjumlah 666 gambar, yang terdiri atas 334 gambar roti berjamur dan 332 gambar roti tidak berjamur. Seluruh gambar diperoleh dari Kaggle, tepatnya dari dataset bertema mold on bread.

Tabel 1. Jumlah dari masing-masing kelas

Sampel	Jumlah Data	Sumber Data
Berjamur 	334	Kaggle (Mould on bread)
Tidak Berjamur 	332	

## 2. Pemrosesan Data

Sebelum digunakan dalam proses pelatihan, data harus melalui tahap pra-pemrosesan agar model machine learning dapat mengenali pola dengan lebih baik dan menghasilkan prediksi yang lebih akurat. Tahap ini mencakup beberapa proses penting, seperti normalisasi, augmentasi data, dan pembersihan data untuk menghilangkan noise atau inkonsistensi yang dapat mengganggu kinerja model saat belajar. Salah satu strategi utama yang digunakan dalam penelitian ini adalah augmentasi data, yaitu teknik yang bertujuan untuk memperkaya variasi fitur dalam dataset dan meningkatkan kemampuan model dalam melakukan generalisasi. Beberapa teknik augmentasi yang diterapkan antara lain:

1. Rotasi – Memutar gambar pada sudut tertentu untuk menciptakan variasi orientasi.
2. Pergeseran horizontal (width shift) – Menggeser posisi gambar secara horizontal untuk mengubah letak objek.
3. Pergeseran vertikal (height shift) – Menggeser gambar secara vertikal untuk memberikan variasi posisi.
4. Shearing – Mengubah bentuk gambar dengan memiringkan bagian tertentu secara horizontal atau vertikal.
5. Zooming – Memperbesar bagian tertentu dari gambar untuk menciptakan variasi skala.
6. Horizontal flip – Membalik gambar dari kiri ke kanan untuk membuat citra cermin dari gambar asli.

Dengan menerapkan teknik-teknik augmentasi tersebut, jumlah data secara efektif bertambah tanpa perlu menambahkan gambar asli baru. Hal ini membantu mengurangi risiko overfitting, meningkatkan kemampuan model dalam mengenali pola pada data yang belum pernah dilihat sebelumnya, serta memperbaiki akurasi prediksi. Setelah melalui tahap pra-pemrosesan, dataset dibagi menjadi dua bagian, yaitu 533 gambar untuk pelatihan (training) dan 133 gambar untuk validasi, dengan ukuran gambar yang telah diubah menjadi  $150 \times 150$  piksel.

## 3. Pelatihan Model

Penelitian ini menggunakan Convolutional Neural Network (CNN), yaitu salah satu metode deep learning yang paling umum digunakan dalam klasifikasi citra. CNN memproses gambar input melalui sejumlah lapisan konvolusi yang secara otomatis mengekstraksi fitur visual penting seperti warna, tekstur, dan bentuk sehingga tidak memerlukan proses ekstraksi fitur secara manual. Arsitektur CNN terdiri atas lapisan konvolusi dan pooling, yang memungkinkan model mengenali pola visual yang

kompleks serta mampu melakukan generalisasi meskipun gambar memiliki pencahayaan atau sudut pengambilan yang berbeda-beda[14]. Untuk meningkatkan efisiensi komputasi, penelitian ini menggunakan MobileNetV2 sebagai model dasar. MobileNetV2 dirancang secara khusus untuk aplikasi ringan dan sangat cocok digunakan pada gambar berukuran kecil maupun untuk implementasi di perangkat mobile[15]. Proses konvolusi pada arsitektur ini dilakukan dalam dua tahap, yaitu depthwise convolution dan pointwise convolution, yang bertujuan untuk mengurangi jumlah parameter namun tetap mempertahankan kinerja model. Proses ini juga dapat dijelaskan secara matematis melalui operasi per-pixel, yang semakin menegaskan efisiensi komputasi MobileNetV2 tanpa mengorbankan performa klasifikasi.

1. **Depthwise Convolution**

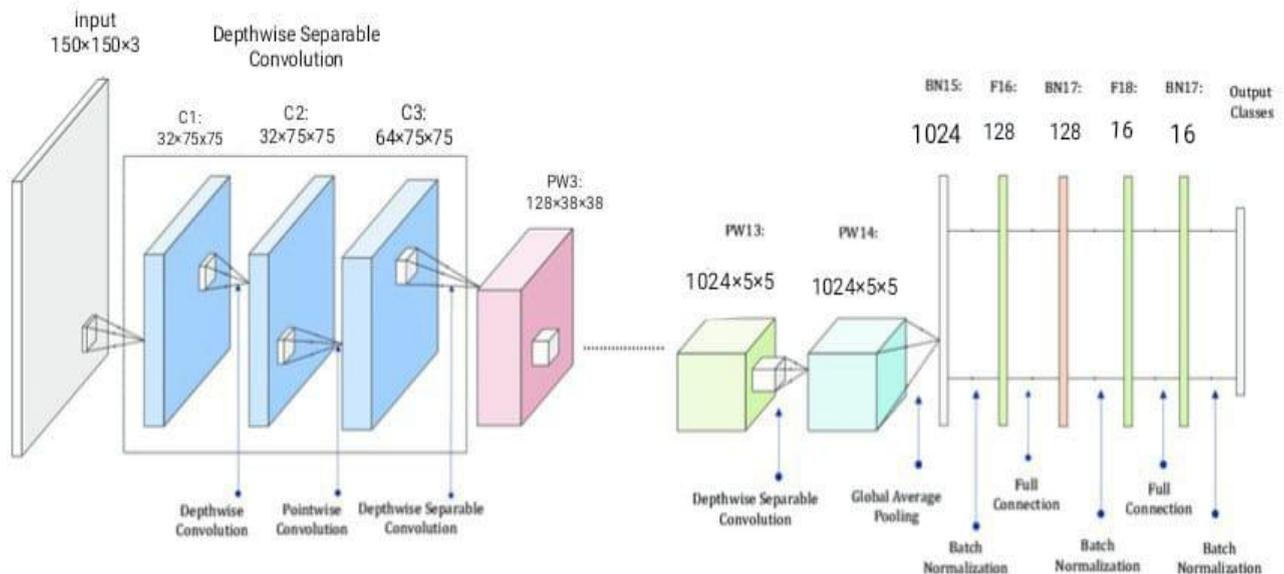
$$y_{i,j,k} = \sum_m X_{i+m,j+n,k} \cdot W_{m,n,k} \quad (1)$$

- $i, j, k$  = Nilai keluaran (peta fitur yang dihasilkan dari konvolusi) pada posisi (i, j) pada saluran ke-k
- $X_{i+m,j+n,k}$  = Nilai masukan (peta gambar/fitur) pada posisi (i+m, j+n) di saluran ke-k
- $W_{m,n,k}$  = Bobot filter kedalaman (kernel) berukuran  $m \times n$  untuk saluran k
- $m, n$  = Indeks dimensi filter (misalnya untuk filter  $3 \times 3$ , m dan n = 0, 1, 2)
- $k$  = Saluran k, karena setiap saluran memiliki filturnya sendiri.

2. **Pointwise Convolution (1x1)**

$$y^1_{i,j,k} = \sum_c y_{i,j,c} \cdot W^1_{c,k} \quad (2)$$

- $y^1_{i,j,k}$  = Hasil akhir dari konversi titik demi titik pada posisi (i, j) untuk saluran k
- $y_{i,j,c}$  = Input (hasil depthwise) pada posisi (i, j) pada saluran c
- $W^1_{c,k}$  = Bobot filter titik demi titik (1x1) dari saluran masukan ke-c ke saluran keluaran ke-k
- $c$  = Saluran masukan (sebelum diubah menjadi saluran keluaran secara pointwise)
- $k$  = Saluran keluaran baru (hasil gabungan dari beberapa saluran masukan)



Gambar 2. Diagram Blok Arsitektur MobileNetV2

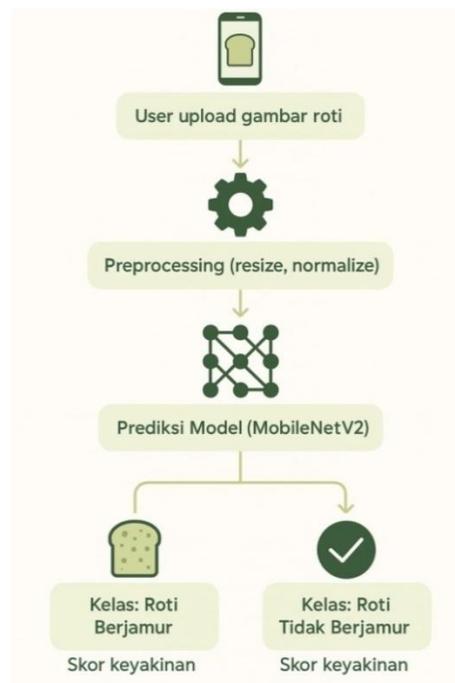
Model menerima masukan berupa gambar berukuran  $150 \times 150 \times 3$  piksel, yang kemudian diproses melalui beberapa tahap depthwise separable convolution untuk mengekstraksi fitur awal. Selanjutnya, fitur yang telah diperoleh diperkuat melalui lapisan pointwise convolution, hingga menghasilkan output berdimensi  $128 \times 38 \times 38$ . Proses dilanjutkan dengan beberapa lapisan konvolusi tambahan dan global pooling, sehingga fitur diratakan menjadi dimensi  $1024 \times 5 \times 5$ .

Setelah itu, fitur yang telah diringkas diteruskan ke beberapa lapisan fully connected dan batch normalization, yang bertugas menstabilkan distribusi data selama pelatihan. Proses ini menghasilkan vektor akhir yang siap untuk diklasifikasikan ke dalam kelas output, yaitu berjamur atau tidak berjamur.

Tabel 2. Arsitektur MobilnetV2

Layer	Jenis Layer	Ukuran Output (H×W×C)	Keterangan
Input	Input Gambar	$150 \times 150 \times 3$	Gambar RGB
Conv2D (stride=2)	Standard Conv (3×3)	$75 \times 75 \times 32$	32 filter, padding=sama
Bottleneck 1	Depthwise + Pointwise	$75 \times 75 \times 16$	t=1, c=16, n=1, s=1
Bottleneck 2	Depthwise + Pointwise	$38 \times 38 \times 24$	t=6, c=24, n=2, s=2
Bottleneck 3	Depthwise + Pointwise	$19 \times 19 \times 32$	t=6, c=32, n=3, s=2
Bottleneck 4	Depthwise + Pointwise	$10 \times 10 \times 64$	t=6, c=64, n=4, s=2
Bottleneck 5	Depthwise + Pointwise	$10 \times 10 \times 96$	t=6, c=96, n=3, s=1
Bottleneck 6	Depthwise + Pointwise	$5 \times 5 \times 160$	t=6, c=160, n=3, s=2
Bottleneck 7	Depthwise + Pointwise	$5 \times 5 \times 320$	t=6, c=320, n=1, s=1
GlobalAveragePooling2D	Global Pooling	$1 \times 1 \times 320$	Diratakan menjadi 320
Dense (FC1)	Fully Connected + ReLU	$1 \times 512$	512 neurons
Dropout (rate=0.3)	Regularization Layer	$1 \times 512$	Mencegah overfitting
Dense (Output Layer)	Fully Connected + Sigmoid	$1 \times 1$	1 neuron untuk klasifikasi biner

Model klasifikasi ini menerima input berupa gambar RGB berukuran  $150 \times 150$  piksel. Gambar tersebut pertama-tama diproses melalui lapisan konvolusi awal dengan stride 2 dan padding sama (same padding), yang menghasilkan peta fitur berukuran  $75 \times 75 \times 32$ . Selanjutnya, model menggunakan serangkaian blok bottleneck MobileNetV2, yang terdiri dari kombinasi depthwise convolution dan pointwise convolution. Blok-blok ini secara bertahap mengurangi dimensi spasial dan meningkatkan kedalaman fitur, dimulai dari ukuran  $75 \times 75 \times 16$  hingga menjadi  $5 \times 5 \times 320$ . Setelah proses ekstraksi fitur selesai, dilakukan global average pooling untuk meratakan fitur menjadi vektor berukuran  $1 \times 1 \times 320$ . Vektor ini kemudian diteruskan ke lapisan fully connected (FC) dengan 512 neuron dan fungsi aktivasi ReLU. Untuk mencegah overfitting, diterapkan dropout sebesar 0.5. Pada tahap akhir, digunakan lapisan output dengan 1 neuron dan aktivasi sigmoid untuk melakukan klasifikasi biner, yaitu menentukan apakah gambar roti tersebut berjamur atau tidak berjamur. Model MobileNetV2 yang telah selesai dilatih kemudian dikonversi ke dalam format TensorFlow Lite (.tflite) agar dapat digunakan pada sistem operasi Android. Aplikasi dikembangkan menggunakan Android Studio dengan bahasa Kotlin, sedangkan desain antarmuka (UI) dibuat menggunakan komponen XML dan Jetpack. Setelah pengguna mengunggah gambar roti, hasil klasifikasi ditampilkan langsung pada layar aplikasi.



Gambar 3. User Flow Diagram

Pada gambar diatas ditampilkan bahwa dalam aplikasi ini, pengguna dapat memilih untuk mengunggah gambar roti dari galeri atau mengambil foto secara langsung melalui kamera perangkat. Gambar yang diperoleh kemudian melewati tahap pra-pemrosesan, seperti pengubahan ukuran (resizing) dan normalisasi nilai piksel, agar sesuai dengan format input model. Selanjutnya, gambar diproses oleh model MobileNetV2 untuk mengekstraksi fitur visual penting dari citra roti tersebut. Hasil ekstraksi fitur ini kemudian dikirim ke lapisan Dense yang dilengkapi dengan fungsi aktivasi sigmoid guna melakukan klasifikasi biner, yaitu menentukan apakah roti termasuk kategori berjamur atau tidak berjamur.

#### 4. Optimasi

Model akan dilatih dengan parameter berikut,

1. Learning rate sebesar 0,0001
2. Jumlah epoch 30-50
3. Menggunakan dropout sebesar 0,5
4. Fungsi aktivasi convolutional layer pada penelitian ini menggunakan rectified linear unit (RELU) yang sudah tersedia di mobilenet
5. Optimizer yang digunakan adalah Adam
6. Loss function yang digunakan adalah Binary Crossentropy
7. Fungsi aktivasi pada layer terakhir pada penelitian ini menggunakan Sigmoid

#### 5. Evaluasi

Evaluasi model dilakukan untuk mengukur performa dalam mendeteksi keberadaan jamur pada roti, dengan menggunakan metrik evaluasi berupa akurasi, presisi, nilai kesalahan prediksi (recall), dan Skor-F1 [16]. Hasil evaluasi ini bertujuan untuk menunjukkan sejauh mana model mampu mengenali pola dasar yang terdapat pada data. Nilai-nilai tersebut diperoleh melalui rumus-rumus berikut:

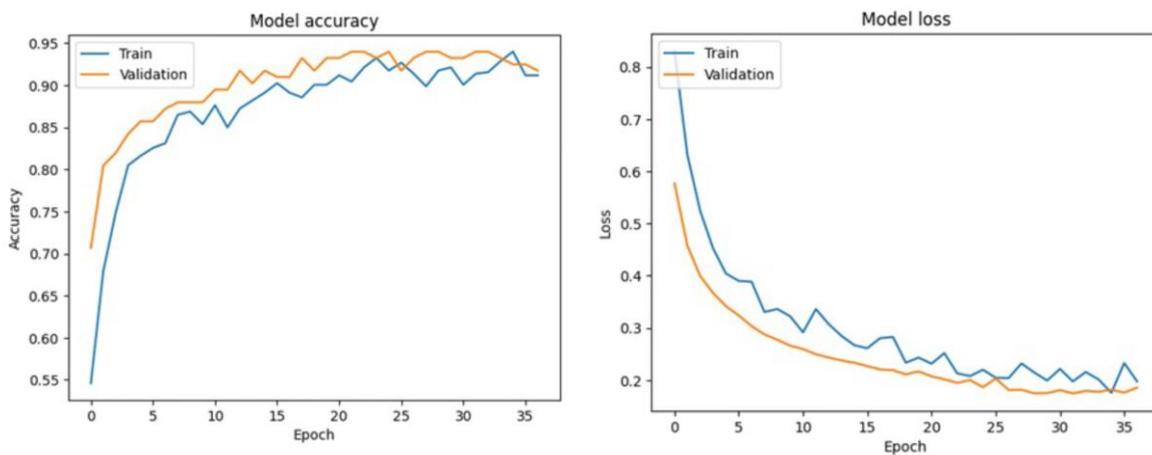
$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$presisi = \frac{TP}{TP+FP} \tag{4}$$

$$nilai\ kesalahan\ prediksi\ (recal) = \frac{TP}{TP+FN} \tag{5}$$

$$skor-f1 = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \tag{6}$$

### III. HASIL DAN PEMBAHASAN

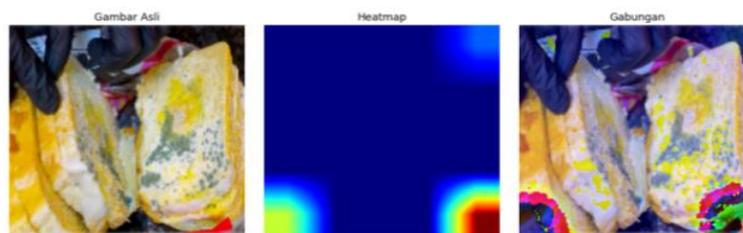


Gambar 4. Grafik Akurasi dan Loss (nilai kesalahan prediksi) Model

Peneliti menampilkan gambar grafik di atas untuk menunjukkan tren akurasi dan loss dari model MobileNetV2 selama proses pelatihan selama 37 epoch. Pada grafik akurasi (kiri), terlihat bahwa akurasi pelatihan meningkat stabil dari awal pelatihan yang dimulai di bawah 60%, dan terus naik hingga melampaui 90% di akhir epoch. Titik konvergensi model tampak terjadi sekitar epoch ke-25 hingga ke-30, di mana akurasi pelatihan dan validasi sudah tidak mengalami peningkatan signifikan dan mulai stabil. Akurasi validasi sendiri meningkat tajam dalam 5 epoch pertama, lalu mempertahankan konsistensi di kisaran 90%–94% sejak epoch ke-10. Hal ini menandakan bahwa model tidak hanya belajar dengan baik dari data pelatihan, tetapi juga memiliki kemampuan generalisasi yang kuat terhadap data baru. Tidak ada fluktuasi tajam yang menunjukkan overfitting, sehingga model dinilai stabil dan dapat dipercaya. Grafik loss (kanan) juga menunjukkan penurunan yang konsisten sepanjang pelatihan. Nilai loss awal data pelatihan mencapai lebih dari 0.8 dan terus menurun hingga berada di bawah 0.2 pada akhir epoch. Sementara itu, loss validasi bahkan lebih rendah dan lebih stabil dari loss pelatihan di hampir semua tahap, menunjukkan bahwa model tidak mengalami overfitting. Selain itu, tidak terlihat lonjakan drastis pada loss validasi, yang biasanya menjadi tanda bahwa model hanya menghafal data. Meskipun performa model cukup tinggi dengan akurasi uji mencapai 94%, dibandingkan penelitian sebelumnya seperti Khan et al. (2023) yang menggunakan citra mikroskopik dengan MobileNetV2 dan mencapai akurasi 95,67% hingga 98,52%, hasil penelitian ini sedikit lebih rendah. Namun, pendekatan pada studi ini menggunakan gambar dalam kondisi pencahayaan alami dan resolusi rendah, serta ditujukan untuk integrasi mobile, sehingga jauh lebih praktis dan realistis untuk penerapan di dunia nyata.

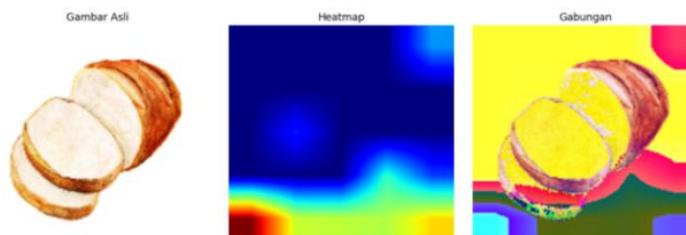
Kesalahan klasifikasi (misclassification) yang terjadi dapat dianalisis dari beberapa faktor. Misalnya, beberapa gambar roti dengan sudut pengambilan gambar tidak ideal, pencahayaan redup, atau warna jamur yang terlalu menyatu dengan warna roti menjadi sumber error. Gambar roti yang berwarna gelap di bagian pinggir kadang diklasifikasikan sebagai berjamur, padahal tidak. Sebaliknya, keberadaan jamur tipis yang hampir transparan tidak terdeteksi. Hal ini menunjukkan bahwa performa model masih bisa ditingkatkan dengan menambahkan variasi data pelatihan yang lebih kompleks dan realistis, serta mengombinasikan pemrosesan awal seperti kontras adaptif atau segmentasi sebelum klasifikasi. Studi ini juga belum membandingkan langsung hasil model MobileNetV2 dengan arsitektur CNN lain seperti VGG16, Inception, atau ResNet. Penelitian Treepong & Theera-Ampornpant (2023) menyebut bahwa MobileNetV2 memberikan efisiensi tinggi dengan parameter lebih ringan dibandingkan arsitektur-arsitektur tersebut. Oleh karena itu, meskipun secara akurasi sedikit di bawah, model ini tetap lebih unggul dalam hal kecepatan dan skalabilitas untuk digunakan dalam aplikasi Android.

Gradcam Berjamur CNN



(a)

GradCam Tidak Berjamur CNN



(b)

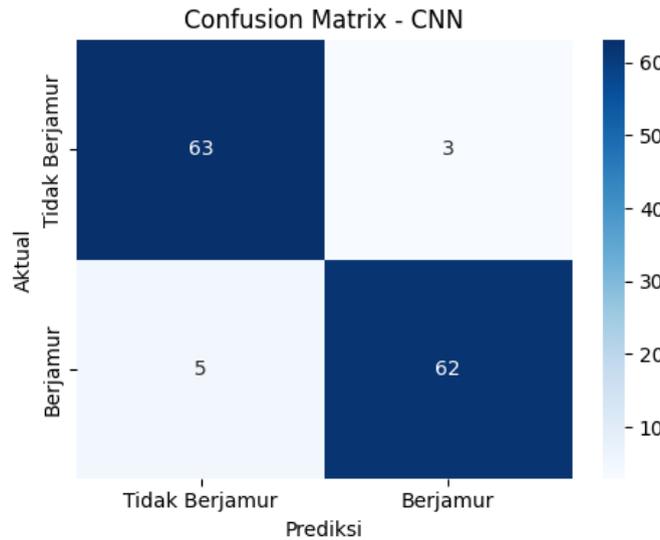
Gambar 5. (a) GradCam Roti Berjamur (b) Tidak Berjamur

Sebagaimana terlihat pada gambar di atas bahwa hasil visualisasi Grad-CAM pada dua sampel citra roti. Pada gambar pertama, area yang terindikasi mengandung jamur berhasil terdeteksi dengan baik, yang ditunjukkan oleh sebaran warna merah dan kuning pada heatmap serta citra gabungannya. Warna-warna ini menandakan area yang menjadi fokus perhatian model dalam melakukan klasifikasi sebagai roti berjamur. Sementara itu, pada gambar kedua yang merupakan sampel roti tanpa jamur, hasil visualisasi heatmap didominasi oleh warna biru, yang menunjukkan bahwa tidak ada area signifikan yang menjadi fokus model. Hal ini mengindikasikan bahwa model tidak mendeteksi pola visual yang mengarah pada keberadaan jamur pada gambar tersebut, sehingga klasifikasi dilakukan sebagai roti tidak berjamur.

Tabel 3. Evaluasi Metrik CNN

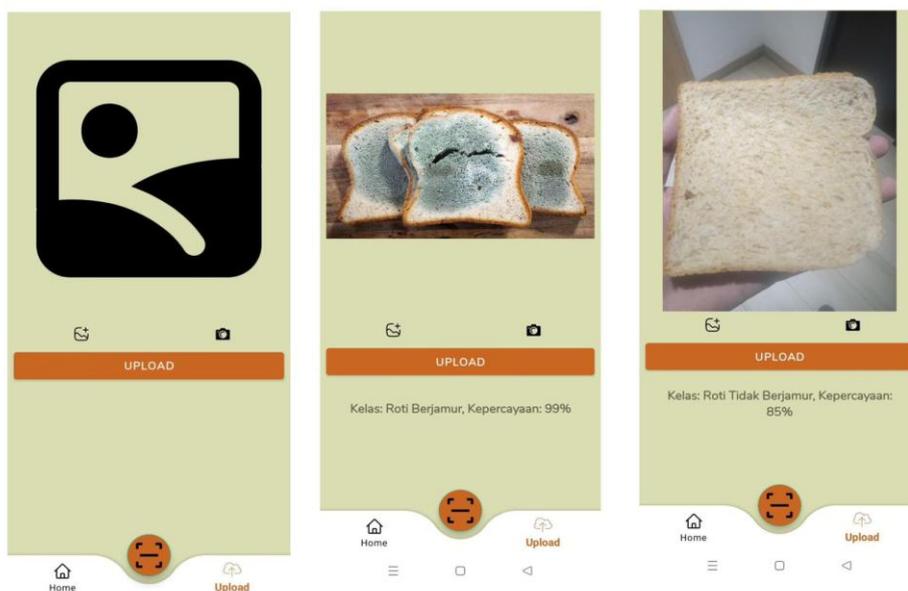
Laporan Klasifikasi	Presisi	Sensitivitas	Skor-F1	Jumlah Data
Tidak Berjamur	0.93	0.95	0.94	66
Berjamur	0.95	0.93	0.94	67
Akurasi	0.94	0.94	0.94	133
Rata-rata Makro	0.94	0.94	0.94	133
Rata-rata Tertimbang	0.94	0.94	0.94	133

Setelah proses pelatihan selesai, model dievaluasi menggunakan data uji yang sebelumnya tidak dikenali oleh model. Evaluasi dilakukan dengan menggunakan sejumlah metrik kinerja, yaitu akurasi, presisi, sensitivitas, dan Skor-F1. Berdasarkan hasil laporan klasifikasi, model menghasilkan nilai presisi, sensitivitas, dan Skor-F1 sebesar 0,94 untuk kedua kelas, yaitu “Berjamur” dan “Tidak Berjamur”. Akurasi keseluruhan yang dicapai oleh model adalah 94%, dari total 133 data uji.



Gambar 6. Confusion Matrix

Visualisasi confusion matrix di atas untuk menunjukkan kinerja model CNN dalam mengklasifikasikan gambar roti berjamur dan tidak berjamur. Model berhasil mengklasifikasikan 63 gambar tidak berjamur dan 62 gambar berjamur secara benar. Terdapat beberapa kesalahan klasifikasi, yaitu 3 gambar tidak berjamur yang diprediksi sebagai berjamur, dan 5 gambar berjamur yang diklasifikasikan sebagai tidak berjamur. Secara keseluruhan, hasil ini menunjukkan bahwa model memiliki akurasi yang tinggi dan mampu membedakan kedua kelas dengan cukup baik. Jumlah kesalahan yang relatif kecil memperlihatkan bahwa model telah belajar mengenali karakteristik visual dari masing-masing kelas secara efektif.



Gambar 7. Prototipe Antarmuka Aplikasi Untuk Mendeteksi Jamur Pada Roti

Gambar diatas memperlihatkan visualisasi antarmuka aplikasi yang dirancang dengan tampilan sederhana dan berfokus pada fungsi utama, sehingga memudahkan pengguna dalam mengunggah gambar roti dan melihat hasil klasifikasinya secara langsung. Pada halaman utama, tersedia pilihan untuk memilih gambar dari galeri atau mengambil foto langsung melalui kamera. Setelah gambar diproses, aplikasi akan menampilkan hasil klasifikasi berupa informasi apakah roti tersebut termasuk kategori Berjamur atau Tidak Berjamur, lengkap dengan persentase keyakinan (*confidence*) dari model terhadap prediksi tersebut. Desain yang minimalis dan ramah pengguna ini memungkinkan aplikasi digunakan oleh siapa saja, termasuk pengguna yang tidak memiliki latar belakang teknis. Dengan pendekatan ini, aplikasi siap dimanfaatkan sebagai alat deteksi jamur otomatis pada roti, baik untuk kebutuhan sehari-hari maupun untuk skala industri.

#### IV. KESIMPULAN

Penelitian ini berhasil mengembangkan prototipe aplikasi Android yang mampu mendeteksi keberadaan jamur pada roti secara otomatis dengan menggunakan model MobileNetV2. Dengan melatih model pada 666 gambar dan menjalankan pelatihan selama 37 epoch, model menunjukkan proses pembelajaran yang stabil tanpa indikasi overfitting. Hal ini membuktikan bahwa model tidak hanya memahami data pelatihan, tetapi juga mampu menggeneralisasi pada data baru secara efektif. Evaluasi akhir menunjukkan akurasi model mencapai 94%, dengan presisi, sensitivitas (*recall*), dan skor-F1 yang konsisten di angka 0,94 untuk masing-masing kelas “Berjamur” dan “Tidak Berjamur”. Nilai-nilai ini tidak hanya mencerminkan performa prediksi yang baik, tetapi juga menunjukkan keseimbangan klasifikasi antar kelas. Confusion matrix turut memperkuat temuan ini dengan menunjukkan jumlah prediksi yang benar cukup tinggi dan tingkat kesalahan klasifikasi yang rendah. Lebih dari sekadar pencapaian numerik, keberhasilan model ini memiliki implikasi penting bagi industri makanan, khususnya dalam hal peningkatan efisiensi inspeksi kualitas roti dan deteksi dini terhadap kontaminasi jamur secara otomatis. Aplikasi yang dikembangkan juga memiliki potensi besar untuk digunakan langsung oleh konsumen dalam kehidupan sehari-hari karena bersifat ringan dan praktis. Namun, penelitian ini memiliki keterbatasan, yaitu belum mengeksplorasi variasi jenis roti secara luas, pencahayaan ekstrem, maupun kondisi lingkungan yang lebih kompleks. Oleh karena itu, penelitian lanjutan disarankan untuk mengeksplorasi penggunaan metode segmentasi atau object detection seperti YOLO atau U-Net, agar sistem tidak hanya mengklasifikasikan gambar, tetapi juga mampu mengidentifikasi lokasi jamur secara presisi. Penelitian ini menunjukkan bahwa model klasifikasi berbasis MobileNetV2 tidak hanya akurat, tetapi juga efisien, sehingga sangat sesuai untuk diintegrasikan ke dalam aplikasi mobile yang bertujuan mendeteksi jamur pada produk roti secara otomatis. Sistem ini memiliki potensi besar untuk berkontribusi dalam peningkatan kontrol kualitas makanan, dengan memungkinkan pengguna melakukan deteksi jamur secara dini dari perangkat smartphone mereka. Untuk pengembangan di masa mendatang, prototipe ini dapat ditingkatkan dengan kemampuan deteksi waktu nyata (*real-time detection*) langsung melalui kamera perangkat, serta integrasi ke dalam perangkat keras tertanam seperti Raspberry Pi atau perangkat edge lainnya, guna memungkinkan penerapan langsung di industri roti.

## REFERENSI

- [1] N. Pacher *et al.*, “Ropiness in Bread A Re-Emerging Spoilage Phenomenon,” *Foods*, vol. 11, no. 19, pp. 1–16, 2022, doi: 10.3390/foods11193021.
- [2] N. Alshammari, “Mycotoxin source and its exposure causing mycotoxicoses,” *Bioinformation*, vol. 19, no. 4, pp. 348–357, 2023, doi: 10.6026/97320630019348.
- [3] D. D. Cordero-Mendoza, M. del C. A. Hernández-Ceruelos, S. Muñoz-Juárez, A. J. Izquierdo-Vega, I. Vega-Gaitan, and J. C. R. Ledezma, “Exposure to Mycotoxins and Its Importance in Public Health,” *Eur. J. Nutr. Food Saf.*, vol. 15, no. 10, pp. 21–28, 2023, doi: 10.9734/ejnfs/2023/v15i101342.
- [4] M. H. Fendiyanto and R. D. Satrio, “Identification, aflatoxin content, and antagonistic test of spoilage fungi in bread to *Aspergillus niger*,” *Bioteknologi*, vol. 17, no. 2, pp. 60–66, 2020, doi: 10.13057/biofar/c170202.
- [5] M. H. Boshra, G. S. El-Housseiny, M. M. S. Farag, and K. M. Aboshanab, “Innovative approaches for mycotoxin detection in various food categories,” *AMB Express*, vol. 14, no. 1, 2024, doi: 10.1186/s13568-024-01662-y.
- [6] C. Lin, M. A. A. Rahman, and P. G. Maropoulos, “Enhancing the quality inspection process in the food manufacturing industry through automation,” *2022 8th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2022*, pp. 379–384, 2022, doi: 10.1109/CoDIT55151.2022.9803939.
- [7] A. K. Pandey, M. K. Samota, A. Kumar, A. S. Silva, and N. K. Dubey, “Fungal mycotoxins in food commodities: present status and future concerns,” *Front. Sustain. Food Syst.*, vol. 7, no. May, pp. 1–21, 2023, doi: 10.3389/fsufs.2023.1162595.
- [8] M. Trigka and E. Dritsas, “A Comprehensive Survey of Deep Learning Approaches in Image Processing,” *Sensors*, vol. 25, no. 2, 2025, doi: 10.3390/s25020531.
- [9] S. Coleman, D. Kerr, and Y. Zhang, “Image Sensing and Processing with Convolutional Neural Networks,” *Sensors*, vol. 22, no. 10, pp. 10–12, 2022, doi: 10.3390/s22103612.
- [10] C. Y. Kim, K. S. Um, and S. W. Heo, “A novel MobileNet with selective depth multiplier to compromise complexity and accuracy,” *ETRI J.*, vol. 45, no. 4, pp. 666–677, 2023, doi: 10.4218/etrij.2022-0103.
- [11] R. Jain, S. K. Bekele, D. Palaniappan, K. Parmar, and T. Premavathi, “Employing Deep Convolutional Neural Networks for Enhanced Precision in Potato and Maize Leaf Disease Detection and Classification,” vol. 9, no. 2, pp. 290–301, 2025.
- [12] P. Treepong and N. Theera-Ampornpant, “Early bread mold detection through microscopic images using convolutional neural network: Early bread mold detection,” *Curr. Res. Food Sci.*, vol. 7, no. July, p. 100574, 2023, doi: 10.1016/j.crfs.2023.100574.
- [13] T. D. Nigat, T. M. Sitote, and B. M. Gedefaw, “Fungal Skin Disease Classification Using the Convolutional Neural Network,” *J. Healthc. Eng.*, vol. 2023, 2023, doi: 10.1155/2023/6370416.
- [14] F. Harby, A. Thaljaoui, D. Nayab, S. Aladhadh, S. E. L. Khediri, and R. U. Khan, “Data Augmentation and Random Multi-Model Deep Learning for Data Classification,” *Comput. Mater. Contin.*, vol. 74, no. 3, pp. 5191–5207, 2023, doi: 10.32604/cmc.2022.029420.
- [15] J. Lu, X. Liu, X. Ma, J. Tong, and J. Peng, “Improved MobileNetV2 crop disease identification model for intelligent agriculture,” *PeerJ Comput. Sci.*, vol. 9, 2023, doi: 10.7717/peerj-cs.1595.
- [16] Z. N. Ali, I. Askerzade, and S. Abdulwahab, “Estimation Model for Bread Quality Proficiency Using Fuzzy Weighted Relevance Vector Machine Classifier,” *Appl. Bionics Biomech.*, vol. 2021, 2021, doi: 10.1155/2021/6670316.