

PENGEMBANGAN APLIKASI PENDETEKSI KERETAKAN JALAN BERBASIS ANDROID DENGAN IMPLEMENTASI ALGORITMA HYBRID CNN-LSTM

Ihham Akbar Pradana¹⁾, Ani Dijah Rahajoe²⁾, Andreas Nugroho Sihananto³⁾

E-mail : ¹⁾anidijah.if@upnjatim.ac.id

^{1,2,3}Departemen Informatika, Fakultas Ilmu Komputer,
Universitas Pembangunan Nasional Veteran Jawa Timur

Abstrak

Infrastruktur jalan yang berkualitas memegang peran penting dalam pertumbuhan ekonomi suatu negara. Namun, dengan meningkatnya volume kendaraan dan faktor lingkungan, kerusakan jalan menjadi masalah yang tak terhindarkan dan memerlukan perhatian serius. Metode tradisional dalam mendeteksi kerusakan jalan seringkali dilakukan melalui inspeksi manual yang tidak hanya memakan waktu tetapi juga cenderung subjektif dan kurang akurat. Penelitian ini mengusulkan pengembangan aplikasi Android yang inovatif, yang memanfaatkan teknologi Deep Learning untuk mendeteksi kerusakan jalan secara akurat dan efisien. Aplikasi ini menggabungkan Convolutional Neural Network (CNN) untuk ekstraksi ciri visual dari gambar dan Long Short-Term Memory (LSTM) untuk memahami konteks sekuensial dari data hasil luaran lapisan-lapisan CNN. Dataset yang digunakan dalam pengembangan model ini bersumber dari kumpulan gambar kerusakan jalan sebagai representasi dari berbagai kondisi jalanan perkotaan di Indonesia. Melalui proses pelatihan, model CNN-LSTM yang ini diintegrasikan ke dalam aplikasi dengan menggunakan TensorFlow Lite. Pengembangan aplikasi Android dilakukan dengan mempertimbangkan arsitektur aplikasi yang baik dan efisien, menjamin bahwa aplikasi tidak hanya responsif dan intuitif tetapi juga hemat sumber daya. Melalui integrasi teknologi canggih dan pendekatan pengembangan yang terfokus, aplikasi ini berpotensi menjadi alat penting dalam usaha pemeliharaan infrastruktur jalan, memberikan solusi yang praktis dan inovatif untuk mendeteksi kerusakan jalan dengan cepat dan akurat.

Kata kunci: *keretakan jalan, android, CNN, LSTM*

1. PENDAHULUAN

Infrastruktur jalan yang berkualitas memegang peran penting dalam pembangunan dan pertumbuhan ekonomi suatu negara. Jalan yang baik tidak hanya menambah efisiensi mobilitas barang dan jasa, tetapi juga memastikan keselamatan pengguna jalan. Namun, dengan meningkatnya volume kendaraan dan faktor lingkungan, kerusakan jalan menjadi masalah yang tak terhindarkan dan memerlukan perhatian serius [3].

Secara tradisional, deteksi kerusakan jalan dilakukan oleh pekerja melalui inspeksi manual untuk melihat kondisi jalan atau menggunakan metode yang memerlukan investasi peralatan mahal dan kompleks. Tentunya proses ini akan memakan waktu, tenaga dan biaya yang besar. Selain itu, cara tradisional ini seringkali subjektif dan tidak konsisten, tergantung pada pengalaman dan penilaian inspektur. Dengan berkembangnya teknologi, terutama di bidang kecerdasan buatan (AI) dan pengolahan citra digital, terbuka peluang untuk mengatasi permasalahan ini secara lebih efisien [3].

Convolutional Neural Networks (CNN) yang merupakan sebuah metode dalam deep learning kerap digunakan untuk mengolah data berbentuk grid seperti gambar. CNN telah menunjukkan hasil yang menjanjikan dalam berbagai aplikasi pengolahan citra berdasarkan penelitian-penelitian sebelumnya. CNN memiliki potensi yang besar untuk diaplikasikan dalam pengembangan teknologi pendeteksi pola keretakan jalan pada gambar dengan akurasi pengenalan pola yang baik [2].

Long Short-Term Memory (LSTM) networks adalah varian dari Recurrent Neural Networks (RNN) yang dirancang untuk mengatasi masalah dependensi jangka panjang. LSTM sangat efektif dalam memproses dan membuat prediksi berdasarkan data sekuensial, seperti seri waktu atau urutan teks. Dengan menggunakan struktur sel memori, LSTM dapat mengingat informasi untuk durasi yang panjang, membuatnya sangat sesuai untuk tugas-tugas yang memerlukan pemahaman konteks atau pola yang berkembang seiring waktu. Dalam konteks deteksi keretakan jalan, LSTM dapat dimanfaatkan untuk mempelajari bagaimana keretakan jalan berkembang, memberikan wawasan penting untuk prediksi dan pemeliharaan preventif [1].

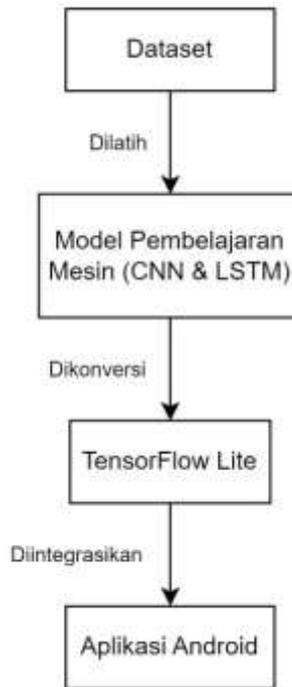
Menggabungkan kekuatan Convolutional Neural Networks (CNN) dan Long Short-Term Memory (LSTM) dapat menciptakan algoritma hybrid yang sangat kuat untuk mendeteksi dan menganalisis keretakan jalan. Dalam pendekatan ini, CNN digunakan untuk ekstraksi fitur visual dari gambar keretakan jalan, mengidentifikasi dan mengklasifikasikan jenis kerusakan dengan presisi tinggi. Output dari CNN kemudian disalurkan ke LSTM untuk memanfaatkan aspek temporal dan sekuensial dari data [2]. Algoritma hybrid ini menggabungkan kemampuan analisis citra yang canggih dengan pemahaman mendalam tentang dinamika keretakan jalan, menawarkan solusi komprehensif untuk pendeteksian keretakan jalan yang lebih efektif dan akurat.

Di sisi lain, penetrasi smartphone, khususnya yang berbasis sistem operasi Android, telah mencapai angka yang sangat tinggi di berbagai lapisan masyarakat. Platform Android menawarkan kemudahan akses dan fleksibilitas yang tinggi, menjadikannya platform yang ideal untuk aplikasi yang memanfaatkan AI. Dengan memanfaatkan kamera smartphone, aplikasi Android yang dilengkapi dengan AI dapat digunakan untuk mengambil gambar permukaan jalan, kemudian algoritma hybrid CNN-LSTM akan menganalisis gambar tersebut untuk mendeteksi adanya keretakan.

Penelitian ini diharapkan dapat memberikan kontribusi signifikan bagi bidang teknik sipil dalam hal pemeliharaan infrastruktur, serta dalam bidang teknologi informasi melalui pengembangan aplikasi mobile yang inovatif. Selain itu, penelitian ini juga memberikan wawasan baru dalam aplikasi teknologi AI, khususnya CNN dan LSTM, untuk solusi praktis dalam masalah kehidupan nyata.

2. METODOLOGI

Arsitektur sistem aplikasi pendeteksi keretakan jalan ini dirancang untuk memfasilitasi pengolahan data secara efisien dan akurat langsung di perangkat mobile pengguna dengan menggunakan TensorFlow Lite. Sistem ini tetap menggunakan teknik pembelajaran mesin yang canggih, yaitu Convolutional Neural Network (CNN) untuk pengolahan gambar dan Long Short-Term Memory (LSTM) untuk analisis urutan data. Kombinasi Teknik dan integrasi menggunakan TensorFlow Lite ini memungkinkan deteksi keretakan jalan dari gambar yang diambil oleh pengguna tanpa memerlukan koneksi ke server backend [5]. Berikut adalah alur utama dari pembangunan arsitektur sistem :



Gambar 1. Alur Arsitektur Pengembangan Aplikasi

Proses pengembangan sistem aplikasi pendeteksi keretakan jalan berbasis Android ini dimulai dari tahap pengumpulan dataset yang akan digunakan untuk melatih model pembelajaran mesin. Dataset yang berkualitas dan representatif dari kondisi jalanan perkotaan di Indonesia ini merupakan fondasi penting dalam mengembangkan model yang akurat.

Selanjutnya, model pembelajaran mesin, yang menggabungkan arsitektur Convolutional Neural Network (CNN) untuk pengolahan gambar dan Long Short-Term Memory (LSTM) untuk analisis data urutan, dilatih menggunakan dataset yang ada. Model yang telah dilatih tersebut kemudian diubah menjadi format yang kompatibel dengan TensorFlow Lite, sebuah kerangka kerja ringan yang memungkinkan model pembelajaran mesin untuk dijalankan pada perangkat mobile.

Setelah konversi, model yang dioptimalkan ini diintegrasikan ke dalam aplikasi Android. Integrasi ini memungkinkan aplikasi untuk memproses gambar keretakan jalan langsung pada perangkat pengguna. Aplikasi bertanggung jawab untuk menangkap gambar melalui kamera perangkat, menjalankan model deteksi keretakan yang telah diintegrasikan, dan menampilkan hasil analisis secara instan kepada pengguna.

2.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini bersumber dari dataset yang tersedia pada platform Kaggle. Dataset ini mengandung serangkaian gambar dengan dua jenis kerusakan jalan yaitu jalan berlubang dan retak dengan berbagai tingkat kerusakan. Dataset ini dipilih karena kekayaan fitur visual yang dapat dimanfaatkan untuk melatih model dengan kemampuan deteksi yang baik.

Table 1. Rincian Jumlah Dataset

No	Label	Jumlah
1.	Jalan Retak Berlubang	335
2.	Jalan Retak	200
3.	Jalan Tidak Rusak	199
	Total	734



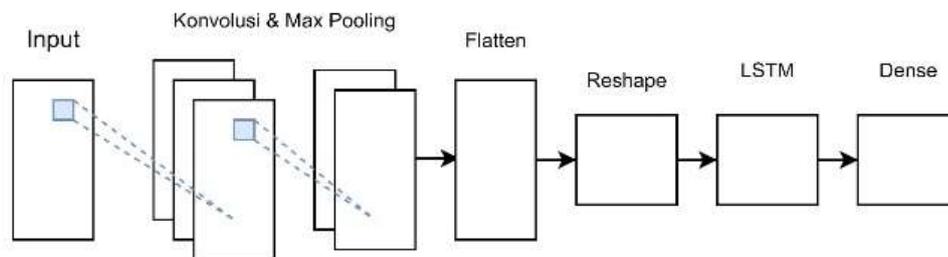
Gambar 2. Contoh Gambar Dataset Kondisi Jalan

Dataset akan dibagi menjadi tiga set berbeda dengan proporsi 90%, 80% dan 70% untuk pelatihan, dan sisanya masing-masing 10%, 20% dan 30% untuk pengujian. Model yang memberikan akurasi tertinggi akan dipilih untuk penggunaan lebih lanjut. Dataset yang digunakan dalam penelitian ini terdiri dari dua set data utama yaitu data pelatihan (training data) dan data pengujian (test data). Berikut adalah rincian dari dataset tersebut berdasarkan proporsinya :

Table 2. Jumlah Data Pelatihan dan Pengujian

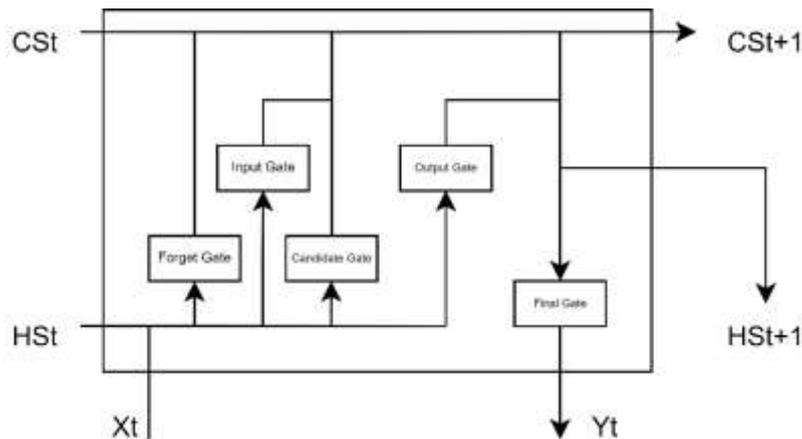
Data	10% Uji	20% Uji	30% Uji
Pelatihan	660	587	513
Pengujian	74	147	221

2.2 Implementasi Algoritma



Gambar 3. Arsitektur Algoritma Hybrid CNN dan LSTM

Gambar diatas menggambarkan arsitektur dari algoritma Hybrid CNN-LSTM yang digunakan dalam aplikasi pendeteksi keretakan. Proses dimulai dengan input, yaitu gambar yang diambil oleh pengguna. Gambar tersebut pertama kali melalui serangkaian lapisan konvolusi, di mana fitur-fitur penting seperti tepi dan tekstur diekstrak. Selanjutnya, lapisan pooling mengurangi dimensi dari fitur-fitur tersebut untuk mempercepat komputasi dan mengurangi kemungkinan overfitting. Hasilnya kemudian diratakan (flatten) menjadi vektor satu dimensi yang diubah bentuknya (reshape) menjadi urutan fitur yang siap diolah oleh lapisan LSTM. Lapisan LSTM kemudian menangkap informasi sekuensial dari fitur-fitur tersebut. Akhirnya, lapisan dense dengan aktivasi softmax digunakan untuk menghasilkan prediksi akhir, untuk tugas klasifikasi non biner [2], [4].



Gambar 4. Arsitektur LSTM dalam Algoritma Hybrid CNN-LSTM

Gambar di atas adalah pembedahan arsitektur LSTM dalam algoritma hybrid yang digunakan dalam penelitian ini. Input (X_t), yang dalam konteks deteksi keretakan jalan adalah vektor fitur yang diekstraksi oleh CNN dari gambar, bersama hidden state sebelumnya (HSt) dan cell state sebelumnya (CSt), digunakan untuk menentukan informasi apa yang harus dilupakan melalui forget gate. Input gate dan candidate gate memutuskan informasi baru yang akan disimpan dalam cell state yang diperbarui ($CSt+1$). Output gate kemudian menentukan bagian dari cell state yang akan digunakan untuk menghasilkan hidden state baru ($HSt+1$), yang juga menjadi output (Yt) pada waktu t . LSTM memproses urutan vektor fitur yang dihasilkan oleh lapisan CNN, mempertahankan informasi penting dan mengabaikan informasi yang tidak relevan, untuk mendeteksi pola temporal yang mungkin menunjukkan keretakan. Output dari LSTM ini kemudian dapat digunakan untuk membuat prediksi akhir tentang keberadaan keretakan pada gambar jalan yang diproses [1].

3. HASIL DAN PEMBAHASAN

3.1 Pra-pemrosesan Data

Dalam proses pra-pemrosesan gambar untuk aplikasi pendeteksi keretakan jalan, langkah awal melibatkan penggunaan fungsi pemuatan gambar untuk mengumpulkan gambar dari direktori yang telah ditentukan, yang terpisah menjadi tiga kategori: jalan retak, retak berlubang dan jalan tidak rusak. Gambar-gambar ini kemudian diubah menjadi grayscale dan diubah ukurannya menjadi 100x100 piksel untuk mengurangi kompleksitas dan memastikan konsistensi ukuran, yang sangat penting untuk pemrosesan oleh model Convolutional Neural Network (CNN). Selanjutnya, gambar diberi label berdasarkan folder asalnya, dengan label 1 untuk jalan retak, 0 untuk jalan tidak rusak, dan 2 untuk jalan retak berlubang, memfasilitasi identifikasi kelas selama pelatihan dan pengujian model.

Setelah proses labelisasi, data dibagi menjadi set pelatihan dan pengujian menggunakan fungsi `train_test_split` dari scikit-learn, dengan proporsi yang telah ditentukan untuk memastikan bahwa model dapat diuji dengan data yang independen. Normalisasi dilakukan dengan membagi nilai piksel gambar dengan 255.0, mengubahnya dari skala 0-255 menjadi 0-1, yang membantu dalam mempercepat konvergensi selama pelatihan. Akhirnya, data gambar diubah bentuknya untuk menambahkan dimensi channel yang diperlukan, menyesuaikan data menjadi bentuk (100, 100, 1) yang sesuai dengan input yang diharapkan oleh arsitektur CNN, memastikan bahwa setiap aspek dari pra-pemrosesan dirancang untuk mendukung efisiensi dan efektivitas model dalam mendeteksi keretakan jalan.

3.2 Pelatihan Model

Arsitektur model ini meliputi tiga lapisan konvolusional dengan jumlah filter yang meningkat, 32, 64, dan 128, masing-masing menggunakan kernel ukuran 3x3 dan fungsi aktivasi ReLU, diikuti oleh lapisan MaxPooling2D untuk mengurangi dimensi data setelah setiap lapisan konvolusi. Setelah lapisan konvolusional, data di-flatten menjadi vektor satu dimensi dan di-reshape untuk memasukkan ke dalam lapisan LSTM yang memiliki 50 unit, yang kemudian diikuti oleh dense layer dengan tiga unit dan fungsi aktivasi softmax untuk klasifikasi tiga kelas yaitu jalan retak, tidak retak, dan retak berlubang.

Untuk konfigurasi pelatihan, model menggunakan optimizer Adam karena efisiensinya yang tinggi dan kemampuan adaptasi dengan data yang bervariasi, serta menggunakan Sparse Categorical Crossentropy sebagai loss function untuk menangani tugas klasifikasi multi kelas. Metrik yang digunakan untuk mengevaluasi performa model adalah accuracy. Pelatihan dilakukan selama 100 epochs dengan batch size sebanyak 32 sampel per update. Berikut ini adalah tabel ringkasan arsitektur model yang menyediakan detail struktur setiap lapisan, jumlah parameter yang dapat dilatih, dan total parameter :

Table 3. Ringkasan Model

Layer	Bentuk Output	Jumlah Parameter
Conv2D	(None, 98, 98, 32)	320
MaxPooling2D	(None, 49, 49, 32)	0
Conv2D	(None, 47, 47, 64)	18496
MaxPooling2D	(None, 23, 23, 64)	0
Conv2D	(None, 21, 21, 128)	73856
MaxPooling2D	(None, 10, 10, 128)	0
Flatten	(None, 12800)	0
Reshape	(None, 1, 12800)	0
LSTM	(None, 50)	2570200
Dense	(None, 3)	153

Tabel ringkasan model menunjukkan bahwa arsitektur kombinasi Convolutional Neural Network (CNN) dan Long Short-Term Memory (LSTM) yang digunakan dalam penelitian ini memiliki total 2.661.923 parameter yang dapat dilatih. Layer CNN terdiri dari tiga lapisan konvolusi yang diikuti oleh lapisan pooling, yang secara bertahap mengurangi dimensi data dari 98x98x32 menjadi 10x10x128, sambil mengekstraksi fitur penting dari gambar input. Layer Flatten mengubah data menjadi vektor 1D yang kemudian diubah bentuknya untuk dimasukkan ke dalam layer LSTM, yang memiliki 50 unit untuk menangkap informasi sekuensial dari fitur-fitur yang diekstraksi. Layer terakhir adalah Dense layer dengan tiga unit untuk klasifikasi kelas. Arsitektur ini dirancang untuk menggabungkan kemampuan CNN dalam ekstraksi fitur visual dan kemampuan LSTM dalam memproses informasi sekuensial.

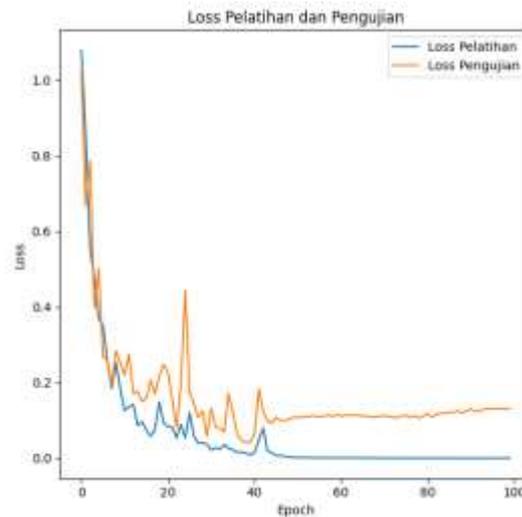
3.3 Hasil Evaluasi Model

Table 4. Akurasi Berdasarkan Proporsi

Data Latih	Data Uji	Akurasi	Presentase Akurasi
90%	10%	0,9459	94,59%
80%	20%	0,9728	97,28%
70%	30%	0,9276	92,76%

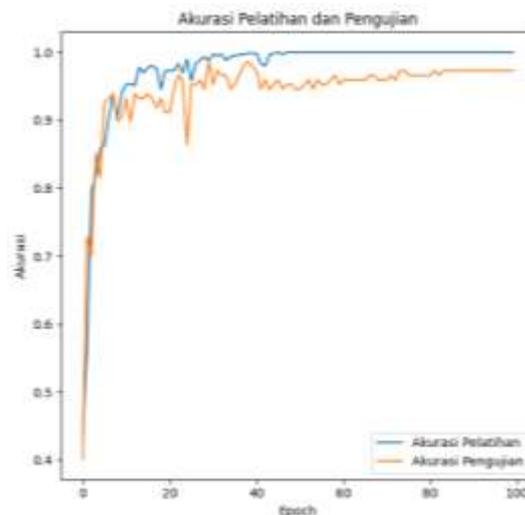
Tabel di atas menampilkan akurasi model berdasarkan tiga skema pembagian data yang berbeda. Untuk set data latih dan uji dengan proporsi 90%/10%, akurasi yang diperoleh adalah 94,59%. Ketika proporsi pelatihan diubah menjadi 80% dan data uji ditingkatkan menjadi 20%, akurasi meningkat menjadi 97,28%. Namun, dengan penurunan proporsi pelatihan lebih lanjut menjadi 70% dan peningkatan proporsi pengujian menjadi 30%, akurasi sedikit menurun menjadi 92,76%. Dengan ini, model

yang akan digunakan adalah model dengan proporsi 80%/20% karena memiliki akurasi tertinggi.



Gambar 5. Grafik Nilai Loss Pelatihan Model

Grafik diatas menggambarkan kurva loss untuk data pelatihan dan pengujian selama 100 epoch. Terlihat bahwa loss pelatihan (garis biru) dan loss pengujian (garis oranye) menurun dengan cepat selama 20 epoch pertama, menunjukkan bahwa model belajar dengan baik dari data pelatihan. Setelah sekitar 20 epoch, loss pelatihan terus menurun mendekati nol, sementara loss pengujian mulai stabil dan berfluktuasi di sekitar nilai yang lebih rendah, tanpa peningkatan yang signifikan. Hal ini menunjukkan bahwa model mencapai konvergensi dan performa yang optimal, dengan tidak adanya tanda-tanda overfitting.



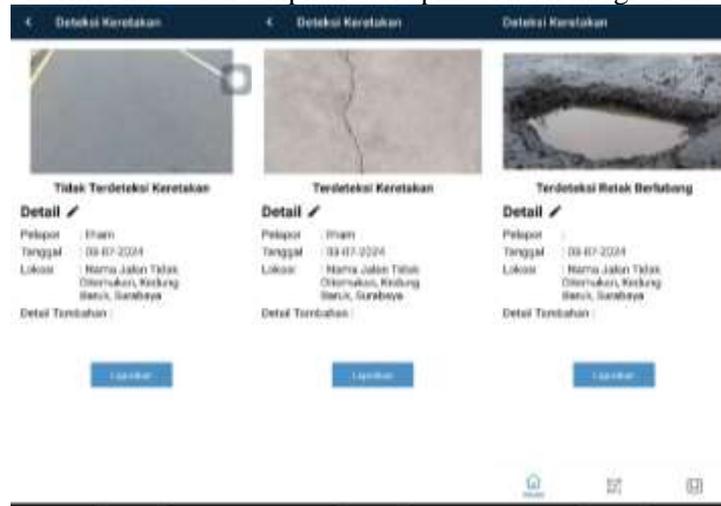
Gambar 6. Grafik Nilai Akurasi Pelatihan Model

Grafik yang ditampilkan menunjukkan kurva akurasi untuk data pelatihan dan pengujian selama 100 epoch. Kurva akurasi pelatihan (garis biru) meningkat dengan cepat dalam 20 epoch pertama dan terus meningkat mendekati 1,0 (100%) seiring dengan bertambahnya epoch. Ini menunjukkan bahwa model belajar untuk mengklasifikasikan data pelatihan dengan sangat baik. Akurasi pengujian (garis oranye) juga meningkat tajam pada awal pelatihan, mencapai nilai yang tinggi mendekati 1,0 setelah sekitar 40 epoch dan kemudian stabil. Stabilitas kurva akurasi pengujian setelah mencapai nilai tinggi menunjukkan bahwa model mempertahankan kinerja yang baik pada data yang tidak terlihat, dengan kemampuan generalisasi yang kuat dan tidak menunjukkan tanda-tanda overfitting yang signifikan. Kedua kurva ini menunjukkan bahwa model berhasil

belajar dan melakukan klasifikasi dengan akurasi tinggi baik pada data pelatihan maupun data pengujian.

3.4 Hasil Integrasi Android

Setelah menyelesaikan proses pelatihan dan memastikan bahwa model memiliki akurasi yang memuaskan, model tersebut disimpan dalam format HDF5 menggunakan fungsi save dari library Keras. Format HDF5 memfasilitasi penyimpanan komprehensif termasuk arsitektur model, bobot, dan konfigurasi pelatihan, yang memudahkan proses pemuatan kembali model di masa mendatang. Setelah penyimpanan, model diubah menjadi format TensorFlow Lite (.tflite), memungkinkan integrasi yang lebih mudah ke dalam aplikasi Android, sehingga meningkatkan kinerja dan aksesibilitas model pada perangkat mobile. Berikut ini adalah tampilan dari aplikasi hasil integrasi model :



Gambar 7. Hasil Output Model dalam Aplikasi

Pada gambar pertama, aplikasi menunjukkan hasil "Tidak Terdeteksi Keretakan" untuk gambar jalan yang tidak memiliki retakan. Ini menunjukkan bahwa model berhasil mengenali kondisi jalan yang baik dan tidak rusak. Gambar kedua menunjukkan hasil "Terdeteksi Keretakan" untuk gambar jalan yang memiliki retakan. Model berhasil mendeteksi keretakan jalan tersebut, seperti yang terlihat dari retakan pada gambar. Fungsi untuk menambahkan detail lokasi otomatis dengan mengambil lokasi dari perangkat juga berjalan dengan baik.

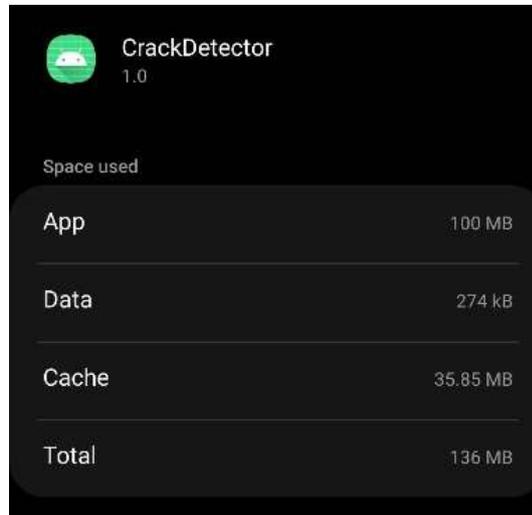
3.5 Analisis Kinerja Aplikasi

Analisis ini akan mencakup dua aspek penting yaitu kecepatan pemrosesan model dan ukuran dari aplikasi. Evaluasi kinerja aplikasi ini bertujuan untuk memastikan bahwa aplikasi tidak hanya akurat dalam mendeteksi keretakan jalan, tetapi juga efisien dan ramah pengguna dalam operasinya. Hasil analisis ini juga akan memberikan wawasan yang dapat digunakan untuk pengembangan atau perbaikan dan optimasi aplikasi.

Table 5. Kecepatan Pemrosesan

Fungsi	Peran Fungsi	Waktu Eksekusi
loadTFLiteModel	Memuat model	2,46 ms
convertToGrayscale	Konversi input menjadi grayscale	12,8 ms
detectCrack	Fungsi <i>preprocessing</i> data sekaligus memproses output model	90,3 ms

Dengan kecepatan yang dijabarkan dalam tabel diatas, aplikasi dapat melakukan deteksi keretakan pada jalan hampir secara instan dari sudut pandang pengguna. Waktu ini termasuk pemrosesan gambar, eksekusi model, dan pengambilan keputusan, yang semuanya dilakukan di bawah satu detik, sehingga tidak menyebabkan keterlambatan yang teras oleh pengguna. Secara keseluruhan, waktu eksekusi yang tercantum dalam tabel menunjukkan bahwa semua fungsi berjalan dengan cepat, memenuhi kebutuhan aplikasi untuk performa real-time.



CrackDetector 1.0	
Space used	
App	100 MB
Data	274 kB
Cache	35.85 MB
Total	136 MB

Gambar 8. Detail Ukuran Aplikasi

Gambar di atas menunjukkan penggunaan ruang penyimpanan oleh aplikasi yang dikembangkan dalam penelitian ini. Ukuran aplikasi sebesar 136 MB cukup wajar mengingat aplikasi ini mencakup model TensorFlow Lite yang memerlukan ruang penyimpanan yang signifikan. Dengan ukuran aplikasi utama sebesar 100 MB, aplikasi ini masih dapat dianggap efisien dan tidak terlalu membebani penyimpanan perangkat, terutama mempertimbangkan fungsionalitasnya. Data dan cache yang relatif kecil menunjukkan bahwa aplikasi mengelola data sementara dengan baik tanpa mengakibatkan penggunaan penyimpanan yang berlebihan

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan mengenai pengembangan aplikasi pendeteksi keretakan jalan berbasis Android menggunakan implementasi algoritma Hybrid CNN-LSTM, dapat disimpulkan beberapa hal sebagai berikut:

- Model Hybrid CNN-LSTM yang diterapkan menunjukkan hasil yang baik dengan akurasi deteksi mencapai 97.5% pada data pengujian, sedangkan model CNN saja menghasilkan akurasi sebesar 92.76%. Hal ini menunjukkan bahwa model hybrid mampu mengenali keretakan jalan dengan tingkat keakuratan yang lebih tinggi.
- Evaluasi kinerja aplikasi menunjukkan bahwa waktu eksekusi fungsi-fungsi penting berada dalam rentang yang sangat cepat untuk aplikasi real-time, dengan waktu eksekusi deteksi keretakan sekitar 90.3 milidetik. Ukuran aplikasi juga cukup kecil, yaitu 136 MB.
- Aplikasi yang dikembangkan melalui integrasi model menggunakan TensorFlow Lite mampu mendeteksi keretakan jalan dengan memanfaatkan kombinasi algoritma CNN untuk ekstraksi fitur visual dan LSTM untuk analisis sekuensial dari data gambar.

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, beberapa saran yang dapat diberikan untuk penelitian lebih lanjut dan pengembangan aplikasi pendeteksi keretakan jalan mencakup perlunya pengujian dan pelatihan lebih lanjut pada

berbagai kondisi lingkungan, seperti pencahayaan yang buruk atau adanya hambatan visual, untuk memastikan keandalan model dalam berbagai situasi. Penggunaan dataset yang lebih besar dan lebih bervariasi dari berbagai wilayah geografis dapat membantu meningkatkan performa dan generalisasi model. Penambahan fitur seperti pelaporan otomatis ke pihak berwenang dapat meningkatkan kegunaan aplikasi bagi pengguna dan pihak berwenang dalam pemeliharaan jalan. Kolaborasi dengan pemerintah daerah dan instansi terkait dapat mendukung implementasi aplikasi di lapangan dan menyediakan umpan balik yang komprehensif untuk perbaikan lebih lanjut.

5. DAFTAR RUJUKAN

- [1] Gill, H. S. and Khehra, B., S., 2021. An integrated approach using CNN RNN-LSTM for classification of fruit images. *Materials Today: Proceedings*, 51, pp591-595.
- [2] Islam, M.Z., Islam, M.M. and Asraf, A., 2020. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked*, 20, 100412.
- [3] Nguyen, N.H.T, Perry, S., Bone, D., Le, H.T., and Nguyen, T.T., 2021. Two-Stage Convolutional Neural Network for Road Crack Detection and Segmentation. *Expert Systems With Applications*, 186, 115718.
- [4] Sainath, T. N., Vinyals, O., Senior, A., and Sak, H., 2015. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp4580-84.
- [5] Moroney, L., 2018. *Using TensorFlow Lite on Android*. [Online] Available at: <https://blog.tensorflow.org/2018/03/using-tensorflow-lite-onandroid.html> [Accessed 18 March 2024]