

Full Paper

Design of the "SRIKANDI" Image Processing Application for Detecting Rice Leaf Diseases using the NASNetMobile Convolutional Neural Network Architecture

Dodi Bayu Prasanto¹,Telkom University, Banyumas, Indonesia,
dodibayup@student.telkomuniversity.ac.idFauzaandhiyaa Shafiananto²,Telkom University, Banyumas, Indonesia,
fauzaandhiyaa@student.telkomuniversity.ac.idMuhammad Rakan Atha³,Telkom University, Banyumas, Indonesia,
rakanatha@student.telkomuniversity.ac.idSeptiyadi Susanto⁴,Telkom University, Banyumas, Indonesia,
septiyadi@student.telkomuniversity.ac.idPutri Anggrani Sijabat⁵,Telkom University, Banyumas, Indonesia,
putrianggrani@student.telkomuniversity.ac.idCindy anti⁶,Telkom University, Banyumas, Indonesia,
cindyanti@student.telkomuniversity.ac.idDany Candra Febrianto^{7*},Telkom University, Banyumas, Indonesia,
danycandra@telkomuniversity.ac.id

ABSTRACT

Rice leaf diseases can cause a significant decrease in productivity if not treated early, while currently still using a manual diagnosis process that is often slow, inconsistent, and dependent on extension workers. In this study, the SRIKANDI application was developed, a mobile application for image processing for rice leaf diseases using the NASNetMobile Convolutional Neural Network (CNN) architecture. This system is designed using five labels, namely bacterial, blast, brownspot, leafsmut, and healthy leaves. The dataset used consists of 2500 images collected from Kaggle, Mendeley Data, and taken directly. All images go through preprocessing stages of resizing, pixel normalization, and augmentation, then divided into 80% train, 10% test, and 10% validation. The model training was carried out in two stages, namely, 40 epochs of fine-tuning with a learning rate of 0.0008 followed by 20 epochs of fine-tuning with a learning rate of 1e-5, the results obtained by the model with a test set accuracy rate of 96.40%. The trained model is then saved in TFLite format to be integrated into the SRIKANDI mobile application so that it can help farmers detect rice leaf diseases in real-time via camera or taken from the gallery.

KEYWORDS

NASNetMobile; Convolutional Neural Network (CNN); Rice Leaf Disease Detection; Mobile Application

Prasanto. D. B., Shafiananto. F, Atha. M. R., Susanto. S., Sijabat. P. A., Anti. C., Febrianto. D. C. (2026). Design of the "SRIKANDI" Image Processing Application for Detecting Rice Leaf Diseases using the NASNetMobile Convolutional Neural Network Architecture. *jasmed*, 4(1), pp. 49-61. <https://doi.org/10.25124/jasmed.v4i1.10235>

Article Submitted 04/12/2025. Revision uploaded 26/03/2026. Accepted 27/03/2026.

© 2026 by the authors of this article. Published under CC-BY 

1. INTRODUCTION

Tanaman padi menghasilkan beras yang merupakan makanan utama yang biasa dimakan di Indonesia, sehingga hal yang sangat penting adalah keberhasilan dalam jumlah padi yang dipanen. Keterlambatan dalam proses diagnosis secara manual dapat menyebabkan penyakit pada tanaman padi berkembang hingga tingkat yang serius, yang berdampak pada gagal panen akibat kurangnya pengetahuan petani serta anggapan bahwa gejala yang muncul pada daun padi adalah hal yang normal selama masa tanam [1]. Alih-alih hanya melakukan pengendalian setelah serangan hama terjadi, lebih baik untuk mengambil langkah-langkah pencegahan [2].

Mengidentifikasi penyakit yang mungkin dihadapi dan mengenali gejala-gejalanya. Penyuluh pertanian masih dihadapkan pada tantangan karakteristik bahwa meskipun telah melakukan perubahan pada tanaman itu dimungkinkan bahwa penyuluh tetap dalam negeri lain sejak mulai bisa mengenali penyakit yang menyebabkan penyakit pada padi sehingga tidak mungkin menciptakan cara yang baik untuk menyelidiki penyakit dan memberikan penyuluhan atau metode untuk mendampinginya. Oleh karena itu, penyuluhan telah sulit menjelaskan pada para petani tentang gejalanya [3], [4].

Artificial Intelligence membentuk otak untuk pengembangan teknologi di setiap halaman, otak *AI* membantu untuk memperkuat ketahanan tanaman pangan di bidang pertanian. Salah satu inovasi terbaru *AI* adalah penggunaan pengenalan objek melalui gambar atau citra di sektor pertanian. Pengenalan gambar atau objek digital adalah teknik yang sering digunakan, salah satunya adalah klasifikasi, dan menjadi salah satu metode utamanya, salah satunya adalah dengan menerapkan jaringan saraf tiruan [5]. *Deep Learning*, sejauh ini, telah menjadi salah satu alat utama untuk memprediksi peluang, mengenali objek, dan mendiagnosis penyakit. Sistem pemrosesan citra dirancang untuk membantu mengenali, mengidentifikasi, dan mengklasifikasikan objek citra dengan akurasi, kecepatan, dan efisiensi tinggi, bahkan ketika menangani data dalam jumlah besar. Metode jaringan saraf tiruan konvolusional dalam pemrosesan citra merupakan salah satu teknik yang paling banyak digunakan karena tingkat akurasi yang sangat tinggi [6].

Convolutional Neural Network (CNN) sangat ideal untuk menjalankan proses pengolahan gambar. Lapisan konvolusi ini adalah bagian dari jaringan saraf konvolusional yang terdiri dari berbagai filter yang dipadukan dengan gambar dan menerima masukan dari lapisan konvolusi yang ada sebelumnya, sehingga terhubung secara menyeluruh ke *output layer* [7]. Proses ini mirip dengan bagaimana *neuron* di otak merespons untuk mengaktifkan bagian tertentu dari tubuh organisme [8].

Salah satu model paling efisien untuk klasifikasi gambar adalah *NASNetMobile*, dan telah umum digunakan pada perangkat akhir berdaya komputasi rendah seperti ponsel pintar. Arsitektur *NASNetMobile* didasarkan pada metode *Neural Architecture Search (NAS)* yang merupakan metode pencarian otomatis terbaik dengan parameter lebih sedikit untuk performa tinggi [9]. Meskipun sedikit lebih sederhana daripada beberapa arsitektur *CNN* saat ini, *NASNetMobile* tetap memiliki performa yang unggul dalam tugas klasifikasi gambar, sehingga masih banyak digunakan pada aplikasi android dengan transfer *learning* [10].

Banyak petani sangat kesulitan untuk mengenali penyakit pada padi karena sebagian besar petani mengandalkan pengetahuan tradisional, dan gejala berbagai penyakit serupa selain itu, petani membutuhkan waktu dan tenaga ahli khusus untuk mengidentifikasi penyakit tanaman secara manual. Hal ini menyulitkan diagnosis di

lapangan di mana pengukuran tersebut tidak memungkinkan. Memanfaatkan tren *AI* tersebut, dengan perkembangan teknologi *deep learning*, penelitian ini berupaya membangun sistem deteksi penyakit padi dengan menggunakan gambar daun dengan mengadopsi metode *CNN* dan arsitektur *NASNetMobile* agar proses klasifikasi penyakit dilakukan secara otomatis dan efisien [11]. Lebih lanjut, penelitian ini juga mengembangkan aplikasi android sederhana yang mudah digunakan oleh pemilik pertanian dan penyuluh untuk mendiagnosis penyakit daun di lapangan dari foto.

Meskipun penelitian deteksi penyakit daun padi berbasis *deep learning* telah banyak yang dilakukan, masih terdapat *research gap* terutama pada aspek implementasi model yang ringan dan efisien untuk pengembangan perangkat mobile serta integrasi ke dalam aplikasi yang mudah digunakan di lapangan. Sebagian besar penelitian masih berfokus pada akurasi tanpa mempertimbangkan keterbatasan komputasi dan kebutuhan pengguna.

Oleh karena itu, penelitian ini berkontribusi dengan mengembangkan sistem deteksi penyakit daun padi berbasis citra menggunakan metode *CNN* dengan arsitektur *NASNetMobile* yang ringan dan efisien. Model ini diintegrasikan kedalam aplikasi *android* untuk memungkinkan diagnosis penyakit secara cepat, praktis, dan dapat digunakan langsung oleh petani sebagai upaya deteksi dini mencegah risiko gagal panen.

2. METHODS

2.1 Alur Penelitian

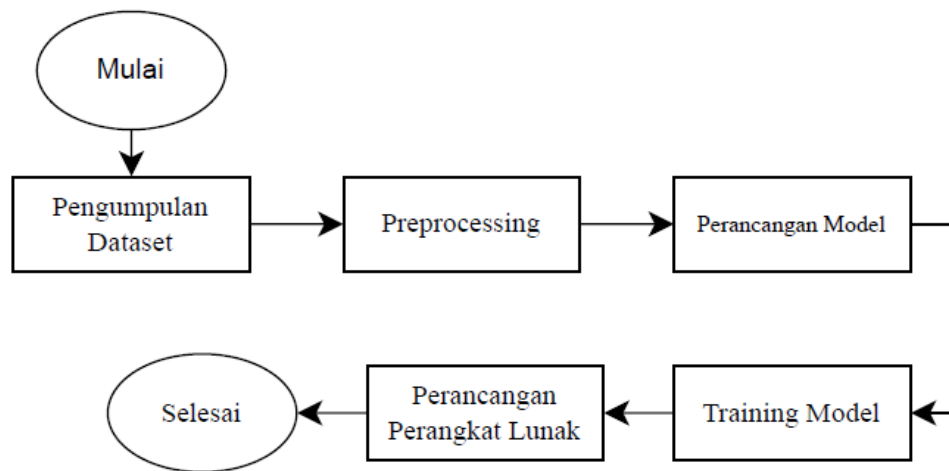


Figure 1 Alur Penelitian.

Alur penelitian dalam riset ini adalah sebagai berikut. Pertama, mengumpulkan *dataset* gambar yang nantinya akan digunakan sebagai bahan latihan model. Lalu, melakukan tahap *preprocessing* untuk memastikan konsistensi, seperti penyesuaian ukuran gambar. Tahap berikutnya fokus pada pembangunan arsitektur model yang akan digunakan. Selanjutnya, tahap pembangunan arsitektur model yang akan digunakan. Kemudian, melatih arsitektur dengan data gambar yang telah di *preprocessing* agar model mampu membedakan jenis penyakit yang menyerang daun padi sehingga model mampu memprediksi dengan akurat. Terakhir adalah membangun aplikasi sederhana yang memungkinkan para petani memanfaatkan model tersebut untuk mendiagnosis kondisi kesehatan tanaman padi mereka di lapangan.

2.2 Pengumpulan *Dataset*

Dalam penelitian ini *dataset* yang digunakan adalah gambar penyakit daun padi dan daun sehat. *Dataset* ini harus beragam untuk mencakup berbagai aspek seperti pencahayaan, pengambilan gambar dan elemen lainnya yang ada

pada gambar. Setelah *dataset* terekumpul, langkah selanjutnya adalah melakukan *preprocessing* pada gambar yang sudah terkumpul sebelumnya. *Preprocessing* ini meliputi pengubahan ukuran gambar agar konsisten, normalisasi *pixel*, dan *augmentasi* untuk meningkatkan jumlah data latih. Pengumpulan *dataset* ini diambil dari *kaggle*, *mandeley* data, dan diambil secara langsung. Penelitian ini menggunakan *dataset* yang terdiri dari 2500 gambar, yang dibagi menjadi empat daun yang terkena penyakit yaitu *bakterial*, *blast*, *brownspot*, dan *leafsmut* serta satu daun sehat (masing-masing 500 foto). *Dataset* ini dibagi menjadi 80% *train*, 10% *test*, dan 10% *validation*, sehingga terdapat 2000 gambar untuk *train*, 250 gambar untuk *test*, dan 250 gambar untuk *valid*.



Figure 2 *Bakterial*.



Figure 3 *Blast*.

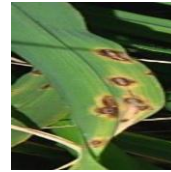


Figure 4 *Brownspot*.



Figure 5 *Leafsmut*.

2.3 Preprocessing

Tahap preprocessing dilakukan untuk mempersiapkan data citra agar memiliki format yang seragam dan optimal untuk pelatihan model *CNN*. Pada penelitian ini, seluruh gambar dalam *dataset* digunakan tanpa ada yang dibuang, karena data telah melalui seleksi awal dan untuk menjaga jumlah data tetap memadai. Selain itu, untuk meningkatkan kemampuan generalisasi model, digunakan teknik data augmentation daripada mengurangi data.

Proses preprocessing meliputi beberapa tahapan, yaitu resizing gambar menjadi 224×224 piksel agar sesuai dengan input model *NASNetMobile*, serta rescaling nilai piksel dari 0–255 menjadi 0–1 untuk mempercepat dan menstabilkan proses pelatihan. Selanjutnya dilakukan data augmentation berupa rotasi, zoom, pergeseran horizontal dan vertikal, shear transformation, flip horizontal dan vertikal, serta penyesuaian kecerahan. Tahapan ini bertujuan untuk menambah variasi data sehingga model lebih robust terhadap berbagai kondisi seperti perubahan sudut pengambilan gambar dan pencahayaan di lapangan.

2.4 Perancangan Model

Penelitian ini menggunakan *Convolutional Neural Network (CNN)* melalui arsitektur *NASNetMobile*. *Dataset* yang terdiri dari gambar daun padi dibagi sedemikian rupa sehingga 80% digunakan untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Untuk mengatasi fenomena *overfitting*, *augmentasi* data diterapkan melalui penambahan berbagai teknik manipulasi data seperti *rotasi*, *zooming*, pergeseran, *shearing*, *flipping*, dan normalisasi piksel. Arsitektur model yang disesuaikan dibentuk dengan mengambil *NASNetMobile* sebagai ekstraktor fitur tanpa lapisan atas, dan kemudian menambahkan lapisan lain yang sepenuhnya disesuaikan untuk tugas klasifikasi. Model tersebut dilatih menggunakan pendekatan dua langkah yang terdiri dari fase *preprocessing* yang menerapkan *freeze layer*, dan fase *fine-tuning* yang didefinisikan dengan membuka lapisan-lapisan di *NASNetMobile*. Pelatihan memanfaatkan *optimizer* adam dengan *categorical crossentropy* model dilatih selama 60 *epoch* dengan menerapkan *early-stopping*. Model ini menghasilkan keluaran multi-kelas untuk klasifikasi berbagai penyakit pada daun tanaman padi. Model tersebut kemudian dikonversi ke format *tf lite*, untuk diterapkan pada aplikasi mobile.

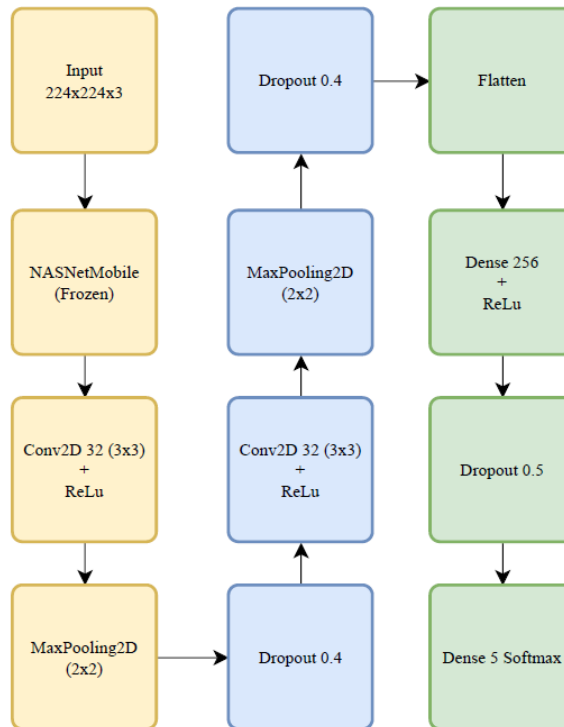


Figure 6 Diagram Perancangan Model.

2.5 Training Model

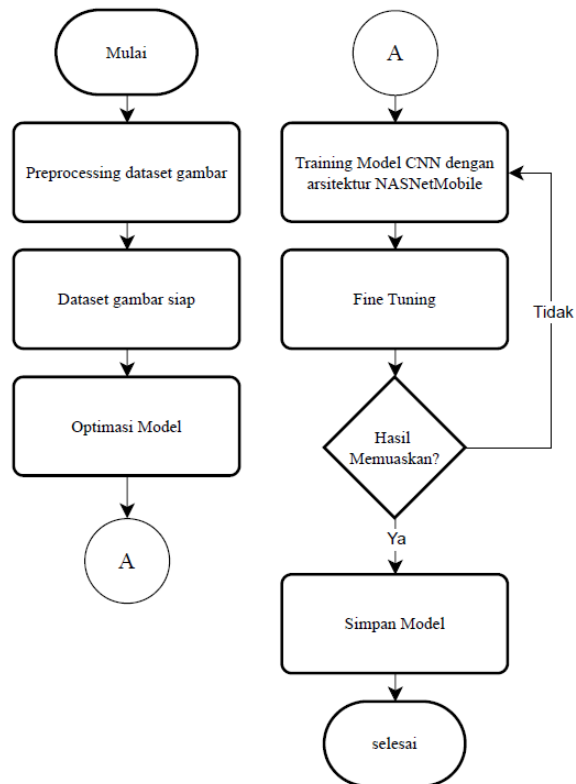


Figure 7 Rancangan Training Model.

Proses ini meliputi *preprocessing dataset* gambar serta menyiapkan data untuk pelatihan. Selanjutnya, *dataset* yang sudah siap masuk ke tahap optimasi model. Selanjutnya, model dilatih menggunakan arsitektur *CNN NASNetMobile* dan kemudian dilakukan *fine-tuning* model untuk menghasilkan model yang lebih berkualitas. Evaluasi model dilakukan, jika performa model belum memenuhi standar, maka kembali ke tahap *training* untuk dilakukan perbaikan. Begitu seterusnya hingga mencapai model yang maksimal. jika hasilnya sudah memuaskan, model yang sudah dilatih akan disimpan dalam format *tf lite*.

2.6 Perancangan Perangkat Lunak

Perancangan perangkat lunak diperlukan sebelum proses pembangunan sistem dilakukan, agar program yang dibangun, terstruktur dengan baik dan memenuhi kebutuhan pengguna. Dalam perancangan sistem, perancangan perangkat lunak adalah bagian penting yang menentukan perkembangan sistem [12].

2.6.1 Use Case Diagram

Use case diagram merupakan cara untuk menggambarkan bagaimana aktor (pengguna) dapat berinteraksi dengan suatu *system* untuk mencapai tujuan yang diinginkan. Use Case berfungsi untuk menjelaskan apa yang bisa dilakukan oleh *system*, bukan bagaimana *system* itu melakukannya [13]. Pada Gambar 4. *Use case diagram* ini menggambarkan interaksi pengguna dengan *system* aplikasi dengan fitur utama yaitu, *Home*, *Edukasi*, *Scan*, *Hasil*, *Informasi*, *Login*, dan *Register*. *User* dapat mengakses fitur *Home*, *Edukasi*, *Scan*, *Hasil*, *Informasi*, dimana setiap fitur bergantung pada proses *login* dengan melalui relasi *include*, menandakan bahwa pengguna harus login sebelum menggunakan fungsi tersebut. Sementara itu proses *register* dilakukan untuk membuat akun baru tetapi bersifat opsional karena ditandai sebagai *exclude*.

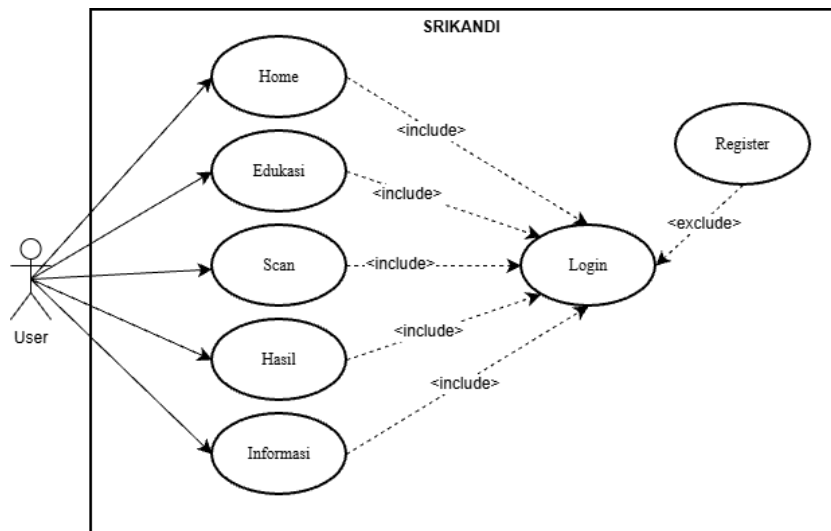


Figure 8 Use Case Diagram SRIKANDI.

2.6.2 Activity Diagram

Activity diagram merupakan salah satu jenis *diagram Unified Modelling Language (UML)* yang mengilustrasikan alur kerja atau aktivitas dari suatu sistem, proses bisnis, atau menu yang terdapat dalam perangkat lunak [14]. Sebelum menggunakan aplikasi, pengguna akan melihat *splash screen* dan *intro* lalu menekan tombol 'Mulai Sekarang' untuk masuk ke halaman *login*. Bagi pengguna yang tidak memiliki akun, akan diarahkan ke halaman *registrasi* untuk mengisi data kemudian *login* dengan *username* dan *password*. Setelah sistem memverifikasi, pengguna akan masuk ke halaman *home*. Di menu edukasi, pengguna dapat melihat berbagai jenis penyakit yang menyerang daun padi. Pengguna dapat memilih satu untuk melihat info lengkapnya. Di menu *scan*, sistem akan menampilkan menu untuk memilih menggunakan kamera atau *image gallery*. Jika perangkat menggunakan kamera, sistem akan meminta izin. Jika sudah disetujui pengguna dapat berfoto. Jika *gallery* dipilih, pengguna langsung dapat memilih gambar yang akan digunakan. Setelah sistem akan memproses gambar yang diambil dan menampilkan prediksi yang dihasilkan. Setelah melihat halaman hasil, pengguna dapat menekan tombol 'Lihat Selengkapnya' untuk mendalami lebih detail penyakit yang diidap, berapa akurasi pediksinya dan ringkasan

pencegahan dan pengobatan penyakitnya. Di halaman informasi, pengguna dapat memilih menu ‘Kenali Tim Kami’ atau ‘Penggunaan Aplikasi’ dan sistem akan menampilkan deskripsi tim atau panduan aplikasi berdasarkan menu yang dipilih pengguna.

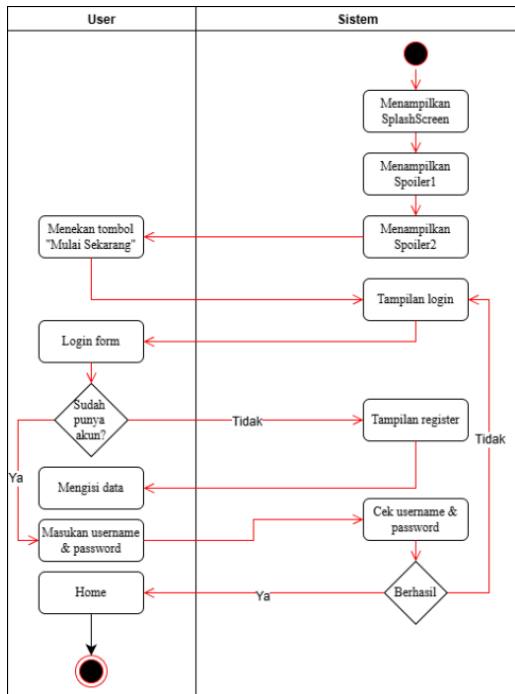


Figure 9 Activity Diagram Login dan Register.

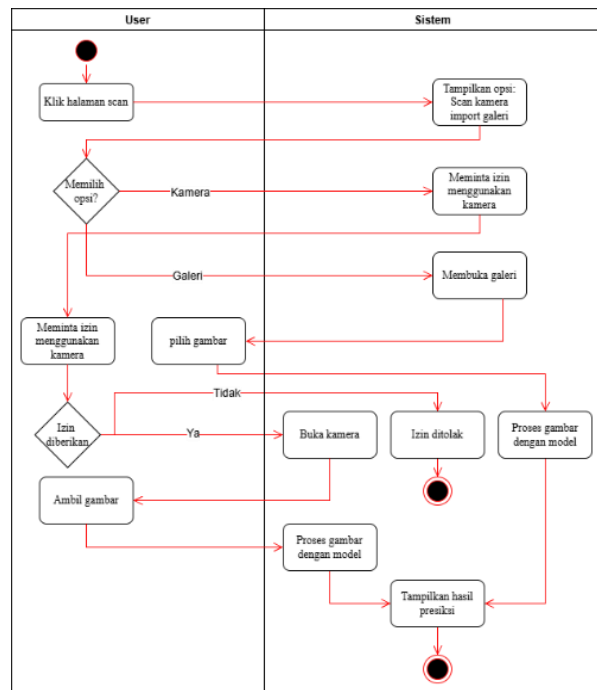


Figure 10 Activity Diagram Scan

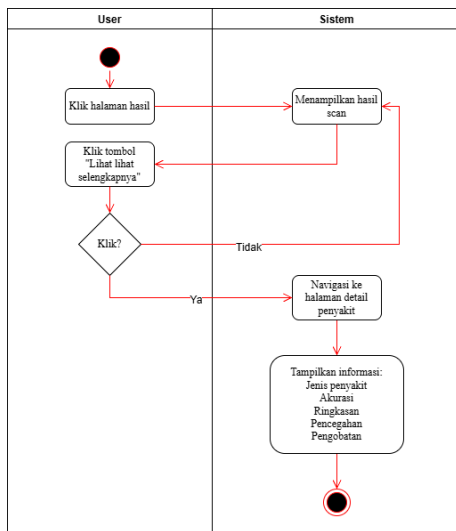


Figure 11 Activity Diagram Hasil.

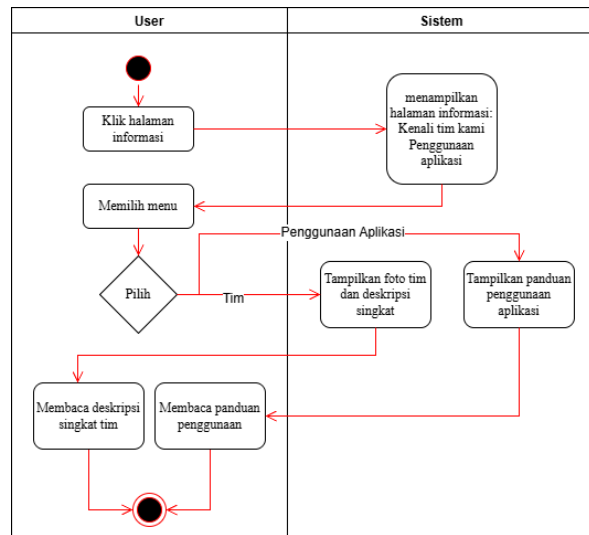


Figure 12 Activity Diagram Informasi.

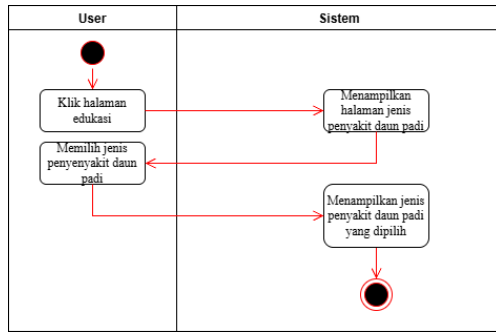


Figure 13 Activity Diagram Edukasi.

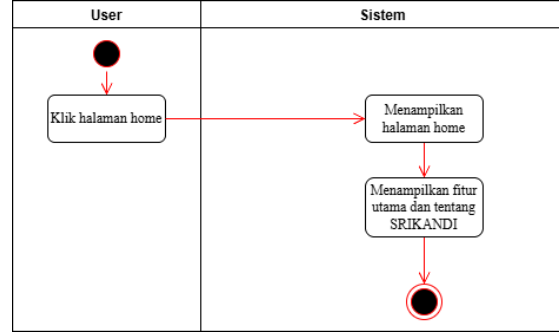


Figure 14 Activity Diagram Home.

2.6.3 Perancangan User Interface

User Interface (UI) terjadi ketika kita ingin menentukan bagaimana pengguna berinteraksi dan bernavigasi melalui sebuah aplikasi. elemen yang berintraksi dengan pengguna seperti tombol dan ikon, disebut objek. Tata letak halaman, tombol, ikon, menu, tipografi, dan pilihan warna merupakan elemen penting yang perlu dirancang pada tahap ini [15]. Pada tahapan ini pembuatan desain menggunakan tools figma, pada tahap ini prorotipe dibuat untuk memvisualisasikan alur interaksi sebelum masuk pada tahap pengembangan aplikasi.

3. RESULT

3.1 Hasil Pengumpulan Dataset

Hasil pengumpulan *dataset* dalam penelitian ini adalah berupa gambar penyakit daun padi yaitu bakterial, *blast*, *brownspot*, *leafsmut* dan daun sehat yang diperoleh dari *kaggle*, *mandeley* data, dan diambil secara langsung. Total terdapat 2500 foto bakterial, *blast*, *brownspot*, *leafsmut* dan daun sehat dengan jumlah masing masing 500 gambar. Kemudian dibuat folder dengan nama “*datasetdaunpadi*”.

3.2 Hasil Preprocessing Dataset

Setelah dilakukan tahap *preprocessing* pada dataset, seluruh citra telah berhasil disesuaikan ke dalam format yang seragam dan siap digunakan dalam proses pelatihan model. Hasil *preprocessing* menunjukkan bahwa semua gambar memiliki ukuran yang konsisten yaitu 224×224 piksel dengan nilai piksel yang telah dinormalisasi pada rentang 0–1. Selain itu, melalui penerapan data *augmentation*, diperoleh variasi citra yang lebih beragam dari sisi orientasi, posisi, skala, dan pencahayaan, sehingga dataset menjadi lebih representatif terhadap kondisi nyata di lapangan.

Tidak terdapat data yang dibuang dalam proses ini, sehingga seluruh dataset tetap digunakan untuk menjaga jumlah data tetap optimal. Hasil akhir *preprocessing* menghasilkan dataset yang lebih *robust* dan siap digunakan dalam proses pelatihan model *CNN*.

Selanjutnya, *dataset* dibagi menjadi tiga bagian, yaitu data *training* sebesar 80% (2000 gambar), data *validation* sebesar 10% (250 gambar), dan data *testing* sebesar 10% (250 gambar). Pembagian ini bertujuan untuk memastikan model dapat dilatih, divalidasi, dan diuji secara optimal.

3.3 Hasil Pelatihan Model

Setelah proses *preprocessing* selesai, model dilakukan pelatihan dengan dua tahap, pada tahap pertama *base* model *NASNetMobile* dilakukan *freeze layer* dan *fine-tuning* dengan membuka lapisan dari *base* model. Pada tahap pertama, model dilatih menggunakan 40 *epoch* *fine-tuning* dengan *learning rate* 0.0008, selanjutnya pada tahap *fine-tuning* model dilatih menggunakan 20 *epoch* dengan *learning rate* 1e-5. Pelatihan ini menggunakan *EarlyStopping* dan *ReduceLROnPlateau*. Model ini dilatih menggunakan *batch size* 32 didapat hasil akhir berikut.

Table 1 Hasil Training Model.

	Tahap 1 (Fine-Tuning 40 epoch)	Tahap 2 (Fine-Tuning 20 epoch)
Accuracy	97%	86%

Loss	10%	43%
Val Accuracy	97%	86%
Val Loss	12%	45%

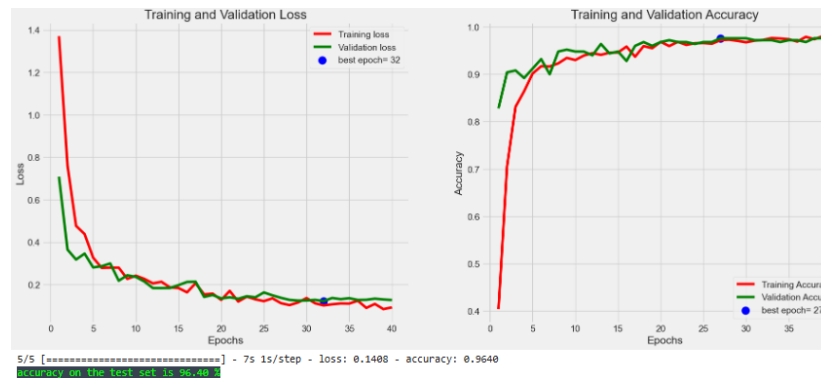


Figure 15 Grafik training validation loss dan accuracy.

Pada Gambar 15. Berdasarkan hasil pelatihan, pada tahap pertama (dengan freeze base model), model menunjukkan peningkatan akurasi yang cepat dan penurunan loss yang signifikan hingga mencapai kondisi stabil. Hal ini menandakan bahwa lapisan klasifikasi mampu memanfaatkan fitur yang telah diekstraksi oleh NASNetMobile dengan baik. Namun, pada tahap kedua (fine-tuning), terlihat adanya penurunan akurasi serta peningkatan nilai loss. Kondisi ini terjadi karena beberapa layer pada base model dibuka (unfreeze), sehingga bobot yang sebelumnya sudah optimal mengalami penyesuaian ulang. Proses ini dapat menyebabkan ketidakstabilan sementara karena model kembali menyesuaikan representasi fitur terhadap dataset yang digunakan.

Meskipun terjadi fluktuasi, pola antara training dan validation tetap menunjukkan tren yang sejalan tanpa adanya perbedaan yang signifikan, sehingga tidak mengindikasikan terjadinya overfitting atau underfitting yang serius. Hal ini diperkuat dengan hasil evaluasi pada data uji yang mencapai akurasi sebesar 96.40%, yang menunjukkan bahwa model tetap memiliki kemampuan generalisasi yang baik.

Setelah dilakukannya *training*, dan sudah mendapatkan hasil akurasi yang memuaskan, maka hasil dari *training* tersebut kita dapatkan model yang nantinya akan disimpan dalam format *file tflite*.

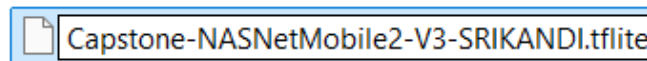


Figure 16 Grafik training validation loss dan accuracy.

Pada Gambar 16. Menunjukkan model dengan format *tflite* yang nantinya akan digunakan untuk daun padi yang terkena penyakit maupun yang sehat.

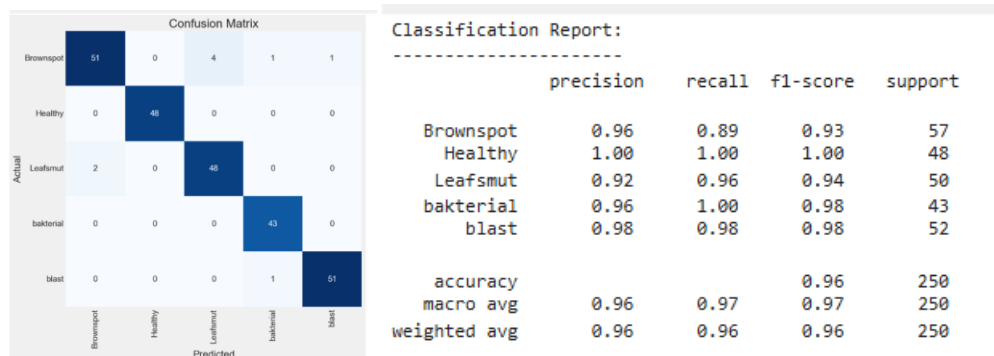


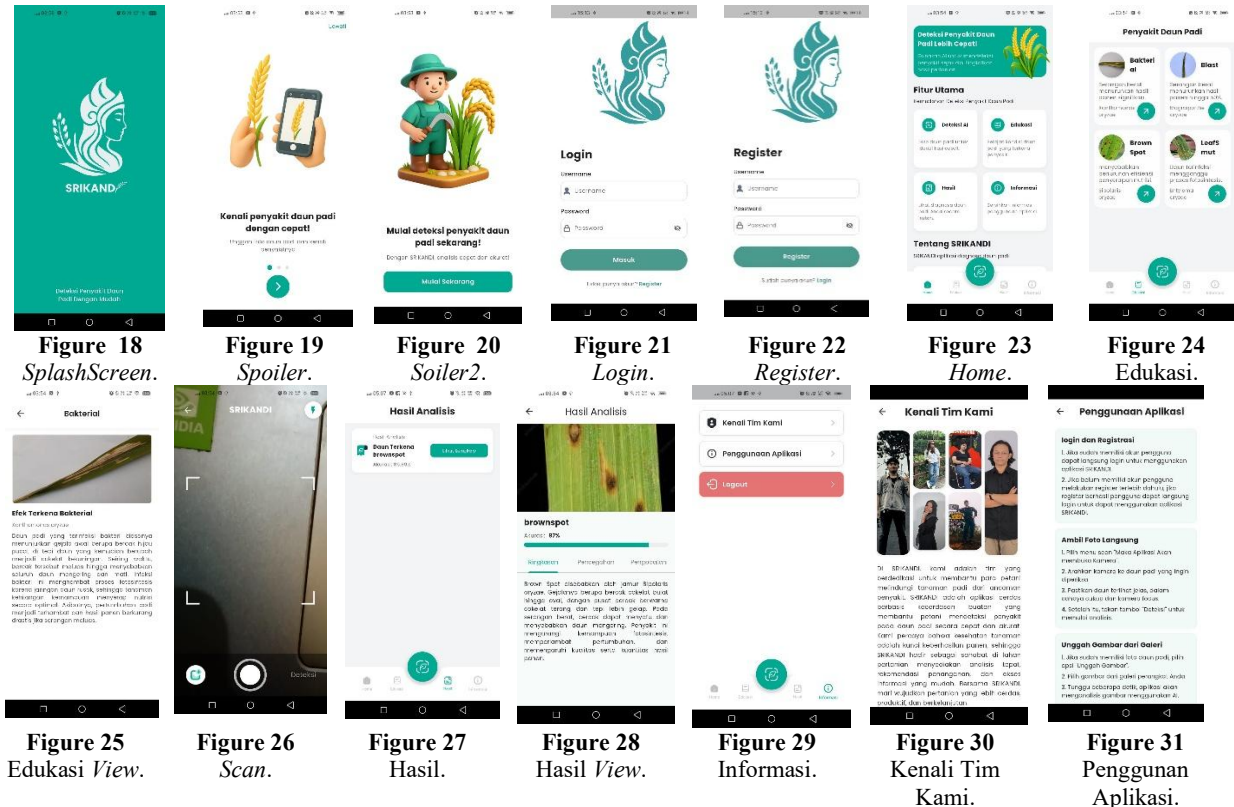
Figure 17 Confusion Matrix dan Classification Report.

Pada Gambar 17. Berdasarkan hasil pengujian menggunakan *confusion matrix*, terlihat bahwa sebagian besar prediksi model berada pada diagonal utama, yang menunjukkan bahwa model mampu mengklasifikasikan data

dengan baik pada setiap kelas. Kesalahan klasifikasi hanya terjadi pada beberapa sampel, terutama pada kelas *Brownspot* dan *Leafsmut*, namun jumlahnya relatif kecil dibandingkan total data uji. Selain itu, hasil *classification report* menunjukkan bahwa nilai *precision*, *recall*, dan *F1-score* untuk masing-masing kelas berada pada kisaran tinggi, yaitu antara 0.92 hingga 1.00, dengan rata-rata keseluruhan (*macro* dan *weighted average*) sebesar sekitar 0.96–0.97. Hal ini menunjukkan bahwa model tidak hanya memiliki akurasi tinggi, tetapi juga mampu mengenali setiap kelas secara seimbang. Dengan demikian, performa model dalam klasifikasi dapat dikatakan baik dan konsisten pada seluruh kategori penyakit daun padi.

3.4 Hasil Perancangan Perangkat Lunak

Berdasarkan *activity diagram* dan perancangan *UI* yang sudah dibuat, dihasilkan perangkat lunak berbasis mobile seperti gambar yang ada dibawah ini:



3.5 Hasil Deteksi Daun Padi

Setelah perangkat lunak berhasil dikembangkan, selanjutnya adalah dilakukan *testing* dengan lima *label* yang ada pada model yaitu *bakterial*, *blast*, *brownspot*, *leafsmut* dan daun sehat. *Testing* dilakukan dengan klik menu *scan* pada aplikasi dan *user* bisa melakukan foto langsung dan mengambil foto dari *galery* foto, setelah itu hasil deteksi akan muncul ke menu hasil dan *user* dapat melihat hasil dengan lengkap dengan klik tombol “Lihat Lengkap”.

Table 2 Hasil Deteksi Daun Padi.

Jenis Penyakit Daun Padi dan Daun Padi Sehat	Akurasi	Status
Bakterial	100%	Model berhasil mendeteksi jenis penyakit yang dialami daun padi dengan tingkat akurasi penyakit 100%
Blast	99.96%	Model berhasil mendeteksi jenis penyakit yang dialami daun padi dengan tingkat akurasi penyakit 99.96%

Brownspot	87%	Model berhasil mendeteksi jenis penyakit yang dialami daun padi dengan tingkat akurasi penyakit 87%
Leafsmut	98.92%	Model berhasil mendeteksi jenis penyakit yang dialami daun padi dengan tingkat akurasi penyakit 98.92%
Daun sehat	94.18%	Model berhasil mendeteksi jenis daun padi yang tidak terkena penyakit dengan tingkat akurasi penyakit 94.18%

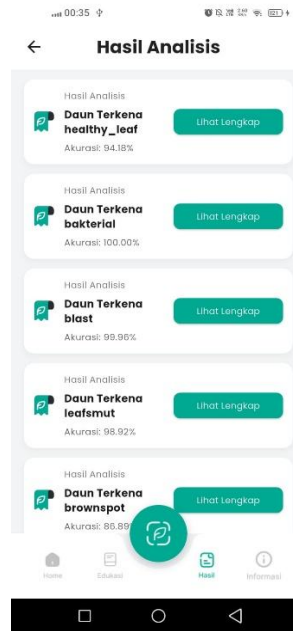


Figure 32 Hasil Deteksi Daun Padi.

4. DISCUSSION

Pada penelitian ini menunjukan arsitektur *NASNetMobile* dapat memberikan performa sangat baik dalam mendeteksi daun padi, dengan akurasi sebesar 96.40%. Dengan menerapkan *resize*, *rescale*, *rotation range*, *zoom range*, *width shift & height shift*, *shear range*, *horizontal flip & vertical flip*, dan *brightness range*. Meski dalam tahap *fine-tuning* kedua akurasi sedikit *fluktuatif*, namun performa keseluruhan tetap konsisten. Selanjutnya model akan di simpan dalam format *TFLite* untuk digunakan pada implementasi aplikasi *mobile* SRIKANDI yang dapat membantu petani dalam mendiagnosis penyakit daun padi secara cepat.

5. CONCLUSION

Penelitian ini mendapatkan hasil yang memuaskan dalam pengembangan aplikasi pengolahan citra untuk deteksi penyakit daun padi dengan menggunakan arsitektur *Convolutional Neural Network NASNetMobile*. *Dataset* yang digunakan terdiri dari 2500 gambar yang dibagi menjadi *bakterial*, *blast*, *brownspot*, *leafsmut* dan daun sehat, masing- masing label berisi 500 gambar daun padi. Pembagian data sebesar 80% *train*, 10% *test*, dan 10% *validation*. Pada tahap pelatihan model dibagi menjadi dua tahap yaitu, tahap pertama model dilatih menggunakan 40 *epoch fine-tuning* dengan *learning rate* 0.0008, selanjutnya tahap kedua *fine-tuning* 20 *epoch* dengan *learning rate* 1e-5. training tahap pertama menunjukan hasil *training* tahap pertama dengan *fine-tuning* menggunakan 40 *epoch* memperoleh *accuracy* sebesar 97%, *loss* sebesar 10%, *val_accuracy* sebesar 97%, dan *val_loss* sebesar 12%. Sedangkan *training* tahap kedua menggunakan *fine-tuning* yang memperoleh *accuracy* sebesar 86%, *loss* sebesar 43%, *val_accuracy* sebesar 86, dan *val_loss* sebesar 45% dengan hasil terbesbut didapat model dengan tingkat akurasi *test set* sebesar 96.40%, dan model disimpan sebagai Capstone-NASNetMobile-V3-SRIKANDI.tflite. Hasil perangkat lunak yaitu aplikasi SRIKANDI dapat berjalan sesuai yang diinginkan, aplikasi dapat melakukan deteksi pada daun

padi dengan masing masing labelnya, dimana untuk jenis penyakit bakterial mendapat akurasi sebesar 100%, *Blast* mendapat akurasi sebesar 99.96%, *Brownspot* mendapat akurasi sebesar 87%, *Leafsmut* mendapat akurasi sebesar 98.92%, sedangkan untuk daun sehat mendapat akurasi sebesar 94.18%. Hal tersebut menunjukan model yang sudah dibangun sesuai dengan hasik yang didapatkan.

6. REFERENCE

- [1] I. Journal, O. F. Latest, and T. In, "Optimizing Leafnet for Better Rice Leaf Disease Classification," vol. XIV, no. Viii, pp. 724–730, 2025, doi: 10.51583/IJLTEMAS.
- [2] H. K. K. Chinna Gopi Simhadri, "Automatic Recognition of Rice Leaf Diseases Using," 2023, doi: 10.3390/agronomy13040961.
- [3] M. N. Fawaiq, E. Utami, and D. Ariatmanto, "Rice Plant Disease Detection with Data Augmentation Using Transfer Learning International Journal of Research Publication and Reviews Rice Plant Disease Detection with Data Augmentation Using Transfer Learning," no. April, 2023, doi: 10.55248/gengpi.2023.4.4.35530.
- [4] D. J. Richter and K. Kim, "Assessing the performance of domain-specific models for plant leaf disease classification : a comprehensive benchmark of transfer-learning on open datasets," pp. 1–31, 2025, doi: 10.1038/s41598-025-03235-w 1.
- [5] R. Tiwari, J. Patel, N. R. Khan, and A. Dadhich, "automated identification of rice diseases," pp. 1–16, 2025, doi: 10.1371/journal.pone.0307461.
- [6] C. Gopi, H. Kishan, and V. Kumari, "Information Processing in Agriculture Deep learning for rice leaf disease detection : A systematic literature review on emerging trends , methodologies and techniques," *Inf. Process. Agric.*, vol. 12, no. 2, pp. 151–168, 2025, doi: 10.1016/j.inpa.2024.04.006.
- [7] A. Y. Ashurov, M. S. A. M. Al-gaashani, N. A. Samee, R. Alkanhel, and G. Atteia, "Enhancing plant disease detection through deep learning : a Depthwise CNN with squeeze and excitation integration and residual skip connections," no. January, pp. 1–16, 2025, doi: 10.3389/fpls.2024.1505857.
- [8] K. Syah, K. Mayya, I. F. Makarim, Z. Yudi, and A. Darva, "Implementasi Metode Convolutional Neural Network untuk Memilah Foto Pemandangan Pantai dan Pegunungan," vol. 2, no. 2, pp. 62–72, 2024, doi: 10.20895/jasmed.v2i2.1525.
- [9] I. G. Rian, L. Oka, A. Agung, G. Bagus, W. S. Peradhayana, and N. L. Wiwik, "Performance Comparison of MobileNetV2 and NASNetMobile Architectures in Soybean Leaf Disease Classification," vol. 6, no. 2, pp. 251–259, 2025, doi: 10.56705/ijodas.v6i2.243.
- [10] J. Padhi, K. Mishra, A. Kumar, S. Kumari, P. K. Sethy, and A. Nanthaamornphong, "Smart Agricultural Technology Enhancing paddy leaf disease diagnosis -a hybrid CNN model using simulated thermal imaging," *Smart Agric. Technol.*, vol. 10, no. February, p. 100814, 2025, doi: 10.1016/j.atech.2025.100814.
- [11] Y. K. Rathore, R. R. Janghel, C. Swarup, and S. K. Pandey, "Detection of rice plant disease from RGB and grayscale images using an LW17 deep learning model," vol. 31, no. December 2022, pp. 2813–2833, 2023, doi: 10.3934/era.2023142.
- [12] J. Ahamed, "A Decision-Making Technique for Software Architecture Design," vol. 9, no. 4, pp. 43–49, 2023, doi: 10.5815/ijmsc.2023.04.05.
- [13] V. Vranić *et al.*, "Use case modeling in a research setting of developing an innovative pilgrimage support system," *Univers. Access Inf. Soc.*, vol. 23, no. 4, pp. 1543–1560, 2024, doi: 10.1007/s10209-023-01047-1.
- [14] F. Siewe, "On the Execution and Runtime Verification of UML Activity Diagrams," pp. 1–42, 2025, doi: 10.3390/software4010004.
- [15] S. Syamala, A. Shrivastava, and S. Karmakar, "Social Sciences & Humanities Open Identifying interface design factors impacting user experience in digital learning platforms- A pilot study," *Soc. Sci. Humanit. Open*, vol. 11, no. May, p. 101662, 2025, doi: 10.1016/j.ssaho.2025.101662.

7. AUTHORS

Dodi Bayu Prasanto is a Bachelor of Science in Informatics Engineering student at Telkom University, Purwokerto Campus. His academic interests include Artificial Intelligence and Software Development. Email: dodibayup@student.telkomuniversity.ac.id.

Fauzaandhiyaa Shafiananto is a Bachelor of Science in Informatics Engineering student at Telkom University, Purwokerto Campus. His academic interests include Web and mobile Development. Email: fauzaandhiyaa@student.telkomuniversity.ac.id.

Muhammad Rakan Atha is a Bachelor of Science in Informatics Engineering student at Telkom University, Purwokerto Campus. His academic interests include Web and mobile Development. Email: rakanatha@student.telkomuniversity.ac.id.

Septiyadi Susanto is a Bachelor of Science in Informatics Engineering student at Telkom University, Purwokerto Campus. His academic interests include Database Systems and Software Engineering. Email: septiyadi@student.telkomuniversity.ac.id.

Putri Anggrani Sijabat is a Bachelor of Informatics Engineering student at Telkom University, Purwokerto Campus. Her academic interests include Software Testing. Email: putrianggrani@student.telkomuniversity.ac.id.

Cindy Anti is a Bachelor of Informatics Engineering student at Telkom University, Purwokerto Campus. Her academic interests include software testing. Email: cindyanti@student.telkomuniversity.ac.id.

Dany Candra Febrianto is Lecturer at the Informatics Program, Telkom University Kampus Purwokerto. His research and development interests include Artificial Intelligence and Software Engineering. Email: danycandra@telkomuniversity.ac.id.