DIAGNOCAR: PENDETEKSIAN OTOMATIS KERUSAKAN MOBIL MENGGUNAKAN DEEP LEARNING BERBASIS CITRA LEWAT PERANGKAT MOBILE

Ilham Julian Efendi[1*], Jabir Muktabir[2], Khaerunni Salsa Billah[3], Muhammad Vannes Al Qadri[4], Wa Ode Asriyani[5], Amar Adi Ismoyo[6].
[1]Program Studi Ilmu Komputer, Fakultas MIPA, Universitas Halu Oleo, [2,3,4,5,6]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo
[1]ilham.julian.efendi@uho.ac.id , [2]jabirmuktabir01@gmail.com ,
[3]khaerunnisalsabillah22062@gmail.com , [4]muhammadvannesalqadri@gmail.com ,
[5]waodeasriyanie1e122034@gmail.com, [6]adismoyam@gmail.com

Penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis berbasis arsitektur Convolutional Neural Network untuk mendeteksi jenis kerusakan pada mobil melalui data citra. Sistem ini diharapkan dapat mempercepat dan meningkatkan akurasi proses penilaian kerusakan pada klaim asuransi dan perbaikan kendaraan. Lima arsitektur telah diimplementasikan, yaitu MobileNetV2, EfficientNetV2S, NASNetMobile, ResNet50, dan model konvolusional yang dirancang sendiri, dengan menggunakan dataset berjumlah 1.594 citra yang terbagi dalam enam kelas kerusakan. Proses pelatihan mencakup praproses data, augmentasi, dan evaluasi performa berdasarkan akurasi pengujian. Hasil menunjukkan bahwa EfficientNetV2S memberikan performa terbaik dengan akurasi pengujian sebesar 84,2%. Studi ini membuktikan efektivitas arsitektur pretrained dalam membangun sistem inspeksi visual otomatis pada kendaraan dan membuka potensi pemanfaatan lebih lanjut dalam digitalisasi proses klaim asuransi otomotif.

***Kata Kunci*— Citra, DeepLearning, Deteksi, Kerusakan, Mobil, Mobile.**

I. PENDAHULUAN

Mobil tidak hanya menjadi simbol mobilitas modern, tetapi juga merupakan kebutuhan utama dalam aktivitas sehari-hari masyarakat. Sebagai sarana transportasi yang memberikan kenyamanan dan efisiensi, kendaraan pribadi seperti mobil telah memainkan peran penting dalam mendukung mobilitas sosial. Namun, seiring meningkatnya jumlah kendaraan, risiko kecelakaan dan klaim asuransi pun ikut bertambah, menuntut adanya sistem penanganan klaim yang lebih cepat, akurat, dan efisien.

Pesatnya pertumbuhan industri otomotif mendorong kebutuhan akan sistem klasifikasi kendaraan dan deteksi kerusakan yang lebih efisien. Di berbagai kota besar, kecelakaan lalu lintas menjadi hal yang lumrah akibat kepadatan kendaraan, kondisi jalan yang kurang baik, serta perilaku mengemudi yang tidak tertib. Kecelakaan semacam ini dapat menimbulkan kerusakan serius pada kendaraan, menyebabkan cedera, bahkan mengancam nyawa. Untuk menjawab tantangan tersebut, dibutuhkan sistem otomatis yang mampu menganalisis citra kendaraan, mengidentifikasi komponen seperti lampu depan, kaca, kap mobil, dan sekaligus mendeteksi kerusakan pada bagian bodi. Metode konvensional dalam pemeriksaan kerusakan cenderung memakan waktu dan rawan kesalahan, sehingga pendekatan berbasis kecerdasan buatan (AI) menjadi alternatif yang menjanjikan.

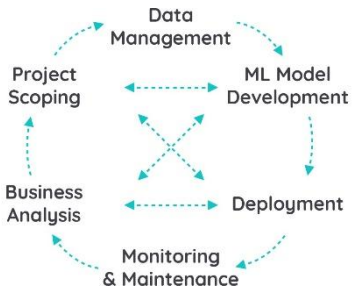
Penelitian sebelumnya tentang Deteksi Kerusakan Ban Mobil Menggunakan Convolutional Neural Network dengan Arsitektur ResNet-34 telah dilakukan oleh Mayana dan Leni [1], Dmytro dkk [2] yang telah melakukan penelitian tentang Automated Car Damage Assessment Using Computer Vision: Insurance Company Use Case, Cuevas dkk [3] telah meneliti mengenai A Computational Intelligence Approach for Car Damage Assessment, dan juga Smart Car Damage Assessment Using Enhanced YOLO Algorithm and Image Processing Techniques telah dianalisis oleh Ramazhan dkk [4], serta Deep learning-Based Car Damage Classification and Detection telah diteliti oleh Dwivedi dkk [5].

Penilaian kerusakan mobil dapat dilakukan melalui pendekatan konvensional maupun pendekatan berbasis kecerdasan buatan. Berdasarkan metode konvensional, seperti pemeriksaan visual oleh tenaga ahli, akurasi sangat bergantung pada subjektivitas dan pengalaman individu, serta memerlukan waktu dan biaya yang tidak sedikit.

Sementara itu, dengan pendekatan kecerdasan buatan, khususnya deep learning, penilaian kerusakan dapat dilakukan secara otomatis menggunakan data citra kendaraan yang dilatih melalui model pembelajaran mesin. Penelitian ini menggunakan lima metode arsitektur Convolutional Neural Network (CNN) untuk membangun dan mengevaluasi model klasifikasi kerusakan pada mobil. Model-model tersebut dipilih untuk membandingkan performa dan menentukan arsitektur yang paling optimal dalam mendeteksi serta mengklasifikasikan kerusakan kendaraan berbasis citra..

II. METODE PENELITIAN

Metodologi penelitian yang diterapkan dalam studi ini terdiri dari beberapa langkah atau proses [6]. Tahapan-tahapan penelitian tersebut dijelaskan sebagai berikut:



Gambar 1. Desain Sistem Pembelajaran Mesin (Sumber : Dicoding Indonesia [7])

Tahapan awal proyek dimulai dari Project Scoping, yaitu penentuan ruang lingkup dan tujuan dari proyek machine learning, termasuk identifikasi masalah bisnis dan kebutuhan pengguna. Setelah ruang lingkup ditetapkan, dilanjutkan dengan Data Management, yang mencakup pengumpulan, pembersihan, dan pengelolaan data yang relevan untuk memastikan kualitas input ke dalam sistem. Tahap berikutnya adalah ML Model Development, yaitu proses pengembangan model machine learning menggunakan teknik seperti supervised learning atau deep learning sesuai kebutuhan. Proses ini mencakup pemilihan algoritma, pelatihan model, dan evaluasi awal menggunakan data latih dan validasi. Model yang sudah dikembangkan kemudian masuk ke fase Deployment, di mana model diimplementasikan ke dalam lingkungan produksi agar dapat digunakan dalam pengambilan keputusan atau sistem otomatis. Setelah deployment, model akan melalui tahap Monitoring & Maintenance untuk memantau performa dan melakukan pembaruan jika terjadi degradasi akurasi seiring waktu. Selanjutnya dilakukan Business Analysis untuk menilai dampak dan manfaat model terhadap proses bisnis secara keseluruhan, serta mengidentifikasi potensi perbaikan. Proses kembali ke tahap Project Scoping untuk iterasi lanjutan atau pengembangan lebih lanjut. Selain alur utama yang berbentuk siklus, terdapat hubungan timbal balik antar tahap, seperti antara pengembangan model dengan manajemen data dan deployment, menandakan bahwa proses ini bersifat iteratif dan saling mempengaruhi.

A. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah dataset sekunder yang diambil dari platform Kaggle [8]. Dataset ini berjudul "*Car Damage Assessment*" dan berisi sekitar 1.500 gambar RGB unik berukuran 224 × 224 piksel. Setiap gambar merepresentasikan satu kendaraan dengan satu jenis kerusakan spesifik. Dataset ini dibagi menjadi dua bagian, yaitu subset pelatihan (*training*) dan validasi (*validation*), guna melatih model serta mengevaluasi kinerjanya terhadap data yang belum pernah dilihat sebelumnya. Kategori kerusakan yang tersedia dalam dataset mencakup berbagai jenis, seperti *Broken headlamp*, *Broken tail lamp*, *Glass shatter*, *Door scratch*, *Door dent*, *Bumper dent*, *Bumper scratch*, serta satu kelas tambahan yaitu *Unknown*, yang berisi gambar mobil dalam kondisi baik maupun rusak. Pembagian data yang sistematis ini memungkinkan model untuk belajar membedakan pola visual dari berbagai jenis kerusakan kendaraan secara efektif.

B. Preprocessing Data

Pre-processing data merujuk pada proses pengolahan data atau citra asli sebelum diproses oleh algoritma *Convolutional Neural Network* (CNN) [9]. *Preprocessing* bertujuan untuk meningkatkan representasi fitur, membersihkan gambar, atau memperbaiki *noise* serta nilai yang tidak lengkap pada gambar untuk mempersiapkannya pada tahap pemrosesan selanjutnya [10][11].

C. Arsitektur Convolutional Neural Network (CNN)

1) MobileNetV2

MobileNetv2 merupakan salah satu model arsitektur *Convolutional Neural Network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih dengan 2 *set hyper-parameters* untuk membangun model yang sangat kecil dan latensi rendah yang akan dengan mudah diimplementasikan sesuai kebutuhan *mobile* dan *embedded applications* [12].

2) EfficientNetV2S

EfficientNetV2S adalah model *Convolutional Neural Network* (CNN) yang merupakan bagian dari keluarga EfficientNetV2, dirancang untuk mencapai efisiensi parameter dan kecepatan pelatihan yang lebih baik dibandingkan pendahulunya [13].

3) NASNetMobile

Arsitektur NASNetMobile merupakan sebuah arsitektur yang memiliki jumlah parameter yang lebih sedikit jika dibandingkan dengan arsitektur MobileNet [14].

4) ResNet50

ResNet50 merupakan salah satu varian arsitektur dari CNN yang yang memiliki 50 lapisan atau layer dan memperkenalkan sebuah konsep baru yaitu *shortcut connections*. Penggunaan *shortcut* connection bertujuan untuk mengatasi masalah *vanishing gradient* problem yang muncul pada proses *training* jaringan

deep learning. Ketika sebuah jaringan menjadi semakin dalam, dapat timbul masalah *vanishing gradient* yang menyebabkan gradien menjadi sangat kecil. Hal ini dapat berdampak pada penurunan performa atau akurasi jaringan tersebut[15].

5) *CNN Custom*

Convolutional Neural Network (CNN) adalah algoritma yang dirancang untuk mengolah data seperti gambar atau suara [16]. Berasal dari *multilayer perceptron*, CNN merupakan bagian dari deep *learning* yang paling popular digunakan sebagai klasifikasi pada data [17]. Dalam penelitian ini, CNN bekerja melalui beberapa lapisan utama: lapisan *konvolusi*, *pooling*, *flatten*, dan *fully connected*. Lapisan *konvolusi* bertugas mengekstraksi fitur penting dari citra *input* menggunakan filter, sementara lapisan *pooling*, seperti *max pooling*, mengurangi dimensi data tanpa menghilangkan informasi penting. Data hasil ekstraksi diratakan oleh lapisan *flatten* menjadi vektor 1D untuk diproses lebih lanjut di lapisan *fully connected*. Di lapisan ini, hasil akhir diproses dengan fungsi *softmax* yang mengubah *output* menjadi probabilitas untuk setiap kelas. Fungsi aktivasi Softmax dinyatakan dalam bentuk persamaan:

$$f(x)_i = \frac{e^{xi}}{\sum_{j=1}^{k} e^{xj}} \quad (1)$$

Di mana $f(x)_i$ merepresentasikan *output* fungsi aktivasi Softmax untuk elemen ke-i dari vektor, x adalah vektor *input*, dan k menunjukkan jumlah kelas [18].

D. Training dan Validation

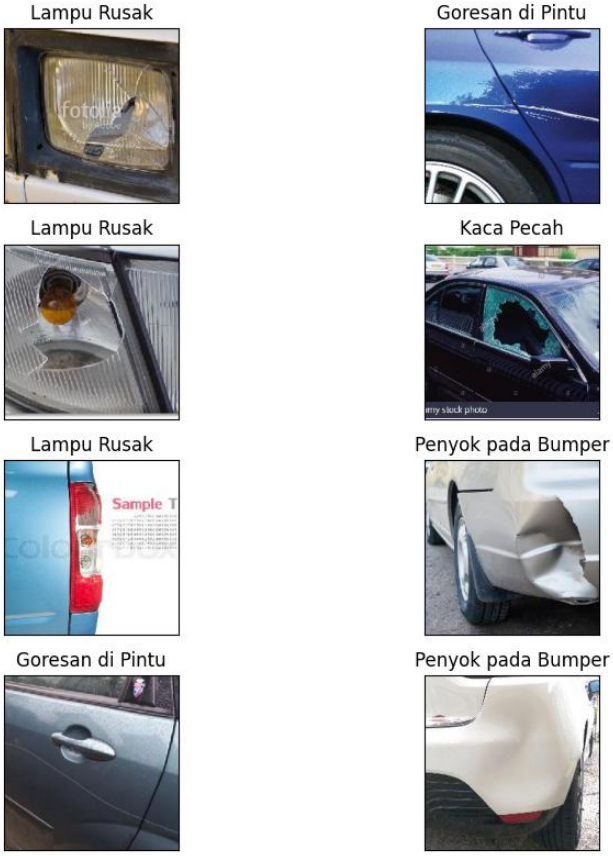
Pada tahap pembentukan data, dataset yang telah melewati proses preprocessing akan dibagi menjadi tiga kategori: data *training*, data *validation*, dan data *testing* [19] Data *training* digunakan untuk membangun model, sementara data *validation* berfungsi untuk menguji serta memvalidasi model yang telah dilatih [20]. Dalam penelitian ini, pembagian dataset mengadopsi proporsi *holdout method* dengan komposisi 80:20, di mana 80% data dialokasikan sebagai *training set* dan 20% sebagai *testing set*. Selanjutnya, *training set* tersebut dibagi kembali dengan rasio 80:20 untuk proses pelatihan dan validasi, menghasilkan distribusi akhir sebesar 64% data latih (*training*), 16% data validasi (*validation*), dan 20% data uji (*testing*). Untuk mengoptimalkan generalisasi model, proses pelatihan menerapkan teknik *data augmentation* seperti *random rotation* (30°), *width/height shift* (0.2), *shear* (0.2), *zoom* (0.2), *horizontal flip*, serta menggunakan *batch normalization* dan *dropout* (0.25–0.5) untuk mitigasi *overfitting*. Hyperparameter utama yang dioptimalkan meliputi *learning rate* (0.0001), *batch size* (32), dan *epochs* (100) dengan *early stopping* berdasarkan kinerja *validation loss*.

III. HASIL DAN PEMBAHASAN

Hasil dalam penelitian ini dimulai dengan tahap persiapan data, dilanjutkan dengan pelatihan model, dan diakhiri dengan evaluasi kinerja masing-masing model CNN yang digunakan untuk mendeteksi kerusakan pada mobil berdasarkan citra.

A. Data Preparation

Dataset yang digunakan dalam penelitian ini terdiri dari 1.594 gambar kerusakan mobil. Gambar-gambar tersebut dikelompokkan ke dalam enam kelas kerusakan, yaitu: goresan di bumper, goresan di pintu, kaca pecah, lampu rusak, penyok pada bumper, dan penyokan pada pintu.



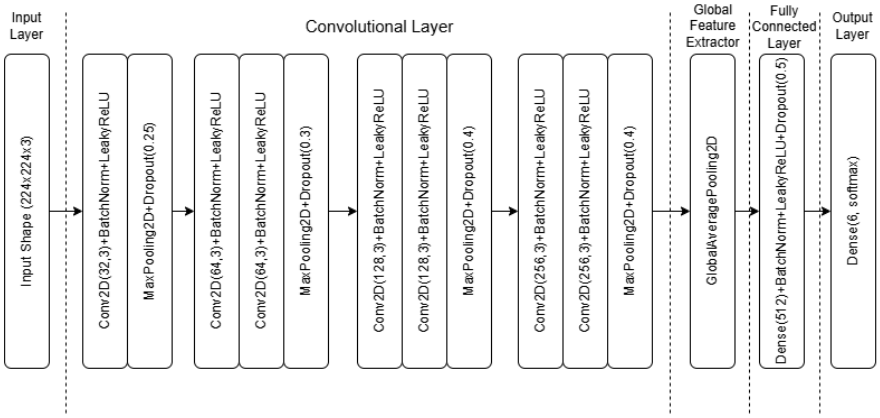
Gambar 2. Sampel Data citra yang digunakan

Sebelum digunakan dalam proses pelatihan model, data citra terlebih dahulu melalui tahap praproses, yang mencakup beberapa langkah berikut:

- *Resizing*: Seluruh gambar diubah ukurannya menjadi 224x224 piksel agar sesuai dengan ukuran *input* standar model CNN seperti MobileNetV2, EfficientNetV2S, dan ResNet50.
- Normalisasi: Nilai piksel citra yang semula berada dalam rentang 0–255 diubah ke rentang 0–1 dengan cara dibagi 255. Tujuannya untuk mempercepat proses pelatihan dan memperbaiki konvergensi model.
- Augmentasi data: Untuk meningkatkan kemampuan generalisasi model dan mencegah *overfitting*, dilakukan augmentasi data menggunakan teknik seperti rotasi acak, horizontal *flip*, *zoom*, dan *shifting*. Teknik ini memperbanyak

variasi data latih tanpa harus menambah jumlah data secara manual.

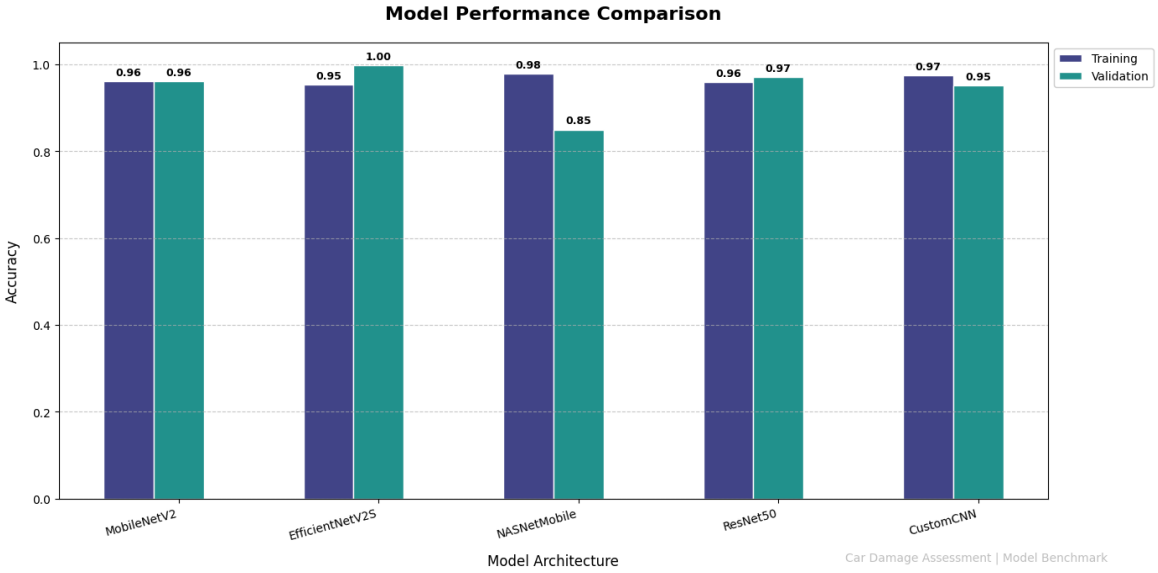
Seluruh gambar kemudian dibagi ke dalam tiga subset, yaitu data latih (training set), data validasi (validation set), dan data uji (testing set) dengan proporsi yang seimbang, dibagi dengan rasio 64:16:20, di mana 64% dialokasikan sebagai *training set* untuk pembelajaran model, 16% sebagai *validation set* untuk evaluasi selama pelatihan, serta 20% sebagai *testing set* yang berfungsi sebagai data independen untuk mengukur kinerja final model. Data tersebut digunakan untuk melatih lima arsitektur CNN, yaitu: MobileNetV2, EfficientNetV2S, NASNetMobile, ResNet50, dan CustomCNN



Gambar 3. Arsitektur Custom CNN

B. *Perbandingan Akurasi Training dan Validation*

Gambar 3. menunjukkan perbandingan akurasi antara data *training* dan *validation* untuk kelima model yang diuji. Secara umum, semua model menunjukkan akurasi *training* dan *validation* yang tinggi, dengan nilai di atas 95% kecuali NASNetMobile pada *validation.*

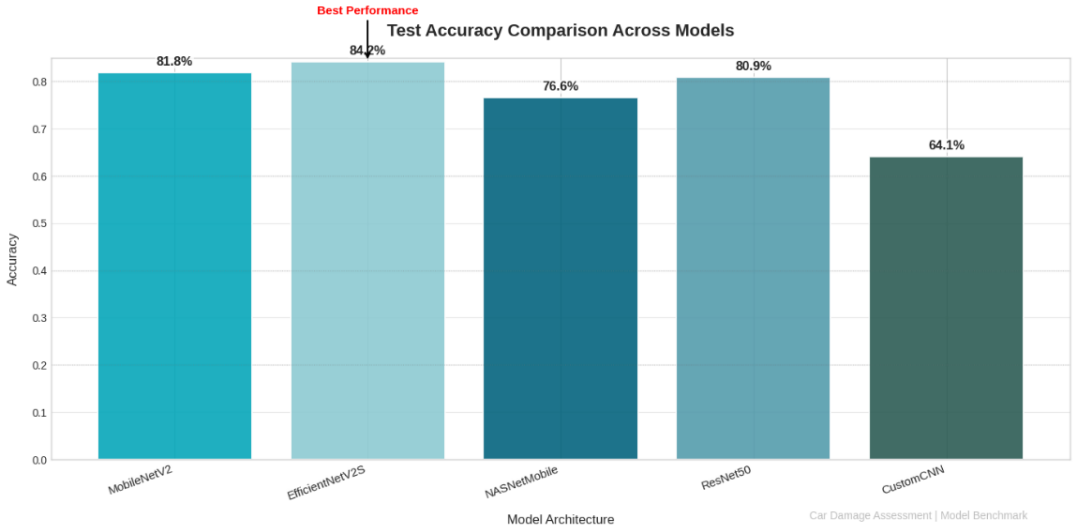


Gambar 4.Grafik perbandingan akurasi training dan validation

MobileNetV2: mencatat akurasi *training* dan *validation* sebesar 96%, menunjukkan kestabilan dan kemampuan generalisasi yang baik. EfficientNetV2S: mencapai akurasi *validation* sempurna sebesar 100%, dengan akurasi *training* sebesar 95%. Ini menunjukkan kemampuan generalisasi yang sangat baik meskipun ada indikasi underfitting ringan. NASNetMobile: memiliki gap signifikan antara *training* (98%) dan *validation* (85%), mengindikasikan *overfitting*. ResNet50: menunjukkan hasil konsisten antara *training* (96%) dan *validation* (97%). CustomCNN: memberikan akurasi *training* 97% dan *validation* 95%, yang tergolong baik untuk model buatan sendiri.

C. *Perbandingan Akurasi Pengujian (Testing)*

Gambar 3.3 memperlihatkan perbandingan akurasi testing pada data uji yang belum pernah dilihat oleh model sebelumnya. Hasil ini menunjukkan performa sebenarnya dari masing-masing arsitektur dalam mengklasifikasi gambar kerusakan mobil:



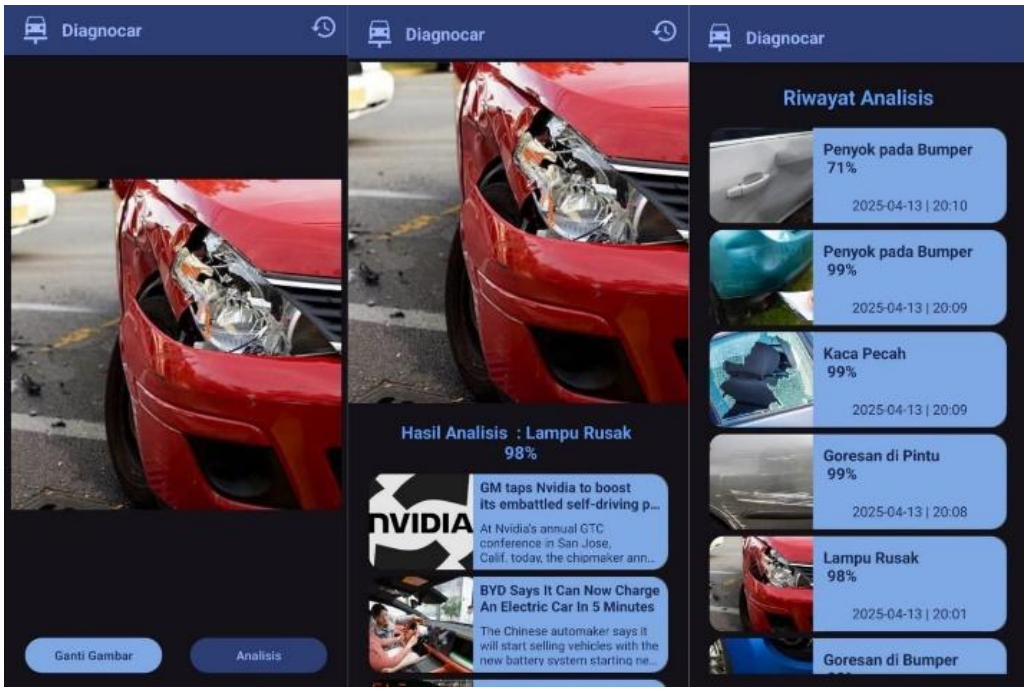
Gambar 5. Grafik perbandingan akurasi testing

EfficientNetV2S menempati peringkat pertama dengan akurasi sebesar 84,2%, ditandai sebagai performa terbaik dalam grafik. MobileNetV2 mengikuti dengan akurasi 81,8%, masih sangat kompetitif. ResNet50 mencatat akurasi 80,9%, sedikit di bawah MobileNetV2. NASNetMobile mengalami penurunan performa dengan akurasi hanya 76,6%, sejalan dengan indikasi *overfitting* yang terlihat pada grafik sebelumnya. CustomCNN menunjukkan performa terendah dengan akurasi 64,1%, menunjukkan keterbatasan model dalam menghadapi data uji yang kompleks.

D. Analisis Performa Model

Berdasarkan dua grafik tersebut, dapat disimpulkan bahwa EfficientNetV2S merupakan model dengan performa paling stabil dan akurat secara keseluruhan. Meskipun MobileNetV2 dan ResNet50 juga memberikan hasil yang sangat baik, EfficientNetV2S unggul secara konsisten baik pada data *validation* maupun *testing*. NASNetMobile yang awalnya menjanjikan pada *training* justru mengalami penurunan drastis saat *testing*, menegaskan adanya *overfitting*. CustomCNN sebagai model buatan sendiri tetap layak diapresiasi karena mampu mencapai performa yang cukup baik pada *training* dan *validation*, meski performa *testing*-nya masih perlu ditingkatkan. Secara keseluruhan, penggunaan arsitektur CNN yang sudah dikembangkan dan terlatih sebelumnya (*pre-trained* model) seperti EfficientNetV2S dan MobileNetV2 terbukti memberikan hasil lebih baik dan lebih stabil dibandingkan model custom.

E. Deployment pada Aplikasi berbasis Mobile



Gambar 6. Aplikasi Diagnocar

Aplikasi Diagnocar merupakan sistem berbasis kecerdasan buatan yang mampu menganalisis kerusakan pada kendaraan melalui citra visual secara otomatis. Dari tampilan antarmuka aplikasi, pengguna dapat mengunggah gambar mobil yang rusak, lalu sistem akan mendeteksi jenis kerusakan seperti lampu rusak, penyok pada bumper, kaca pecah, hingga goresan pada bodi mobil, lengkap dengan tingkat keyakinan dalam persentase. Hasil analisis ditampilkan secara langsung bersama riwayat deteksi sebelumnya, yang memuat informasi jenis kerusakan, waktu analisis, dan dokumentasi gambar.

IV. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan dan mengevaluasi lima model CNN untuk mendeteksi kerusakan pada mobil berdasarkan citra. Dari hasil eksperimen yang dilakukan, dapat disimpulkan bahwa:

- EfficientNetV2S adalah model dengan performa terbaik dan paling stabil, baik pada tahap training, validation, maupun testing. Model ini mencapai akurasi testing tertinggi sebesar 84,2%, menunjukkan kemampuannya dalam menggeneralisasi data yang belum pernah dilihat sebelumnya.
- MobileNetV2 dan ResNet50 menunjukkan performa yang sangat baik, dengan akurasi testing masing-masing sebesar 81,8% dan 80,9%. Kedua model ini memberikan hasil yang kompetitif meskipun tidak sebaik EfficientNetV2S.
- NASNetMobile menunjukkan adanya masalah overfitting, dengan akurasi validation yang jauh lebih rendah (85%) dibandingkan akurasi training (98%). Ini mengindikasikan bahwa model ini terlalu terlatih pada data latih dan tidak dapat menggeneralisasi dengan baik pada data uji.
- CustomCNN sebagai model buatan sendiri menunjukkan hasil yang baik pada training dan validation, namun performanya dalam pengujian data uji jauh di bawah model-model lainnya, dengan akurasi testing hanya mencapai 64,1%. Ini menunjukkan bahwa model buatan sendiri membutuhkan perbaikan lebih lanjut, seperti peningkatan arsitektur, optimasi regularisasi, dan menentukan strategi *training* yang optimal untuk menangani data yang lebih kompleks.

Secara keseluruhan, penggunaan model CNN yang telah terlatih sebelumnya (pre-trained models) seperti EfficientNetV2S dan MobileNetV2 terbukti lebih efektif dan efisien dibandingkan dengan model custom yang dikembangkan dari awal. Penelitian ini juga menunjukkan pentingnya evaluasi terhadap kemampuan generalisasi model agar dapat menghindari masalah seperti overfitting yang ditemukan pada NASNetMobile. Penelitian lebih lanjut dapat difokuskan pada pengembangan dan penyempurnaan model custom serta eksplorasi teknik augmentasi data yang lebih lanjut untuk meningkatkan performa model.

DAFTAR PUSTAKA

[1] H. Candra Mayana and D. Leni, "Deteksi Kerusakan Ban Mobil Menggunakan Convolutional Neural Network dengan Arsitektur ResNet-34," *SURYA TEKNIKA*, vol. 10, pp. 842–851, Dec. 2023.

[2] S. A. Pérez-Zarate, D. Corzo-García, J. L. Pro-Martín, J. A. Álvarez-García, M. A. Martínez-del-Amor, and D. Fernández-Cabrera, "Automated Car Damage Assessment Using Computer Vision: Insurance Company Use Case," *Applied Sciences (Switzerland)*, vol. 14, no. 20, Oct. 2024, doi: 10.3390/app14209560.

[3] F. Cuevas, L. Barenchi, I. Alonso, S. Nesmachnow, and P. Moreno-Bernal, "A Computational Intelligence Approach for Car Damage Assessment," in *Smart Cities*, S. Nesmachnow and L. Hernández Callejo, Eds., Cham: Springer Nature Switzerland, 2025, pp. 118–133.

[4] M. R. S. Ramazhan, A. Bustamam, and R. A. Buyung, "Smart Car Damage Assessment Using Enhanced YOLO Algorithm and Image Processing Techniques," *Information (Switzerland)*, vol. 16, no. 3, Mar. 2025, doi: 10.3390/info16030211.

[5] M. Dwivedi *et al.*, "Deep Learning-Based Car Damage Classification and Detection," in *Advances in Artificial Intelligence and Data Engineering*, N. N. Chiplunkar and T. Fukao, Eds., Singapore: Springer Nature Singapore, 2021, pp. 207–221.

[6] R. Firdaus, Joni Satria, and B. Baidarus, "Klasifikasi Jenis Kelamin Berdasarkan Gambar Mata Menggunakan Algoritma Convolutional Neural Network (CNN)," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 267–273, Dec. 2022, doi: 10.37859/coscitech.v3i3.4360.

[7] "Alur Proyek Machine Learning di Industri," Dicoding Indonesia. Accessed: Apr. 29, 2025. [Online]. Available: https://www.dicoding.com/academies/319/corridor

[8] H. Manssor, "Car Damage Assessment," Kaggle. Accessed: Apr. 03, 2025. [Online]. Available: https://www.kaggle.com/datasets/hamzamanssor/car-damage-assessment

[9] M. Zahir and R. Adi Saputra, "Deteksi Penyakit Retinopati Diabetes Menggunakan Citra Mata dengan Implementasi Deep Learning CNN," *Jurnal Teknoinfo*, vol. 18, no. 1, pp. 121–132, 2024, [Online]. Available: https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification

[10] N. Pramesti and R. Rianto, "Convolutional Neural Network for Identifying Tree Species Using Stem Images Convolutional Neural Network untuk Identifikasi Jenis Pohon Menggunakan Citra Batang," *Jurnal Informatika dan Teknologi Informasi*, vol. 20, no. 2, pp. 163–173, 2023, doi: 10.31515/telematika.v20i2.8774.

[11] A. Ardiansyah and N. F. Hasan, "Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 30–35, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1545.

[12] B. Karnadi and T. Handhayani, "Klasifikasi Jenis Buah dengan Menggunakan Metode MobileNetv2 dan Inceptionv3," *Jurnal Eksplora Informatika*, vol. 14, no. 1, pp. 35–42, Sep. 2024, doi: 10.30864/eksplora.v14i1.1067.

[13] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, 2021. [Online]. Available: https://github.com/google/

[14] T. Sahro Winanto, C. Rozikin, and A. Jamaludin, "Analisa Performa Arsitektur Transfer Learning Untuk Mengindentifikasi Penyakit Daun Pada Tanaman Pangan," *Journal of Applied Informatics and Computing (JAIC)*, vol. 7, no. 1, p. 68, Jul. 2023, [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[15] R. FATURRAHMAN, Y. S. HARIYANI, and S. HADIYOSO, "Klasifikasi Jajanan Tradisional Indonesia berbasis Deep Learning dan Metode Transfer Learning," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 4, p. 945, Oct. 2023, doi: 10.26760/elkomika.v11i4.945.

[16] U. Sri Rahmadhani and N. Lysbetti Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," *Jurnal Informatika : Jurnal Pengembangan IT*, vol. 8, no. 2, pp. 169–173, 2023.

[17] F. Rozzy, D. Candra Rini Novitasari, D. Yuliati, P. Permata Sani, U. Sunan Ampel Surabaya, and B. Meteorologi Klimatologi dan Geofisika, "Forecasting Sea Surface Salinity in the Eastern Madura Strait Using 1D Convolutional Neural Network Peramalan Salinitas Permukaan Air Laut di Selat Madura Bagian Timur Menggunakan

1D Convolutional Neural Network," *Telematika*, vol. 21, no. 1, pp. 14–30, Jun. 2024, doi: 10.31515/telematika.v21i1.8959.

[18] S. A. Sabrina and W. F. Al Maki, "Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *e-Proceeding of Engineering*, vol. 9, no. 3, pp. 1919–1927, Jun. 2022.

[19] Suprihanto, I. Awaludin, M. Fadhil, and M. A. Z. Zulfikor, "Analisis Kinerja ResNet-50 dalam Klasifikasi Penyakit pada Daun Kopi Robusta," *JURNAL INFORMATIKA*, vol. 9, no. 2, pp. 116–122, Oct. 2022, [Online]. Available: http://ejournal.bsi.ac.id/ejurnal/index.php/ji

[20] A. I. Saputra, I. Weni, and U. Khaira, "Implementasi Metode Convolutional Neural Network Untuk Deteksi Penyakit Pada Tanaman Kopi Arabika Melalui Citra Daun Berbasis Android," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 1, pp. 41–51, Oct. 2023, doi: 10.51454/decode.v4i1.231.