JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Identifikasi Risiko Kecelakaan Lalu Lintas Menggunakan Convolutional Neural Network (CNN) Berbasis Citra Dashcam

Sausan Faza[1], Rafika Puteri Wulandari[2], Findi Ayu Sariasih[3], Imam Sutoyo[4], Syifa Nur Rakhmah[5]

[1,2,3] Informatika, Unversitas Bina Sarana Informatika, Kramat Jakarta, Indonesia
[4,5]Teknologi Informasi, Unversitas Bina Sarana Informatika, Kramat Jakarta, Indonesia
Email: [1]sausanfaza47@gmail.com, [2]rafikaputeriwulandari1@gmail.com, [3]syifa.snk@bsi.ac.id, [4]findi.fav@bsi.ac.id, [5]imam.ity@bsi.ac.id

Email Penulis Korespondensi: [1]rafikaputeriwulandari1@gmail.com

Abstrak−Kecelakaan lalu lintas merupakan masalah global yang memerlukan pendekatan inovatif berbasis teknologi visi komputer. Penelitian ini bertujuan mengembangkan aplikasi web yang mampu mengidentifikasi probabilitas terjadinya kecelakaan kendaraan menggunakan citra *dashcam* dengan pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN). Pengembangan aplikasi dilakukan menggunakan metodologi *Feature Driven Development* (FDD) untuk memastikan integrasi fitur yang modular dan berorientasi pada kebutuhan pengguna. Dataset yang digunakan bersumber dari *Kaggle Car Crash Dataset* sebanyak 10.000 citra yang dibagi menjadi data *training* (7.000), *validation* (1.500), dan *testing* (1.500). Hasil penelitian menunjukkan bahwa model CNN berhasil mencapai akurasi pelatihan sebesar 83,66% dan akurasi validasi sebesar 82,13%. Meskipun demikian, terdapat tantangan pada ketidakseimbangan data yang menyebabkan nilai *recall* untuk kelas kecelakaan berada di angka 37,79%. Implementasi sistem pada antarmuka web memungkinkan pengguna mengunggah citra dan menerima hasil klasifikasi risiko berupa "High Risk" atau "Low Risk" secara *real-time*. Sistem ini diharapkan dapat menjadi prototipe awal bagi pengembangan teknologi keselamatan berkendara yang lebih responsif di masa depan.

Kata Kunci: Kecelakaan Lalu Lintas, Convolutional Neural Network, Feature Driven Development, Dashcam, Aplikasi Web.

Abstract−Traffic accidents are a global issue requiring innovative computer vision-based technological approaches. This research aims to develop a web application capable of identifying the probability of vehicle accidents using dashcam images through a Convolutional Neural Network (CNN) deep learning approach. The application was developed using the Feature Driven Development (FDD) methodology to ensure modular and user-oriented feature integration. The dataset, sourced from the Kaggle Car Crash Dataset, consists of 10,000 images divided into training (7,000), validation (1,500), and testing (1,500) sets. The results indicate that the CNN model achieved a training accuracy of 83.66% and a validation accuracy of 82.13%. However, challenges remain regarding data imbalance, which resulted in a recall value of 37.79% for the accident class. The system's implementation on a web interface allows users to upload images and receive real-time risk classification results as either "High Risk" or "Low Risk". This system is expected to serve as an initial prototype for developing more responsive driving safety technologies in the future.

Keywords: Traffic Accidents, Convolutional Neural Network, Feature Driven Development, Dashcam, Web Application.

1. PENDAHULUAN

Kecelakaan lalu lintas merupakan salah satu penyebab utama kematian dan cedera di seluruh dunia, yang menjadi masalah global dan memerlukan pendekatan inovatif berbasis teknologi [1]. Upaya pencegahan kecelakaan saat ini dapat ditingkatkan dengan memanfaatkan kemajuan teknologi visi komputer, salah satunya melalui analisis citra *dashcam* yang merekam kondisi jalan secara *real-time* [2], [3]. Namun, tantangan muncul karena analisis manual terhadap rekaman *dashcam* membutuhkan waktu dan sumber daya yang besar, serta rentan terhadap kesalahan manusia (*human error*) [4].

Oleh karena itu, diperlukan sebuah sistem otomatis yang mampu mengidentifikasi potensi risiko kecelakaan berdasarkan analisis visual secara *data*-driven menggunakan teknik *deep learning* [5]. Penelitian ini bertujuan untuk mengembangkan aplikasi web yang mampu mengidentifikasi probabilitas terjadinya kecelakaan menggunakan citra *dashcam* dengan pendekatan arsitektur *Convolutional Neural Network* (CNN) [6], [7]. Model CNN dipilih karena kemampuannya yang unggul dalam mengenali pola visual yang mengindikasikan kondisi lalu lintas berisiko [8]. Selain itu, mengingat adanya ketidakseimbangan data pada kelas kecelakaan, penelitian ini juga mempertimbangkan teknik penanganan *imbalance data* seperti SMOTE untuk meningkatkan performa model di masa depan [9], [10].

Pengembangan aplikasi ini dilakukan dengan metode *Feature Driven Development* (FDD) untuk memastikan struktur perangkat lunak yang modular dan terintegrasi [11]. Penggunaan metode FDD dalam perancangan aplikasi web memungkinkan pengembangan fungsionalitas yang lebih fokus pada fitur-fitur utama sistem [12]. Melalui pendekatan *Agile* yang sistematis [13], integrasi antara model *machine learning* berbasis Python dengan *interface* web dapat dilakukan secara efisien [14], sesuai dengan standar rekayasa perangkat lunak modern [15].

2. METODOLOGI

Metodologi yang digunakan dalam penelitian ini adalah *Feature Driven Development* (FDD). FDD merupakan salah satu metodologi pengembangan perangkat lunak yang termasuk dalam keluarga *Agile*, yang berfokus pada

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

pembangunan fungsionalitas sistem berdasarkan fitur-fitur yang bernilai bagi pengguna [11]. Tahapan penelitian ini disusun secara sistematis untuk memastikan integrasi antara model *Deep Learning* dan antarmuka website berjalan optimal.

2.1. Sumber Data dan Manajemen Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari *Kaggle Car Crash or Collision Prediction Dataset*. Data tersebut merupakan kumpulan citra yang telah dikurasi secara luas untuk mendukung penelitian sistem transportasi cerdas [8]. Dataset telah melalui tahap pra-pemrosesan dan dibagi ke dalam tiga kategori utama, yaitu *train*, *val* (validasi), dan *test*. Pembagian ini krusial untuk *mencegah* terjadinya kebocoran data (*data leakage*) dan memastikan model memiliki kemampuan generalisasi yang baik sesuai dengan prinsip dasar pengembangan *neural networks* [13], [15]. Dataset ini dikelola dalam folder *processed_dataset* yang telah dikategorikan menjadi dua kelas utama: *accident* (indeks 0) dan *normal* (indeks 1). Untuk memastikan validitas hasil pelatihan, dilakukan proses pembagian data (*data splitting*) dengan distribusi sebagai berikut:

a. **Data Training (7.000 citra):** Digunakan sebagai basis pembelajaran utama bagi model CNN dalam mengenali pola visual risiko kecelakaan.
b. **Data Validation (1.500 citra):** Berfungsi untuk mengoptimalkan parameter model selama proses pelatihan guna menghindari *overfitting*.
c. **Data Testing (1.500 citra):** Berfungsi sebagai data uji akhir untuk mengevaluasi performa riil model dan digunakan untuk pengujian fungsionalitas pada website.

2.2. Tahapan Feature Driven Development (FDD)

Implementasi pengembangan website dilakukan melalui lima fase FDD dapat dilihat pada Gambar 1:



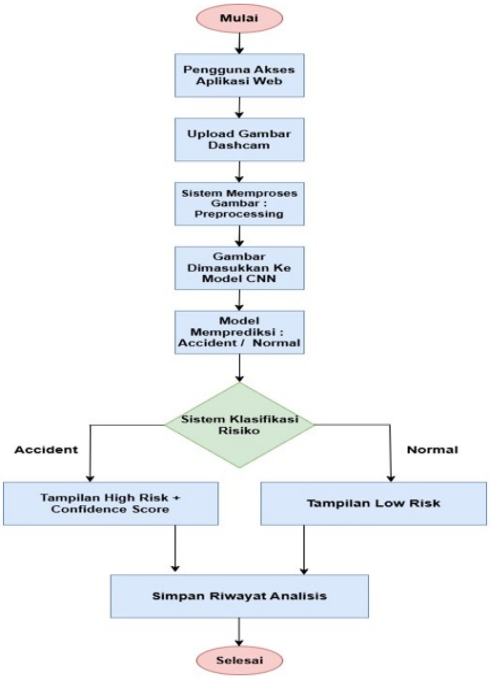
Gambar 1. Lima Fase FDD

a. **Develop Overall Model:** Tahap awal dimulai dengan pembentukan tim dan penentuan cakupan sistem. Pada fase ini, dilakukan analisis kebutuhan fungsional sistem, yaitu sebuah platform berbasis web yang mampu menerima input gambar dari pengguna dan memberikan output berupa probabilitas kecelakaan. Di sini, dilakukan juga perancangan arsitektur *Deep Learning* berbasis CNN yang akan diintegrasikan ke dalam website [5], [7].
b. **Build Feature List:** Berdasarkan model keseluruhan, disusun daftar fitur utama yang mencakup fitur unggah gambar *dashcam*, pemrosesan citra oleh model *deep learning*, visualisasi probabilitas risiko.
c. **Plan by Feature:** Pada tahap ini dilakukan penyusunan jadwal pengembangan berdasarkan kompleksitas fitur. Prioritas utama diberikan pada pengembangan model CNN dan integrasinya dengan *backend* web menggunakan Flask/FastAPI [14]. Perencanaan ini juga mencakup persiapan dataset yang terdiri dari 10.000 citra (8007 normal dan 1993 accident) yang akan digunakan untuk pengujian model.
d. **Design by Feature:** Tahap perancangan melibatkan pembuatan diagram alir sistem (*flowchart*) dan *Activity Diagram* yang sesuai dengan standar rekayasa perangkat lunak [15]. Pada fase ini, dirancang antarmuka website yang intuitif menggunakan HTML/CSS dan JavaScript, serta perancangan skema klasifikasi model agar mampu menghasilkan output nilai probabilitas yang akurat. Pada tahap perancangan, disusun alur logika sistem yang digambarkan melalui *flowchart* penelitian. Perancangan ini mencakup tahapan input gambar oleh pengguna, proses *preprocessing* oleh sistem, hingga logika pengambilan keputusan oleh model CNN untuk menghasilkan output klasifikasi risiko, flowchart dapat dilihat pada Gambar 2.

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 2. Alur Sistem

e. **Build by Feature:** Fase terakhir adalah implementasi kode program. Di sini, model CNN yang telah dilatih kemudian dimuat (*loading model*) ke dalam sistem web. Dilakukan pengujian untuk memastikan setiap fitur berjalan dengan baik, mulai dari proses unggah hingga hasil prediksi muncul di layar pengguna. Pada tahap ini pula, ditemukan karakteristik performa model yang memiliki akurasi 83,66%, namun memerlukan evaluasi lebih lanjut pada aspek *recall* karena adanya ketidakseimbangan data [9].

3. HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil dari eksperimen pelatihan model *Deep Learning* serta implementasinya ke dalam antarmuka perangkat lunak berbasis web. Proses pengujian dilakukan dengan mengacu pada standar evaluasi visi komputer untuk memastikan bahwa model tidak hanya memiliki akurasi yang tinggi secara administratif, tetapi juga memiliki keandalan dalam mendeteksi objek visual yang kompleks di jalan raya [6]. Pembahasan akan dimulai dari karakteristik dataset yang digunakan, metrik performa model CNN, hingga analisis mendalam mengenai keterbatasan deteksi pada kondisi data yang tidak seimbang..

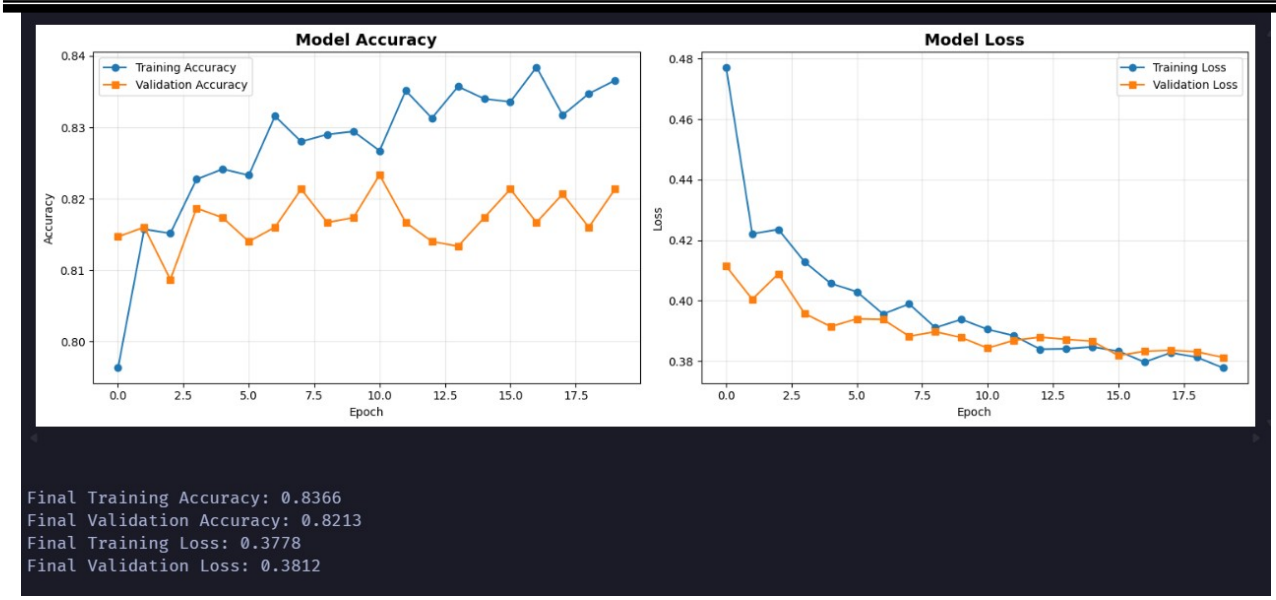
3.1. Performa Pelatihan Model

Berdasarkan proses pelatihan selama 20 *epoch* menggunakan kerangka kerja Python yang umum digunakan dalam pengembangan kecerdasan buatan [7], model menunjukkan hasil yang stabil. Grafik akurasi dan *loss* dapat dilihat pada Gambar 3:

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

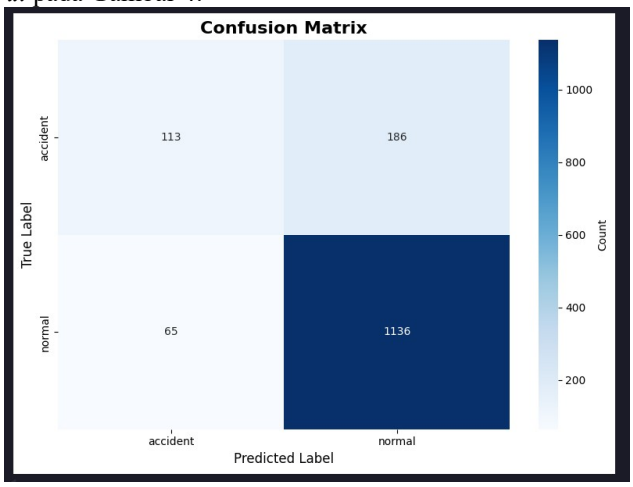


Gambar 3. Accuracy Model

Hasil akhir menunjukkan **Final Training Accuracy sebesar 83,66%** dan **Final Validation Accuracy sebesar 82,13%**. Nilai *loss* yang rendah dan kurva yang konvergen menunjukkan bahwa arsitektur CNN yang dirancang telah mampu mengekstraksi fitur visual dari citra *dashcam* secara efektif [5].

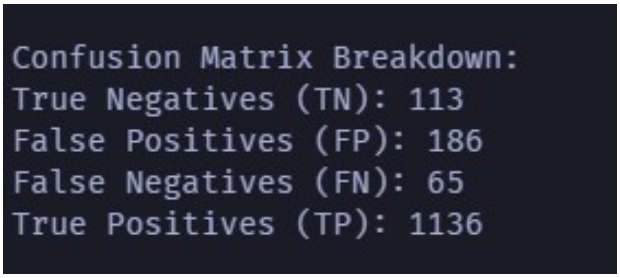
3.2. Evaluasi Klasifikasi

Pengujian dilakukan menggunakan 1.500 citra uji yang terdiri dari 299 data *accident* dan 1.201 data *normal*. Hasil prediksi dirinci melalui *Confusion Matrix* pada Gambar 4:



Gambar 4. *Confusion Matrix*

Berdasarkan *Confusion Matrix* tersebut, model berhasil memprediksi 113 data *accident* dan 1.136 data *normal* dengan benar. Namun, terdapat 186 data *accident* yang salah diprediksi sebagai *normal*. Performa detailnya ditunjukkan pada Gambar 5:



Gambar 5. *Peforma Detail Confusion Matrix*

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

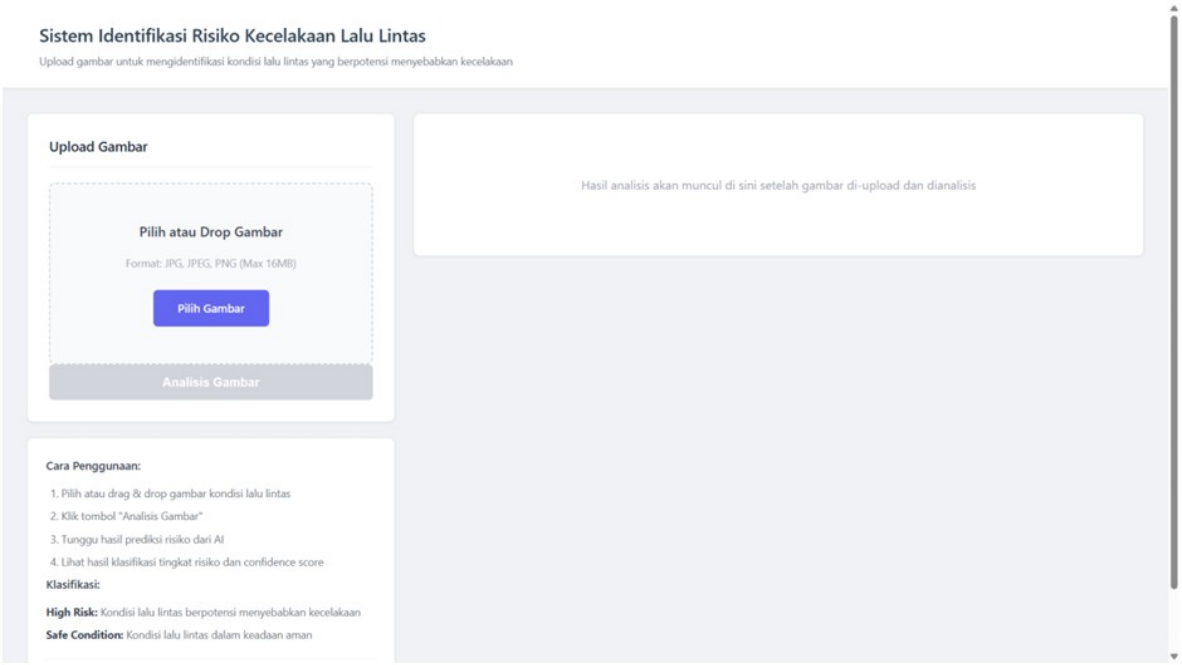
3.3. Analisis Ketidakseimbangan Data

Dari Gambar 3, terlihat bahwa nilai *Recall* untuk kelas kecelakaan hanya **37,79%**. Hal ini merupakan dampak dari *imbalance dataset*, di mana jumlah sampel kelas mayoritas jauh lebih banyak dibandingkan kelas minoritas [10]. Dalam literatur *machine learning*, kondisi ini sering kali menyebabkan model menjadi bias terhadap kelas yang lebih banyak muncul saat pelatihan [12]. Oleh karena itu, pada pengembangan selanjutnya, penerapan teknik *oversampling* seperti SMOTE sangat disarankan untuk menyeimbangkan distribusi kelas dan meningkatkan sensitivitas model terhadap kejadian kecelakaan [9].

3.4. Implementasi pada Sistem Web

Model CNN yang telah optimal kemudian diintegrasikan ke dalam aplikasi web menggunakan *framework* Flask yang mendukung fleksibilitas pengembangan sisi *server* [14]. Antarmuka dikembangkan untuk memudahkan pengguna mengunggah gambar dan menerima umpan balik instan mengenai probabilitas risiko. Integrasi ini menunjukkan bahwa model visi komputer yang kompleks dapat diakses dengan mudah oleh pengguna awam melalui perancangan sistem yang berorientasi pada fitur (FDD) dan pengalaman pengguna yang baik, berikut adalah beberapa fitur yang tersedia dalam aplikasi kami:

a. **Fitur Unggah Gambar dan Analisis** Halaman utama website menyediakan fitur bagi pengguna untuk mengunggah citra yang berasal dari rekaman *dashcam*. Setelah gambar dipilih dari penyimpanan lokal, sistem akan secara otomatis menjalankan proses *preprocessing* dan mengirimkan data tersebut ke model CNN untuk diidentifikasi, tampilan website bisa dilihat pada Gambar 6.



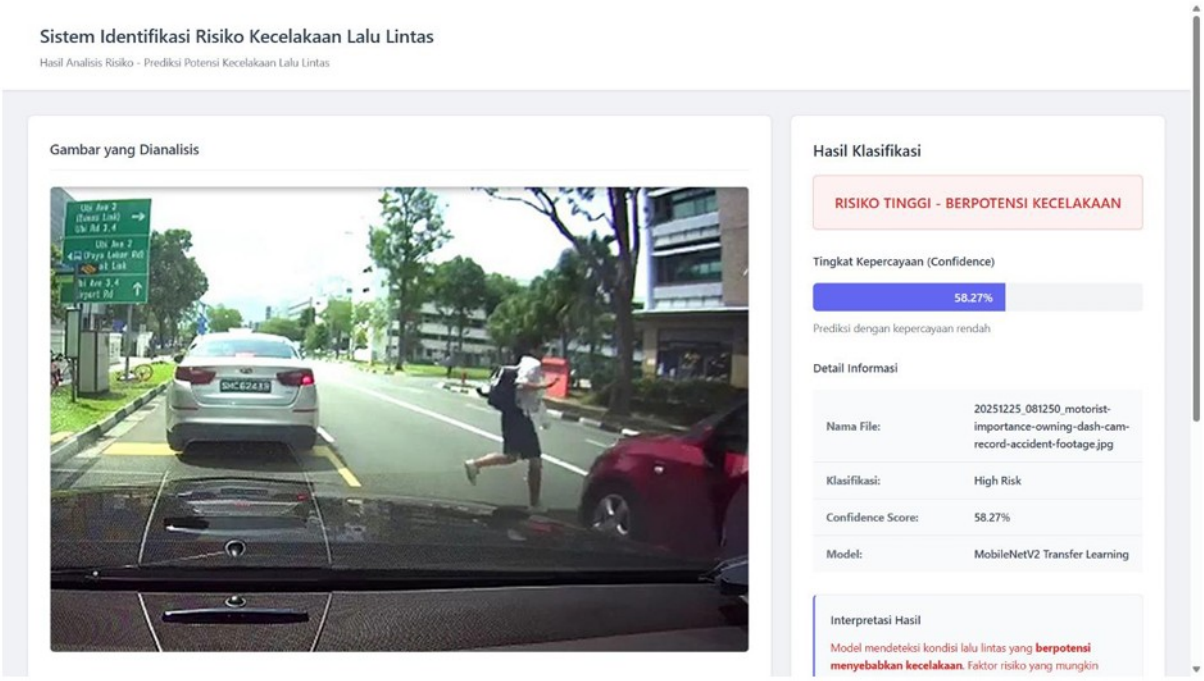
Gambar 6. Tampilan Webiste Fitur Unggah Citra

b. **Visualisasi Hasil Prediksi** Hasil analisis ditampilkan dalam bentuk label klasifikasi yang jelas. Jika model mendeteksi adanya indikasi kecelakaan, sistem akan menampilkan notifikasi **"High Risk"**. Sebaliknya, jika kondisi jalan dinilai aman, sistem akan menampilkan label **"Low Risk/Normal"**. Selain label, sistem juga menampilkan *confidence score* dalam bentuk persentase untuk menunjukkan tingkat keyakinan model terhadap prediksi tersebut, tampilan hasil bisa dilihat pada Gambar 7.

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 7 Nomor 2 April 2026, Page 152-158
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin



Gambar 7. Tampilah hasil klasifikasi probabilitas kecelakaan

4. KESIMPULAN

Berdasarkan rangkaian penelitian yang telah dilaksanakan, dapat disimpulkan bahwa pengembangan sistem prediksi kecelakaan lalu lintas menggunakan pendekatan *Deep Learning* dengan arsitektur *Convolutional Neural Network* (CNN) telah berhasil diimplementasikan ke dalam platform berbasis web. Melalui penerapan metodologi *Feature Driven Development* (FDD), integrasi antara model kecerdasan buatan dengan antarmuka pengguna dapat berjalan secara modular dan terstruktur, memberikan kemudahan bagi pengguna dalam melakukan analisis risiko secara instan melalui unggahan citra *dashcam* [11], [15]. Hasil pengujian menunjukkan bahwa model mampu mencapai akurasi pelatihan sebesar 83,66% dan akurasi validasi sebesar 82,13%, yang mengindikasikan kemampuan model yang cukup baik dalam melakukan generalisasi terhadap data visual jalan raya yang bervariasi [5]. Namun, penelitian ini juga mengungkap tantangan signifikan terkait ketidakseimbangan dataset (*imbalance dataset*), di mana nilai *Recall* untuk kelas kecelakaan (*accident*) hanya menyentuh angka 37,79%. Hal ini menunjukkan bahwa meskipun akurasi sistem secara keseluruhan cukup tinggi, kemampuan model dalam mendeteksi kejadian kecelakaan yang sebenarnya masih perlu ditingkatkan agar tidak terjadi banyak luput deteksi (*false negative*) [10]. Oleh karena itu, sebagai langkah pengembangan di masa depan, penerapan teknik penyeimbangan data seperti *Synthetic Minority Over-sampling Technique* (SMOTE) atau teknik augmentasi data yang lebih lanjut menjadi sangat esensial untuk memperkuat sensitivitas model pada kelas minoritas [9]. Selain itu, transisi dari analisis citra statis menjadi analisis berbasis video secara *real-time* juga dipandang perlu untuk meningkatkan relevansi sistem ini dalam ekosistem transportasi cerdas yang lebih luas [2]. Secara keseluruhan, penelitian ini telah berhasil membuktikan bahwa sinergi antara rekayasa perangkat lunak modern dan teknik *computer vision* dapat menghasilkan alat bantu keselamatan berkendara yang fungsional, edukatif, dan memiliki potensi besar untuk dikembangkan menjadi sistem peringatan dini guna menekan angka fatalitas di jalan raya [1], [14].

UCAPAN TERIMAKASIH

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua yang telah memberikan kesempatan, fasilitas, serta lingkungan belajar yang mendukung selama proses penelitian ini. Terima kasih juga kepada dosen pembimbing yang selalu sabar memberikan arahan, kritik, dan masukan sehingga penelitian ini dapat terselesaikan dengan baik.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[2] S. Tariq and others, "Accident Detection and Classification using Surveillance and Dashcam Images via Deep Learning," *International Journal of Intelligent Transportation Systems Research*, 2021.
[3] S. Li and X. Wu, "Vision-Based Highway Traffic Accident Detection Using Deep CNNs," *IEEE Access*, 2019.

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**

[4] A. Bansal and others, "Real-time Vehicle Accident Detection and Classification using Convolutional Neural Networks," *Journal of Computing and Information Technology*, 2020.
[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun ACM*, 2012.
[7] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.
[8] J. Deng and others, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
[9] N. V Chawla and others, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, 2002.
[10] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans Knowl Data Eng*, 2009.
[11] S. R. Palmer and J. M. Felsing, *A Practical Guide to Feature-Driven Development*. Prentice Hall, 2002.
[12] P. Tamir, "Implementasi Metode Feature Driven Development Pada Perancangan Web Portal Lomba," *Kohesi: Jurnal Multidisiplin Saintek*, vol. 1, no. 6, pp. 1–10, 2023.
[13] J. Highsmith, *Agile Software Development Ecosystems*. Addison-Wesley Professional, 2002.
[14] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2018.
[15] I. Sommerville, *Software Engineering*. Pearson Education, 2015.

This is an open access article under the CC–BY-SA license

Terakreditasi SINTA 5 SK :72/E/KPT/2024
Submitted:**11/11/2025**; Accepted: **05/01/2025**; Published: **30/03/2026**