SENTRI:
Jurnal Riset Ilmiah
ejournal.nusantaraglobal.ac.id/index.php/sentri

Vol. 4, No. 12, 2025
e-ISSN: 2963-1130
pp. 4425-4445

Analisis Model *Convolutional Neural Network* Berbasis Fitur pada *Sobel Edge Detection* untuk Deteksi Tingkat Keausan Ban Mobil

Fahreza Syahrul Ramadhan[1*], Muhammad Zakariyah[1]
[1]Program Studi Informatika, Fakultas Sains & Teknologi, Universitas Teknologi Yogyakarta

*Corresponding Author's e-mail: reza.uteye407@gmail.com

Article History:
Received: November 24, 2025
Revised: December 16, 2025
Accepted: December 30, 2025

Keywords:
Tire, Tire Wear Detection, CNN, Edge Detection, Vehicle Safety.

Abstract: *Car tires play a crucial role in ensuring the safety of drivers and passengers. Tire wear that is not properly identified can reduce vehicle stability and traction, thereby increasing accident risk. Conventional tire wear assessment is generally carried out through manual visual inspection, which is subjective and prone to inconsistency, leading to potential misjudgment of actual wear conditions. This study proposes an automatic tire wear detection system using a Convolutional Neural Network (CNN) combined with an edge detection approach to provide a more objective and accurate assessment. The research stages include data preprocessing, dataset construction, and system evaluation. Model performance was tested using two data partitioning strategies, namely random and stratified classification, while maintaining the same architectural configuration consisting of HiddenL64, 10 epochs, maxpooling2, and iterationmax 20. The evaluation aimed to compare detection performance across both classification schemes. The experimental results indicate variations in training and testing accuracy between the two methods. The CNN model achieved its best performance under the random classification approach, with training accuracy reaching 99.15% and testing accuracy of 86.62% using a 90%–10% data split. Despite these results, the observed performance gap between training and testing data suggests that model generalization is still influenced by dataset size and distribution, indicating potential overfitting. Therefore, the findings should be interpreted within the scope of the experimental setup and dataset limitations. Overall, the proposed CNN-based system demonstrates promising capability for automatic tire wear classification and has potential to support preventive maintenance and timely tire replacement in automotive applications to improve road safety.*

Copyright © 2025, The Author(s).
This is an open access article under the CC–BY-SA license

How to cite: Ramadhan, F. S., & Zakariyah, M. (2025). Analisis Model Convolutional Neural Network Berbasis Fitur pada Sobel Edge Detection untuk Deteksi Tingkat Keausan Ban Mobil. *SENTRI: Jurnal Riset Ilmiah*, *4*(12), 4425–4445. https://doi.org/10.55681/sentri.v4i12.5190

PENDAHULUAN

Ban mobil merupakan elemen krusial dalam sistem kendaraan karena secara langsung menentukan tingkat keselamatan pengemudi dan penumpang. Kondisi ban yang mengalami degradasi, seperti keausan berlebih atau tekanan udara yang tidak sesuai, berpotensi memicu kegagalan fungsi, termasuk pecah ban dan hilangnya kendali kendaraan [1]. Faktor-faktor tersebut berkontribusi signifikan terhadap meningkatnya risiko kecelakaan lalu lintas. Data dari Komite Nasional Keselamatan Transportasi (KNKT) menunjukkan bahwa permasalahan ban menjadi salah satu penyebab dominan

kecelakaan di jalan tol Indonesia, sehingga menegaskan urgensi pemeriksaan kondisi ban secara berkala, meskipun praktik ini masih sering diabaikan oleh pemilik kendaraan.

Pendekatan konvensional dalam evaluasi kondisi ban umumnya mengandalkan inspeksi visual manual, yang sangat dipengaruhi oleh subjektivitas dan pengalaman pemeriksa. Ketergantungan ini menyebabkan hasil pemeriksaan cenderung tidak konsisten dan berpotensi kurang akurat. Kondisi tersebut mendorong perlunya pengembangan metode deteksi keausan ban yang bersifat objektif, terukur, dan terotomatisasi [2]. Sejumlah studi terdahulu mengindikasikan bahwa pemanfaatan pengolahan citra digital, khususnya melalui Convolutional Neural Network (CNN) yang dipadukan dengan teknik edge detection, mampu meningkatkan kemampuan sistem dalam mengenali karakteristik visual keausan ban secara lebih presisi [3].

Berdasarkan permasalahan tersebut, penelitian ini secara spesifik bertujuan untuk menganalisis pengaruh penerapan metode edge detection terhadap kinerja model CNN dalam mendeteksi tingkat keausan ban mobil. Fokus penelitian diarahkan pada evaluasi sejauh mana fitur tepi yang diekstraksi mampu meningkatkan akurasi klasifikasi CNN dibandingkan pendekatan tanpa penekanan fitur tepi [4]. Dengan demikian, penelitian ini tidak hanya mengembangkan sistem deteksi otomatis, tetapi juga menilai kontribusi edge detection sebagai tahap preprocessing terhadap performa model secara analitis.

Pemeriksaan keausan ban yang dilakukan secara rutin dan sistematis menjadi hal yang sangat penting untuk menghindari kecelakaan yang disebabkan oleh ban yang tidak layak pakai. Namun, banyak pengendara yang kurang memperhatikan aspek ini, sehingga potensi terjadinya kecelakaan meningkat. Teknologi yang dapat memberikan solusi otomatis dan lebih tepat dalam mendeteksi kondisi ban sangat dibutuhkan. Selain itu, dengan adanya solusi berbasis teknologi, pemilik kendaraan dapat dengan mudah memonitor kondisi ban secara lebih objektif dan dapat diandalkan tanpa harus bergantung pada pemeriksa manusia.

Hasil yang diharapkan dari penelitian ini adalah tersedianya model deteksi keausan ban yang memiliki kemampuan generalisasi lebih baik dan tingkat akurasi yang dapat dipertanggungjawabkan secara ilmiah. Implementasi sistem berbasis CNN dengan dukungan edge detection diharapkan mampu mendeteksi keausan secara lebih dini, termasuk pada kondisi yang sulit diamati secara visual. Dengan pendekatan ini, pemilik kendaraan dapat melakukan pemeliharaan secara preventif dan tepat waktu, sehingga berkontribusi pada peningkatan keselamatan berkendara dan penurunan risiko kecelakaan akibat kegagalan fungsi ban.

LANDASAN TEORI

Deep Learning

Deep learning merupakan sub-class dari *machine learning* yang menggunakan multi-layer atau banyaknya lapisan untuk mengekstraksi suatu fitur atau karakteristik dari data mentah secara progresif [5]. Kata "deep" dalam *deep learning* mengacu pada kedalaman arsitektur jaringan, di mana informasi dari data mentah diproses melalui beberapa lapisan untuk membentuk representasi abstrak yang makin kompleks. Setiap lapisan memiliki peran spesifik, misalnya dalam pemrosesan citra, lapisan awal dapat mendeteksi garis tepi, sedangkan lapisan berikutnya mampu mengenali pola bentuk hingga ke fitur kompleks seperti objek tertentu. Dengan pendekatan ini, *deep learning* sangat unggul dalam menangani data visual seperti gambar, suara, dan teks dalam jumlah besar.

Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) adalah salah satu metode paling populer dalam deep learning yang dirancang khusus untuk mengolah data citra digital. CNN sangat efektif digunakan dalam klasifikasi gambar, pengenalan objek, serta deteksi fitur visual secara otomatis. CNN bekerja melalui lapisan-lapisan konvolusi dan pooling yang membentuk representasi fitur secara bertahap. Dalam studi yang dilakukan oleh [6], CNN diterapkan untuk mendeteksi tulisan tangan angka Romawi dan berhasil mencapai akurasi tinggi melalui arsitektur yang terdiri dari beberapa lapisan konvolusi dan pooling yang saling berurutan dan terintegrasi.

Kelayakan Ban

Ban merupakan komponen kendaraan yang berperan sangat penting karena menjadi satu-satunya bagian yang bersentuhan langsung dengan permukaan jalan. Fungsi utama ban antara lain untuk meredam getaran jalan, menopang beban kendaraan, serta menjaga stabilitas dan kendali saat berkendara. Pemilihan ban yang tepat tidak hanya berkontribusi pada kenyamanan berkendara, tetapi juga secara signifikan mempengaruhi keselamatan pengendara, efisiensi operasional kendaraan, dan umur pemakaian ban yang lebih lama. Jika aspek seperti beban kendaraan, tekanan angin, dan jenis permukaan jalan tidak diperhatikan, ban akan lebih cepat mengalami keausan atau bahkan kerusakan [7].

Pengolahan Citra Digital

Pengolahan citra digital merupakan teknik pemrosesan gambar menggunakan komputer yang bertujuan untuk memanipulasi citra agar informasi yang terkandung di dalamnya dapat diolah secara optimal untuk keperluan analisis, pengenalan, dan klasifikasi [8]. Dalam praktiknya, pengolahan citra digunakan untuk berbagai aplikasi seperti pengenalan pola, deteksi objek, dan computer vision. Proses ini melibatkan beberapa tahap seperti konversi citra ke bentuk digital (grayscale atau biner), perbaikan kualitas citra, segmentasi, serta ekstraksi fitur penting dari objek visual. Jenis citra yang umum diolah antara lain citra warna (RGB), citra keabuan (grayscale), dan citra biner, yang masing-masing memiliki karakteristik intensitas berbeda. Citra digital terbentuk dari nilai-nilai diskrit yang memungkinkan komputer melakukan pemrosesan numerik secara efisien untuk mendeteksi pola tertentu dalam gambar.

Edge Detection

Edge detection adalah algoritma penting dalam pemrosesan citra dan visi komputer, yang digunakan untuk mengidentifikasi batas atau kontur objek dalam sebuah gambar [9]. Proses ini mendeteksi perubahan tajam pada intensitas piksel yang biasanya menandai tepi objek sehingga memungkinkan ekstraksi fitur penting seperti bentuk, kontur, dan struktur spasial. Teknik ini menjadi dasar dalam banyak aplikasi seperti segmentasi gambar, deteksi objek, dan analisis medis, serta terus berkembang dari metode klasik (Sobel, Canny, Prewitt) menuju teknik canggih berbasis *deep learning, fuzzy logic*, dan optimisasi.

Operator Sobel

Operator Sobel adalah metode deteksi tepi yang memakai dua filter konvolusi untuk menghitung gradien intensitas citra pada arah horizontal dan vertikal. Menurut

[10], operator ini menggunakan jendela 3x3 yang dikonvolusikan ke citra untuk memperkirakan gradien di pusat jendela.

Augmentasi Data

Augmentasi Data adalah salah satu metode penting dalam pengolahan citra untuk memperluas jumlah dan keragaman data pelatihan tanpa menambah data secara manual. Teknik ini melibatkan proses transformasi citra asli melalui rotasi, flipping, pergeseran posisi (*translasi*), zooming, serta penambahan noise, dengan tujuan agar model dapat belajar dari variasi citra yang lebih luas dan tidak terjebak pada pola tertentu [11]. Dalam penelitian oleh [10], augmentasi data digunakan secara ekstensif untuk meningkatkan performa model CNN dalam klasifikasi citra MRI otak. Proses augmentasi dilakukan secara in-place dan on-the-fly, di mana batch citra asli langsung diproses menggunakan teknik augmentasi sebelum dilatih ke dalam jaringan CNN.

Flowchart

Flowchart merupakan representasi visual dari langkah-langkah proses yang disusun secara sistematis untuk menggambarkan aliran kerja dari suatu sistem. Penggunaan flowchart dapat membantu pengguna dan pengembang memahami logika proses yang terjadi serta mengidentifikasi kelemahan atau kekurangan pada sistem. Seperti yang dijelaskan oleh [13], flowchart adalah teknik analitis bergambar yang digunakan untuk menjelaskan beberapa aspek dari sistem informasi secara jelas, ringkas, dan logis. Dalam konteks pengembangan sistem deteksi keausan ban berbasis CNN dan edge detection, flowchart digunakan untuk memetakan alur kerja aplikasi mulai dari input gambar, proses preprocessing, klasifikasi keausan, hingga penyajian hasil ke pengguna. Dengan adanya flowchart, pengembangan sistem dapat lebih terarah, terdokumentasi, dan mudah dievaluasi.

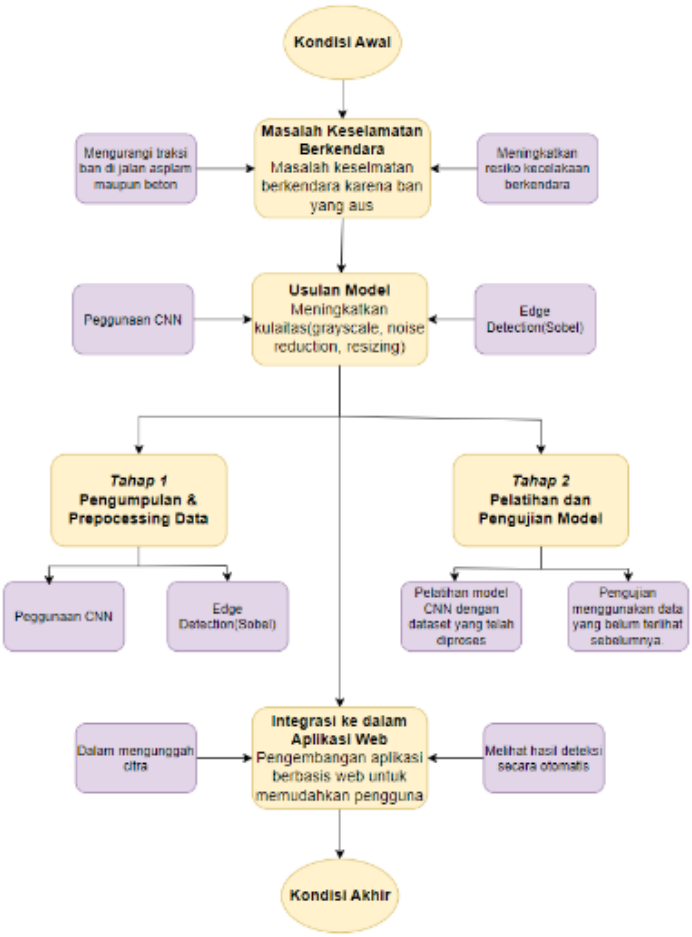
UML (Unified Modelling Language)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan elemen-elemen dalam sistem perangkat lunak. UML menjadi alat bantu yang penting dalam pengembangan sistem karena dapat menggambarkan struktur dan perilaku sistem secara menyeluruh. Dalam pengembangan sistem informasi, UML banyak digunakan untuk mendukung pendekatan pemrograman berorientasi objek, terutama dalam tahap analisis dan perancangan. Seperti dijelaskan oleh [14], UML mencakup dua jenis diagram utama, yaitu *Structural Diagram*, yang digunakan untuk menggambarkan hubungan antar kelas atau entitas dalam sistem, dan *Behavioral Diagram*, yang menggambarkan bagaimana aktor (pengguna) berinteraksi dengan sistem yang dibangun.

METODE PENELITIAN

Kerangka Penelitian

Penelitian akan berjalan dengan baik apabila terdapat tahapan atau *step-by-step* mengenai hal-hal yang perlu dilakukan untuk mendapatkan hasil yang sesuai harapan. Kerangka penelitian yang akan dilakukan dapat diihat dalam Gambar 1.



Gambar 1. Kerangka Penlitian

Data Penelitian

a. Sumber Data

Sumber data dalam penelitian ini berupa kumpulan gambar ban mobil dengan berbagai tingkat keausan yang diklasifikasikan ke dalam dua kategori utama: ban layak dipakai dan ban tidak layak dipakai. Dataset yang digunakan terdiri dari 700 gambar, dengan rincian 350 gambar (ban layak dipakai/safety) dan 350 gambar (ban tidak layak dipakai/*warning*).

Dataset ini diperoleh melalui beberapa metode, termasuk pengambilan gambar langsung di bengkel mobil. Penggunaan dataset yang beragam bertujuan untuk meningkatkan keakuratan model dalam mengenali pola keausan ban dari berbagai kondisi dan lingkungan.

Pada penelitian ini, gambar-gambar ban yang dikumpulkan diperhatikan aspek-aspek penting seperti kondisi pencahayaan, sudut pengambilan gambar, dan resolusi gambar agar dapat menangkap fitur-fitur keausan dengan lebih jelas. Beberapa gambar diperoleh dari lingkungan dengan pencahayaan alami, sementara yang lain diambil dalam kondisi pencahayaan buatan untuk meningkatkan generalisasi model.

Dataset yang digunakan nantinya akan menjalani tahap preprocessing, seperti grayscale conversion, noise reduction, resizing, dan edge detection menggunakan metode Sobel. Setelah preprocessing selesai, data akan dibagi menjadi dua bagian,

yaitu training set dan testing set, dengan proporsi 80% untuk pelatihan (training) dan 20% untuk pengujian (testing).

b. Cara mendapatkan data

Data dalam penelitian ini diperoleh dengan mengambil gambar ban secara langsung dari bengkel otomotif dan toko ban. Gambar-gambar tersebut diambil menggunakan kamera resolusi tinggi (48MP) untuk menangkap detail keausan dengan jelas, serta difoto dari berbagai sudut pandang untuk merekam pola keausan yang berbeda. Selain itu, pengambilan gambar dilakukan di berbagai kondisi pencahayaan, baik di dalam maupun di luar ruangan, untuk memastikan model dapat mengatasi variasi lingkungan.

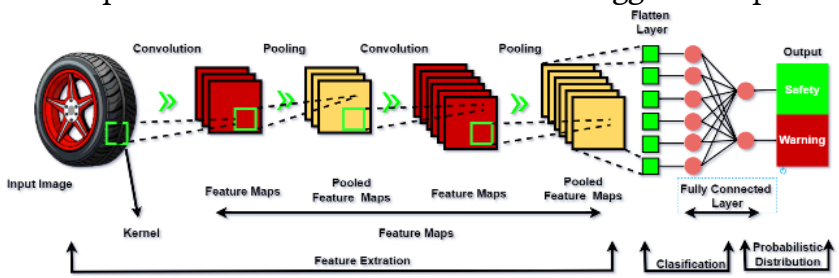
c. Waktu pengumpulan data

Pengumpulan data dilakukan dalam jangka waktu 2 bulan, dengan tahapan sebagai berikut:

1) Bulan Pertama (Maret-April), Pengumpulan dan Validasi Awal:
   a) Gambar ban dikumpulkan dari toko ban dan bengkel otomotif. Selama bulan pertama, sebanyak 500 gambar ban berhasil dikumpulkan.
   b) Setiap gambar diperiksa kualitasnya untuk memastikan resolusi cukup tinggi dan fitur keausan terlihat jelas.
   c) Klasifikasi awal dilakukan berdasarkan kondisi keausan, dengan kategori layak pakai dan tidak layak pakai.
2) Bulan Kedua (Mei-Juni), Preprocessing dan Pembagian Data:
   a) Pada bulan kedua, sebanyak 200 gambar ban tambahan berhasil diperoleh, yang meningkatkan variasi kondisi keausan ban yang tersedia.
   b) *Konversi grayscale* untuk mengurangi kompleksitas warna.
   c) *Noise reduction* agar gambar lebih bersih dan fitur tepi lebih mudah dikenali.
   d) R*esizing* gambar ke ukuran yang seragam agar kompatibel dengan model deep learning.
   e) *Edge detection* (Sobel) untuk menyoroti pola keausan pada permukaan ban.

Arsitektur Model

Arsitektur model pada penelitian *Deteksi Tingkat Keausan Ban Mobil Menggunakan Convolutional Neural Network berbasis Edge Detection* disusun untuk menggambarkan alur pemrosesan citra ban secara sistematis, sebagaimana ditunjukkan pada Gambar 2. Secara konseptual, sistem dimulai dari citra input berupa gambar ban mobil yang selanjutnya diproses melalui tahapan ekstraksi fitur dan klasifikasi menggunakan pendekatan CNN.



Gambar 2. Arsitektur Model

Pada tahap ekstraksi fitur, penelitian ini memanfaatkan arsitektur *pre-trained* InceptionV3 sebagai *feature extractor* utama. Penggunaan InceptionV3 bertujuan untuk

memperoleh representasi fitur visual yang lebih kaya dan stabil, mengingat arsitektur ini telah dilatih pada dataset berskala besar. Lapisan konvolusi pada InceptionV3 berfungsi untuk menangkap pola-pola kompleks pada permukaan ban, termasuk tekstur dan kontur keausan yang telah diperkuat melalui proses edge detection pada tahap praproses. Selama proses ini, operasi pooling digunakan untuk mereduksi dimensi fitur tanpa menghilangkan informasi penting, sehingga meningkatkan efisiensi komputasi dan ketahanan model terhadap variasi data. Selanjutnya, keluaran dari InceptionV3 diratakan (*flatten*) dan diteruskan ke lapisan *fully connected* yang berperan sebagai pengklasifikasi. Pada bagian ini, konfigurasi *hidden layer* dengan jumlah neuron terbatas (HiddenL64) digunakan sebagai lapisan klasifikasi akhir, sehingga istilah HiddenL64 merujuk pada struktur *dense layer* setelah proses ekstraksi fitur, bukan pada keseluruhan arsitektur CNN dari awal. Output model dibagi ke dalam dua kelas, yaitu "Safety" untuk ban yang masih layak digunakan dan "Warning" untuk ban dengan tingkat keausan yang signifikan.

Dengan penjelasan ini, dapat ditegaskan bahwa pendekatan yang digunakan merupakan kombinasi antara *transfer learning* menggunakan InceptionV3 sebagai ekstraktor fitur dan arsitektur CNN sederhana pada tahap klasifikasi. Penyelarasan terminologi ini menunjukkan bahwa model tidak sepenuhnya dibangun dari CNN konvensional sederhana, melainkan mengintegrasikan arsitektur *deep CNN* yang telah terlatih dengan lapisan klasifikasi yang lebih ringan untuk mendeteksi tingkat keausan ban secara otomatis dan terukur.

HASIL DAN PEMBAHASAN

Hasil Implementasi Sistem

Aplikasi yang dikembangkan memiliki dua jenis tampilan utama, yaitu untuk user dan admin. Pengguna *(user)* dapat mengunggah citra ban untuk dilakukan deteksi tingkat keausan dan melihat hasil klasifikasinya secara langsung, sedangkan admin memiliki akses tambahan untuk mengelola dataset, melakukan pelatihan model, serta memantau hasil evaluasi performa sistem secara keseluruhan. Tampilan dirancang sederhana dan responsif agar mudah digunakan oleh teknisi maupun pengguna umum.

User

a. Halaman Home User

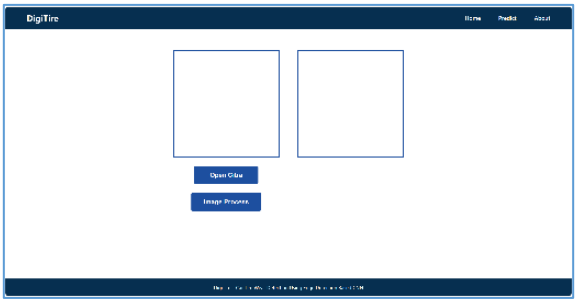
Gambar 3 menampilkan halaman utama aplikasi DigiTire berbasis web yang memperkenalkan fitur inti sistem, seperti akurasi deteksi tinggi, penerapan edge detection, antarmuka yang mudah digunakan, serta dukungan terhadap peningkatan keselamatan berkendara. Tampilan visual dirancang menarik dengan latar belakang citra ban dan ikon fitur utama, serta dilengkapi tombol View More untuk mengakses informasi lanjutan, sehingga memudahkan pengguna memahami fungsi aplikasi secara cepat dan jelas.



Gambar 3. Halaman *Home User*

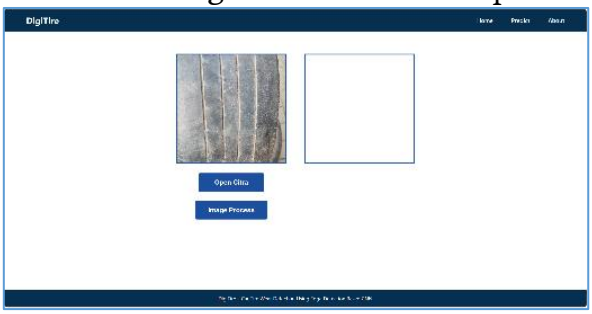
b. Halaman *Predict*

Gambar 4 menunjukkan halaman Predict yang memungkinkan pengguna mengunggah citra ban dan menjalankan proses deteksi keausan. Halaman ini menyediakan area tampilan untuk citra input dan hasil analisis, serta dua tombol utama, yaitu Open Citra untuk mengunggah gambar dan Image Process untuk memulai pemrosesan, sehingga proses analisis dapat dilakukan secara sederhana dan efisien.



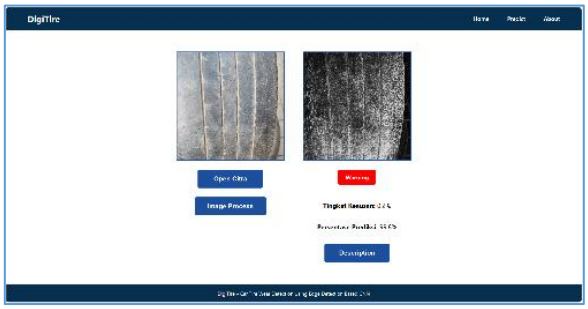
Gambar 4. Halaman Predict

Gambar 4 menampilkan antarmuka DigiTire setelah citra ban diunggah. Bagian kiri menampilkan gambar ban sebagai verifikasi input beserta tombol *Open Citra* dan *Image Process*, sedangkan bagian kanan disediakan untuk menampilkan hasil pemrosesan, termasuk deteksi tepi dan analisis tingkat keausan setelah proses dijalankan.



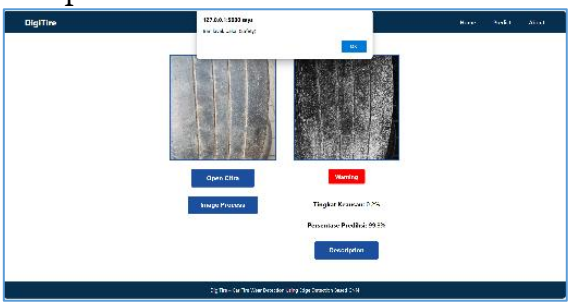
Gambar 5. Tampilan setelah input Citra Ban

Gambar 5 menunjukkan tampilan DigiTire setelah proses deteksi, di mana citra hasil pemrosesan ditampilkan berdampingan dengan gambar asli dan area keausan terlihat lebih jelas. Sistem menampilkan informasi tingkat keausan sebesar 0,2% dengan persentase prediksi 99,8%, disertai peringatan *Warning* sebagai indikator kondisi ban, serta tombol *Description* untuk melihat penjelasan hasil analisis.



Gambar 6. Tampilan setelah proses deteksi

Gambar 6 menampilkan *description* sebagai hasil akhir deteksi, yang menunjukkan citra asli dan citra hasil analisis disertai pop-up informasi kondisi ban. Sistem menyatakan status *"Ban layak pakai (Safety)"* berdasarkan tingkat keausan 0,2% dengan akurasi prediksi 99,8%, serta menyediakan tombol *OK* untuk menutup informasi dan melanjutkan penggunaan aplikasi.



Gambar 7. Tampilan description

c. Halaman About

Gambar 8 menampilkan halaman *About* yang berisi informasi mengenai tujuan dan fungsi aplikasi DigiTire dalam mendeteksi keausan ban menggunakan algoritma CNN dan edge detection. Halaman ini menjelaskan peran aplikasi dalam membantu pemantauan kondisi ban, memberikan indikasi tingkat keausan, serta mendukung pemeliharaan preventif guna meningkatkan keselamatan berkendara.



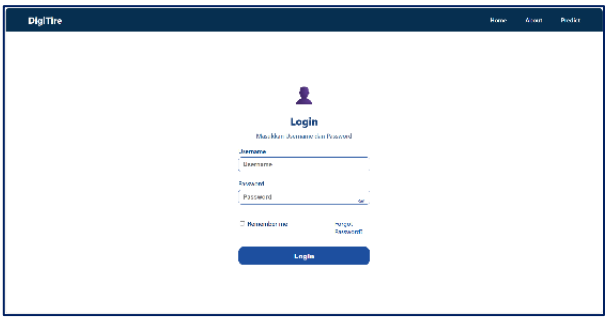
Gambar 8. Halaman About

Admin

a. Halaman Login

Gambar 9 menampilkan halaman *Login* aplikasi DigiTire yang menyediakan kolom *Username* dan *Password* untuk autentikasi pengguna, dilengkapi opsi *Remember me*,

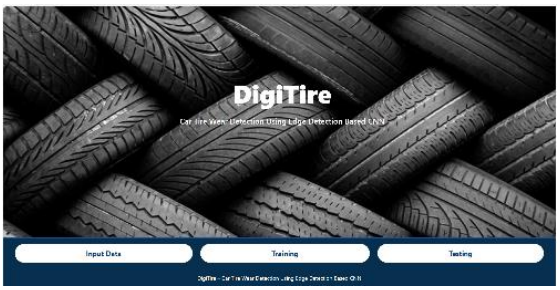
tautan *Forgot Password*, serta tombol *Login* untuk mengakses sistem sesuai kredensial yang dimasukkan.



Gambar 9. Halaman Login

b. Halaman Home Admin

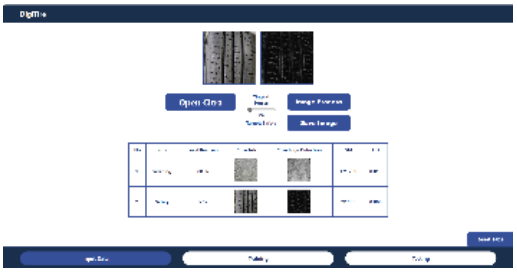
Gambar 10 menampilkan halaman *Home Admin* DigiTire yang memuat logo dan judul aplikasi, serta tiga menu utama, yaitu *Input Data*, *Training*, dan *Testing*, yang digunakan admin untuk mengelola data, melatih model, dan melakukan pengujian sistem.



Gambar 10. Halaman Home Admin

c. Halaman Input Data

Gambar 11 menampilkan halaman *Input Data* pada DigiTire yang digunakan admin untuk mengunggah dan memproses citra ban. Halaman ini menyediakan area tampilan citra asli dan hasil deteksi tepi, tombol *Open Citra*, *Image Process*, dan *Save Image*, serta tabel informasi hasil pemrosesan dan opsi *Reset Tabel* untuk mengelola data.



Gambar 11. Halaman Input Data

d. Halaman Training

Gambar 12 menampilkan halaman *Training* DigiTire yang digunakan untuk melatih model deteksi keausan ban. Halaman ini menyediakan pengaturan rasio data latih, parameter model seperti *hidden layer*, *epoch*, *MaxPooling*, dan *Iterasi Max*, serta tombol

Process dan *Save*. Tabel di bagian bawah menampilkan konfigurasi dan data pelatihan yang digunakan untuk proses pembelajaran model.



Gambar 12. Halaman Training

Gambar 12 menampilkan hasil pelatihan model yang terdiri atas tabel ringkasan parameter dan data pelatihan serta grafik akurasi. Tabel menunjukkan konfigurasi dan hasil tiap epoch, sedangkan grafik memperlihatkan perbandingan akurasi pelatihan dan validasi sebagai indikator perkembangan kinerja model selama proses training.



Gambar 13. Tampilan setelah hasil Train

e. Halaman Testing

Gambar 14 menampilkan halaman *Testing* DigiTire yang digunakan untuk mengevaluasi kinerja model deteksi keausan ban. Halaman ini menyediakan pengaturan rasio data uji, informasi jumlah data *Safety* dan *Warning*, tabel parameter pengujian, serta tombol *Test* dan *Save* untuk menjalankan dan menyimpan hasil evaluasi model.



Gambar 14. Halaman Testing

Gambar 15 menampilkan ringkasan hasil pengujian model dengan jumlah data uji 64 *Safety* dan 93 *Warning*. Tabel menunjukkan metrik evaluasi seperti akurasi 82,80% dan loss 0,5329 beserta parameter pengujian, serta dilengkapi tombol *Test* dan *Save* untuk menjalankan dan menyimpan hasil pengujian.



Gambar 15. Tampilan setelah hasil Test

Hasil Pemodelan

Bagian ini menyajikan implementasi source code metode Sobel dan CNN yang digunakan untuk mendeteksi tepi serta mengekstraksi fitur keausan ban, kemudian mengklasifikasikan kondisinya. Output dari setiap tahap dianalisis untuk menilai efektivitas model dalam mendeteksi tingkat keausan ban secara otomatis.

a. *Edge Detection* (Sobel)

Gambar 16 menampilkan implementasi *edge detection* menggunakan metode Sobel melalui perhitungan *gradient magnitude* dan *Gaussian derivative*. Proses diawali dengan konversi citra ke grayscale, dilanjutkan penerapan filter Sobel pada arah horizontal dan vertikal untuk mengekstraksi tepi, serta *Gaussian blur* untuk mereduksi noise. Hasilnya ditampilkan dalam bentuk visualisasi citra asli, deteksi Sobel, derivatif Gaussian, dan magnitudo gradien, disertai statistik nilai minimum, maksimum, dan rata-rata sebagai indikator karakteristik tepi citra.

```
# Function for Sobel edge detection, gradient method, and Gaussian Derivative
def apply_sobel_edge_detection(image):
    # Convert image to grayscale
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    # Apply Sobel Edge Detection (Gradient Method)
    sobel_x = cv2.Sobel(gray_image, cv2.CV_64F, 1, 0, ksize=3) # Sobel in X direction
    sobel_y = cv2.Sobel(gray_image, cv2.CV_64F, 0, 1, ksize=3) # Sobel in Y direction

    # Calculate gradient magnitude
    gradient_magnitude = cv2.magnitude(sobel_x, sobel_y)

    return gray_image, sobel_x, sobel_y, gradient_magnitude

# Function to calculate Gaussian Derivative
def apply_gaussian_derivative(image):
    # Convert image to grayscale
```

```
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    # Apply Gaussian blur to the image
    gaussian_blurred = cv2.GaussianBlur(gray_image, (5, 5), 1.5)

    # Calculate X and Y derivatives of the Gaussian
        sobel_gaussian_x = cv2.Sobel(gaussian_blurred, cv2.CV_64F, 1, 0, ksize=5)
        sobel_gaussian_y = cv2.Sobel(gaussian_blurred, cv2.CV_64F, 0, 1, ksize=5)

    # Calculate the magnitude of the Sobel Gaussian derivatives
        gaussian_derivative_magnitude = cv2.magnitude(sobel_gaussian_x, sobel_gaussian_y)

    return gaussian_derivative_magnitude

# Assuming `sample_images[0]` is the image from the dataset (use the first image from the batch)
sample_images, sample_labels = next(train_generator)

# Apply Sobel edge detection and calculate gradients and Gaussian derivatives on the first image in the batch
original_image, sobel_x, sobel_y, gradient_magnitude = apply_sobel_edge_detection(sample_images[0])

# Apply Gaussian Derivative
gaussian_derivative = apply_gaussian_derivative(sample_images[0])

# Display the images in a 2x2 grid
plt.figure(figsize=(15, 10))

# Original Image
plt.subplot(2, 2, 1)
plt.imshow(sample_images[0])
plt.title('Original Image')
plt.axis('off')

# Sobel Edge Detection (Gradient Magnitude)
plt.subplot(2, 2, 2)
plt.imshow(gradient_magnitude, cmap='gray')
plt.title('Sobel Edge Detection')
plt.axis('off')

# Gaussian Derivative
plt.subplot(2, 2, 3)
plt.imshow(gaussian_derivative, cmap='gray')
```

```
plt.title('Gaussian Derivative')
plt.axis('off')

# Gradient Magnitude (This is already the result from Sobel Edge Detection)
plt.subplot(2, 2, 4)
plt.imshow(gradient_magnitude, cmap='gray')
plt.title('Gradient Magnitude')
plt.axis('off')

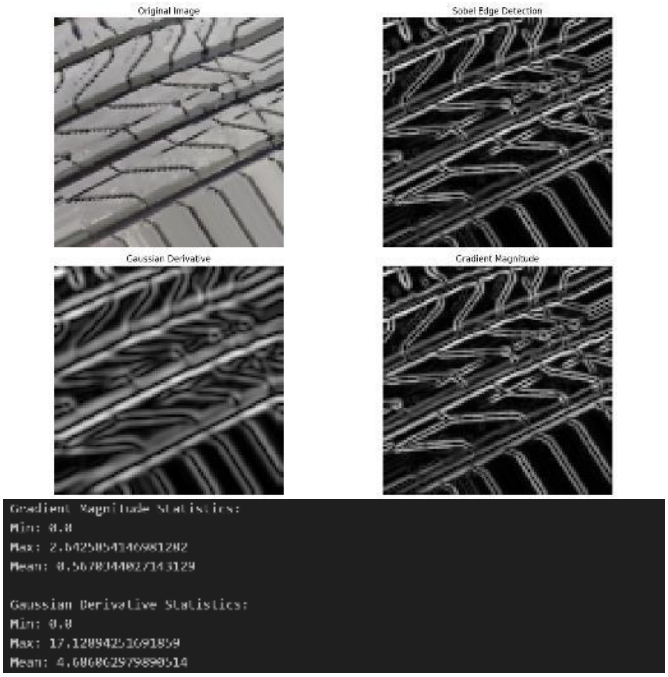
plt.tight_layout()
plt.show()

# Print the numerical values for gradient magnitude and Gaussian derivative
print("Gradient Magnitude Statistics:")
print("Min:", np.min(gradient_magnitude))
print("Max:", np.max(gradient_magnitude))
print("Mean:", np.mean(gradient_magnitude))

print("\nGaussian Derivative Statistics:")
print("Min:", np.min(gaussian_derivative))
print("Max:", np.max(gaussian_derivative))
print("Mean:", np.mean(gaussian_derivative))
```

Gambar 16. Source Code Edge Detection (Sobel)

Pada Gambar 17 output dari kode *Edge Detection (*Sobel*)* menunjukkan hasil dari beberapa teknik pemrosesan citra untuk mendeteksi tepi pada gambar ban. Gambar pertama adalah citra asli ban, yang memberikan gambaran umum tentang permukaan ban. Gambar kedua, hasil dari deteksi tepi Sobel, memperjelas garis-garis tepi yang ada pada permukaan ban, menonjolkan kontur ban yang lebih tajam. Gambar ketiga menunjukkan hasil turunan Gaussian, yang bertujuan untuk mereduksi noise dan meningkatkan kejelasan tepi dengan lebih halus. Gambar terakhir menampilkan magnitudo gradien, yang mengukur intensitas perubahan pada permukaan ban, dengan nilai-nilai statistik seperti nilai minimum, maksimum, dan rata-rata yang memberikan wawasan tentang seberapa besar perubahan intensitas di setiap area gambar, sehingga memungkinkan analisis lebih lanjut terhadap kondisi keausan ban.



Gambar 17. Output Code Edge Detection (Sobel)

b. Convolutional Neural Network (CNN)

Gambar 18 menunjukkan source code untuk membangun arsitektur *Convolutional Neural Network* (CNN) menggunakan model InceptionV3 sebagai base model. Pada kode ini, model InceptionV3 digunakan dengan input ukuran gambar 100x100 piksel dan tiga kanal warna (RGB). Model ini dimuat tanpa lapisan atas (include_top=False), memungkinkan untuk menyesuaikan lapisan klasifikasi akhir sesuai dengan kebutuhan. Semua lapisan dalam base model diset tidak dapat dilatih (layer.trainable = False) untuk memanfaatkan bobot yang sudah dilatih sebelumnya dengan ImageNet. Selanjutnya, lapisan-lapisan tambahan ditambahkan: lapisan Flatten untuk meratakan output, diikuti oleh beberapa lapisan Dense dengan fungsi aktivasi *ReLU* dan *Dropout* untuk mengurangi overfitting. Lapisan akhir menggunakan fungsi aktivasi sigmoid untuk menghasilkan output biner (safety atau warning). Model kemudian dikompilasi menggunakan optimizer Adam dengan tingkat pembelajaran 0.001, menggunakan binary_crossentropy sebagai fungsi kerugian dan akurasi sebagai metrik. model.summary() digunakan untuk menampilkan ringkasan arsitektur model yang telah dibangun.

```
# Setup CNN Architecture
nb_classes = 2
lr = 0.001

base_model = InceptionV3(input_shape=(100, 100, 3), include_top=False, weights='imagenet')

for layer in base_model.layers:
    layer.trainable = False
```

```
        x = Flatten()(base_model.output)
        x = Dropout(0.5)(x)
        x = Dense(1024, activation='relu')(x)
        x = Dense(512, activation='relu')(x)
        x = Dense(256, activation='relu')(x)
        x = Dense(128, activation='relu')(x)
        x = Dropout(0.5)(x)

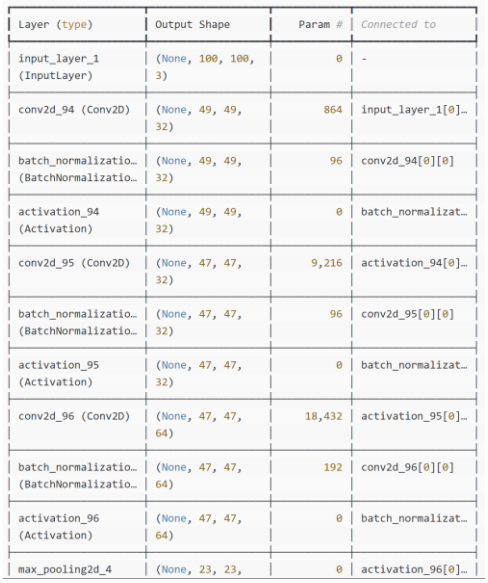
        # Add a final sigmoid layer with 1 node for classification output
        x = Dense(1, activation='sigmoid')(x)

        model = tf.keras.models.Model(base_model.input, x)

        model.compile(optimizer=Adam(learning_rate=lr),
loss='binary_crossentropy', metrics=['acc'])
        model.summary()
```

Gambar 18. *Source Code Convolutional Neural Network* (CNN)

Gambar 19 menunjukkan output dari arsitektur *Convolutional Neural Network* (CNN) yang terdiri dari beberapa lapisan yang mengonvolusi gambar dan mengeluarkan fitur. Pada bagian awal dari output ini, tampak lapisan pertama yaitu *input_layer_1* yang memiliki bentuk output (None, 100, 100, 3), menunjukkan bahwa gambar yang dimasukkan berukuran 100x100 piksel dengan tiga kanal warna (RGB). Selanjutnya, lapisan *conv2d_94* melakukan konvolusi 2D dengan filter sebanyak 32, menghasilkan output berukuran (None, 49, 49, 32), yang berarti ukuran gambar berkurang karena proses konvolusi. Setelah itu, dilakukan normalisasi batch dengan lapisan *batch_normalization_94*, diikuti dengan lapisan aktivasi *activation_94* yang tidak mengubah ukuran output, tetapi memperkenalkan non-linearitas. Proses serupa terjadi di lapisan *conv2d_95*, *batch_normalization_95*, *activation_95*, dan seterusnya, dengan jumlah filter yang meningkat dan ukuran output gambar yang tetap berkurang di setiap lapisan konvolusi, hingga mencapai lapisan *max_pooling2d_4* yang mengurangi ukuran gambar lebih lanjut menjadi (None, 23, 23, 23). Setiap lapisan dilengkapi dengan parameter yang dihitung (misalnya, jumlah parameter pelatihan) yang memungkinkan model untuk belajar representasi gambar lebih dalam. Output ini menunjukkan tahapan dari ekstraksi fitur hingga penurunan dimensi gambar yang digunakan untuk klasifikasi pada tahap selanjutnya dalam CNN.



Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 100, 100, 3)	0	-
conv2d_94 (Conv2D)	(None, 49, 49, 32)	864	input_layer_1[0]…
batch_normalizatio… (BatchNormalizatio…	(None, 49, 49, 32)	96	conv2d_94[0][0]
activation_94 (Activation)	(None, 49, 49, 32)	0	batch_normalizat…
conv2d_95 (Conv2D)	(None, 47, 47, 32)	9,216	activation_94[0]…
batch_normalizatio… (BatchNormalizatio…	(None, 47, 47, 32)	96	conv2d_95[0][0]
activation_95 (Activation)	(None, 47, 47, 32)	0	batch_normalizat…
conv2d_96 (Conv2D)	(None, 47, 47, 64)	18,432	activation_95[0]…
batch_normalizatio… (BatchNormalizatio…	(None, 47, 47, 64)	192	conv2d_96[0][0]
activation_96 (Activation)	(None, 47, 47, 64)	0	batch_normalizat…
max_pooling2d_4	(None, 23, 23,	0	activation_96[0]…

Gambar 19. *Output Code Convolutional Neural Network* (CNN)

Hasil Pengujian

Pengujian Sistem (*Blackbox*)

Pengujian *black box* digunakan untuk melakukan pengujian terhadap sistem guna mengetahui apakah software yang dibuat sesuai dengan yang diharapkan atau tidak. Tujuan pengujian black box adalah untuk menemukan kesalahan fungsi pada program. Pengujian menggunakan metode black box hanya mengamati hasil fungsional dari software. Tabel pengujian black box dapat dilihat pada tabel 1. Berikut.

Tabel 1. Pengujian *Blackbox*

No.	Pengujian	Keterangan	Kesimpulan
1.	Halaman Utama	Menampilkan logo aplikasi, gambar, dan tombol "View More" untuk memperkenalkan DigiTire	Berhasil
2.	Halaman Prediksi	Mengarahkan pengguna ke halaman prediksi dengan fitur tombol "Open Citra" dan "Image Process"	Berhasil
3.	Halaman Input Citra Ban	Menampilkan citra ban yang telah diunggah untuk diproses pada halaman prediksi	Berhasil
4.	Halaman Proses Deteksi	Menampilkan hasil deteksi keausan pada ban lengkap dengan informasi tingkat keausan dan persentase prediksi	Berhasil
5.	Halaman Description	Menampilkan pop-up peringatan dengan status kelayakan ban yang mudah dipahami	Berhasil
6.	Halaman About	Memberikan penjelasan mengenai DigiTire, teknologi yang digunakan, dan tujuan pengembangan sistem	Berhasil
7.	Halaman Login	Menyediakan form untuk login dengan kolom untuk username dan password serta tombol untuk masuk	Berhasil
8.	Halaman	Menampilkan menu utama admin dengan tombol	Berhasil

	Home Admin	untuk mengakses fitur Input Data, Training, dan Testing	
9.	Halaman Input Data	Halaman untuk memasukkan data dengan form dan tombol untuk memproses citra ban	Berhasil
10.	Halaman Training	Menampilkan halaman untuk mempersiapkan data pelatihan, dengan pengaturan parameter dan tabel informasi	Berhasil
11.	Tampilan Setelah Hasil Train	Menampilkan hasil pelatihan data dengan akurasi per epoch dan grafik progres training	Berhasil
12.	Halaman Testing	Menampilkan tampilan untuk memulai pengujian model dengan data yang tersedia dan tombol "Test" untuk memulai proses	Berhasil
13.	Tampilan Setelah Hasil Test	Menampilkan hasil pengujian model dengan informasi akurasi, loss, dan jumlah data yang diuji	Berhasil

Pengujian Model (Matrik Evaluasi/ *Confusion Matrix*)

Pada pengujian model, dilakukan pengujian dengan dua metode klasifikasi, yaitu random dan stratified, menggunakan arsitektur model yang sama, yakni HiddenL64, epoch 10, maxpooling2, dan iterationmax 20. Pengujian ini bertujuan untuk membandingkan performa model dalam mendeteksi tingkat keausan ban pada kedua jenis klasifikasi tersebut. Berdasarkan pengujian ini, dihasilkan akurasi yang berbeda pada data latih dan data uji untuk kedua metode klasifikasi, yang dapat dilihat pada tabel berikut ini.

a. Random

Tabel 2. Tabel Perbandingan Akurasi Model Klasifikasi Random

No.	Presentase Data Latih	Presentase Data Uji	Akurasi Data Latih	Akurasi Data Uji
1.	90	10	99,15%	86,62%
2.	80	20	99.09%	84,71%
3.	70	30	69,33%	68,79%
4.	60	40	95,97%	72,61%
5.	50	50	77,29%	67,40%

b. Stratified

Tabel 3. Tabel Perbandingan Akurasi Model Klasifikasi Stratified

No.	Presentase Data Latih	Presentase Data Uji	Akurasi Data Latih	Akurasi Data Uji
1.	90	10	96,24%	77,70%
2.	80	20	98.49%	84,71%
3.	70	30	69,24%	77,70%
4.	60	40	98,79%	64,23%
5.	50	50	85,99%	65,50%

Pembahasan

Pembahasan hasil penelitian menunjukkan bahwa sistem yang dikembangkan telah mampu mengintegrasikan antarmuka pengguna dan admin secara fungsional untuk mendukung proses deteksi keausan ban, mulai dari input citra, pemrosesan, hingga evaluasi model. Dari sisi implementasi, alur kerja aplikasi berjalan sesuai rancangan dan memungkinkan pengguna maupun admin berinteraksi dengan sistem secara efisien. Namun demikian, aspek ini lebih merefleksikan keberhasilan implementatif daripada kontribusi ilmiah utama penelitian.

Dari hasil pengujian model, perbedaan kinerja antara metode pembagian data *random* dan *stratified* mengindikasikan bahwa performa CNN sangat dipengaruhi oleh strategi pembagian dataset. Akurasi pelatihan yang sangat tinggi pada beberapa skenario, terutama pada metode *random* dengan rasio 90:10, diikuti oleh penurunan akurasi pada data uji, menunjukkan adanya tantangan generalisasi model. Temuan ini menguatkan indikasi terjadinya overfitting, yang kemungkinan dipicu oleh ukuran dataset yang relatif terbatas serta variasi data yang belum sepenuhnya merepresentasikan kondisi keausan ban di lapangan.

Selain itu, meskipun Sobel edge detection digunakan sebagai tahap praproses untuk menonjolkan fitur tepi, kontribusinya terhadap peningkatan performa CNN belum dianalisis secara kuantitatif. Penelitian ini belum membandingkan kinerja model dengan dan tanpa penerapan edge detection, sehingga pengaruh langsung Sobel terhadap akurasi klasifikasi belum dapat disimpulkan secara kuat. Kondisi ini menjadikan klaim keunggulan pendekatan berbasis fitur tepi masih bersifat indikatif dan memerlukan pengujian lanjutan yang lebih terkontrol.

Secara ilmiah, hasil penelitian ini menunjukkan potensi penggunaan CNN untuk deteksi keausan ban secara otomatis, namun juga menegaskan pentingnya evaluasi yang lebih mendalam terkait generalisasi model dan efektivitas setiap tahap praproses. Ke depan, penelitian lanjutan perlu melibatkan dataset yang lebih besar dan beragam, serta analisis komparatif yang jelas untuk mengukur kontribusi Sobel edge detection terhadap peningkatan kinerja CNN secara signifikan dan terukur.

KESIMPULAN

Kesimpulan penelitian ini menegaskan beberapa kontribusi penting, baik secara metodologis maupun konseptual, adalah:

a. Secara metodologis, penelitian ini menunjukkan bahwa integrasi *edge detection* menggunakan operator Sobel sebagai tahap praproses dengan model Convolutional Neural Network (CNN) mampu membentuk alur deteksi keausan ban yang lebih terstruktur dan objektif. Pendekatan ini memperkaya proses ekstraksi fitur visual dengan menonjolkan karakteristik tepi permukaan ban, sehingga memberikan dasar metodologis bagi pengembangan sistem deteksi keausan berbasis citra yang lebih sistematis dibandingkan inspeksi manual yang subjektif..
b. Secara empiris, hasil pengujian mengindikasikan bahwa strategi pembagian data berpengaruh signifikan terhadap kinerja model CNN. Metode *random split* dengan rasio 90% data latih dan 10% data uji menghasilkan kinerja terbaik, yang menunjukkan bahwa konfigurasi pelatihan tertentu dapat mengoptimalkan kemampuan model dalam mempelajari pola keausan ban. Temuan ini memberikan kontribusi metodologis berupa bukti bahwa pemilihan skema pembagian data

merupakan faktor krusial dalam pengembangan model klasifikasi berbasis *deep learning*.

c. Secara teoretis, penelitian ini memperkuat pemahaman bahwa pemanfaatan fitur visual tingkat rendah, seperti tepi citra, tetap relevan dalam konteks *deep learning*, khususnya ketika diterapkan pada domain dengan karakteristik tekstur yang kuat seperti permukaan ban. Meskipun CNN memiliki kemampuan ekstraksi fitur otomatis, hasil penelitian ini menunjukkan bahwa tahap praproses berbasis fitur dapat berperan sebagai pendukung pembelajaran representasi yang lebih efektif. Dengan demikian, penelitian ini memberikan kontribusi awal terhadap diskursus mengenai peran *feature-based preprocessing* dalam meningkatkan kinerja CNN pada aplikasi deteksi kondisi objek.

DAFTAR REFERENSI

[1] K. Prayoga, R. Magdalena, And S. Saidah, "Sistem Deteksi Kecacatan Ban Dengan Convolutional Neural Network," *E-Proceeding Of Engineering*, Vol. 10, No. 3, Pp. 2229–2234, 2023.

[2] L. Listyalina, I. Buyung, A. Q. Munir, I. Mustiadi, And D. A. Dharmawan, "Conv-Tire: Tire Feasibility Assessment Using Convolutional Neural Networks Conv-Tire: Asesmen Kelayakan Ban Berbasis Convolutional Neural Network," *Jurnal Informatika Dan Teknologi Informasi*, Vol. 19, No. 3, Pp. 323–336, 2022, Doi: 10.31515/Telematika.V19i3.7697.

[3] Nugroho Febriyanto, Cahya Rahmad, And C. B. Vista, "Deteksi Kerusakan Dan Kelayakan Ban Mobil Berdasarkan Ekstraksi Tekstur Menggunakan Gray Level Cooccurrence Matrix Dengan Metode Jaringan Syaraf Tiruan," *Jurnal Informatika Polinema*, Vol. 7, No. 4, Pp. 27–32, 2021, Doi: 10.33795/Jip.V7i4.469.

[4] L. Listyalina, I. Buyung, A. Q. Munir, I. Mustiadi, And D. A. Dharmawan, "Conv-Tire: Tire Feasibility Assessment Using Convolutional Neural Networks Conv-Tire: Asesmen Kelayakan Ban Berbasis Convolutional Neural Network," *Jurnal Informatika Dan Teknologi Informasi*, Vol. 19, No. 3, Pp. 323–336, 2022, Doi: 10.31515/Telematika.V19i3.7697.

[5] L. Utari And A. Zulfikar, "Penerapan Convolutional Neural Networks Menggunakan Edge Detection Untuk Identifikasi Motif Jenis Batik," *Teknois : Jurnal Ilmiah Teknologi Informasi Dan Sains*, Vol. 13, No. 1, Pp. 110–123, 2023, Doi: 10.36350/Jbs.V13i1.184.

[6] M. Toyib, T. Decky, And K. Pratama, "Penerapan Algoritma Cnn Untuk Mendeteksi Tulisan Tangan Angka Romawi Dengan Augmentasi Data," *Urnal Matematika, Ilmu Pengetahuan Alam, Kebumian Dan Angkasa*, Vol. 2, No. 3, Pp. 108–120, 2024.

[7] A. Urfiandi, "Analisis Tingkat Keausan Terhadap Pemakaian Ban Merek A, B Dan C Menggunakan Ban Standar 90/90-14 46 P," *Surya Teknika*, Vol. 8, No. 1, Pp. 282–288, 2018.

[8] A. L. A. Shidiq, E. Suhartono, And S. Saidah, "Klasifikasi Kecacatan Ban Untuk Mengendalikan Kualitas Produk Menggunakan Model Cnn Dengan Arsitektur Vgg-16 Classification Of Tire Defect To Control Product Quality Using Cnn Model With Vgg-16 Architecture," *E-Proceeding Of Engineering*, Vol. 8, No. 6, Pp. 3216–3225, 2022.

[9] H. M. Zangana, A. K. Mohammed, And F. Mahmood Mustafa, "Advancements In Edge Detection Techniques For Image Enhancement: A Comprehensive Review," *International Journal Of Artificial Intelligence & Robotics (Ijair)*, Vol. 6, No. 1, Pp. 29–39, 2024, Doi: 10.25139/Ijair.V6i1.8217.

[10] I. G. Prahmana And K. A. Br Sitepu, "Identification Of Land And Water Centella Asiatica Leaf Herbal Plants Using Digital Imagery With The Sobel Edge Detection Algorithm," *Journal Of Artificial Intelligence And Engineering Applications (Jaiea)*, Vol. 2, No. 2, Pp. 48–52, 2023, Doi: 10.59934/Jaiea.V2i2.158.

[11] D. Husen, "Evaluasi Teknik Augmentasi Data Untuk Klasifikasi Tumor Otak Menggunakan Cnn Pada Citra Mri," 2024.

[12] Y. M. R. Et.Al, "Detection Of Tumors From Mri Brain Images Using Cnn With Extensive Augmentation," *Turkish Journal Of Computer And Mathematics Education (Turcomat)*, Vol. 12, No. 6, Pp. 50–55, 2021, Doi: 10.17762/Turcomat.V12i6.1266.

[13] Z. Tuasamu *Et Al.*, "Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan Dfd Dan Flowchart Pada Bisnis Porobico," *Jurnal Bisnis Dan Manajemen(Jurbisman)*, Vol. 1, No. 2, Pp. 495–510, 2023.

[14] A. S. Putra And K. Haryono, "Implementasi Object Oriented Metodologi Dan Uml Pada Pengembangan Sistem Informasi Keuangan Organisasi," *Automata*, Vol. 2, Pp. 1–12, 2021.