

PERBANDINGAN MODEL ARSITEKTUR CNN DALAM KLASIFIKASI PENYAKIT PADA DAUN TANAMAN TOMAT

Ardian Syahputra¹, Anggi Hanafiah²

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Islam Riau

E-mail: ianbasilam@student.uir.ac.id¹, anggihanafiah@eng.uir.ac.id²

Article Info

Article history:

Dikirim 16 Juli 2025

Direvisi 16 Juli 2025

Dikirim 16 Juli 2025

Keyword:

Penyakit Daun Tomat, *CNN*,
VGG16, *VGG19*, *Xception*,
Klasifikasi Citra, *Deep*
Learning

ABSTRAK

Pertanian merupakan salah satu sektor vital dalam perekonomian, termasuk di Indonesia. Salah satu komoditas penting adalah tomat, yang permintaannya terus meningkat. Namun, tanaman tomat rentan terhadap berbagai jenis penyakit yang dapat menurunkan kualitas dan produktivitas. Untuk mengatasi masalah ini, dilakukan klasifikasi penyakit pada daun tomat menggunakan metode *Convolutional Neural Network (CNN)*. Penelitian ini membandingkan tiga arsitektur *CNN*, yaitu *VGG16*, *VGG19*, dan *Xception*. Model dilatih menggunakan dataset dari *website Kaggle* yang terdiri dari 4000 citra daun tomat, terbagi ke dalam empat kelas: *bacterial spot*, *late blight*, *yellow leaf curl virus*, dan daun tomat sehat. Pelatihan model dilakukan dengan kombinasi beberapa *hyperparameter*, seperti jumlah epoch, learning rate, dan optimizer. Untuk mempermudah proses pengujian, model diimplementasikan ke dalam aplikasi *web* berbasis *Flask*. Hasil pengujian menunjukkan bahwa model *VGG16* memberikan performa terbaik dengan akurasi 99,52%, disusul oleh *VGG19* dengan akurasi 98,08%, dan *Xception* dengan akurasi 97,12%. Penelitian ini menunjukkan bahwa arsitektur *CNN*, khususnya *VGG16*, efektif dalam mengklasifikasikan penyakit daun tomat secara akurat.

Copyright© 2025 IT Journal Research and Development.

All rights reserved..

Corresponding Author:

Anggi Hanafiah

Program Studi Teknik Informatika, Fakultas Teknik

Universitas Islam Riau

Jl. Kaharuddin Nst No.113, Simpang Tiga, Kecamatan. Bukit Raya, Kota Pekanbaru, Riau 28284

Email: anggihanafiah@eng.uir.ac.id

1. PENDAHULUAN

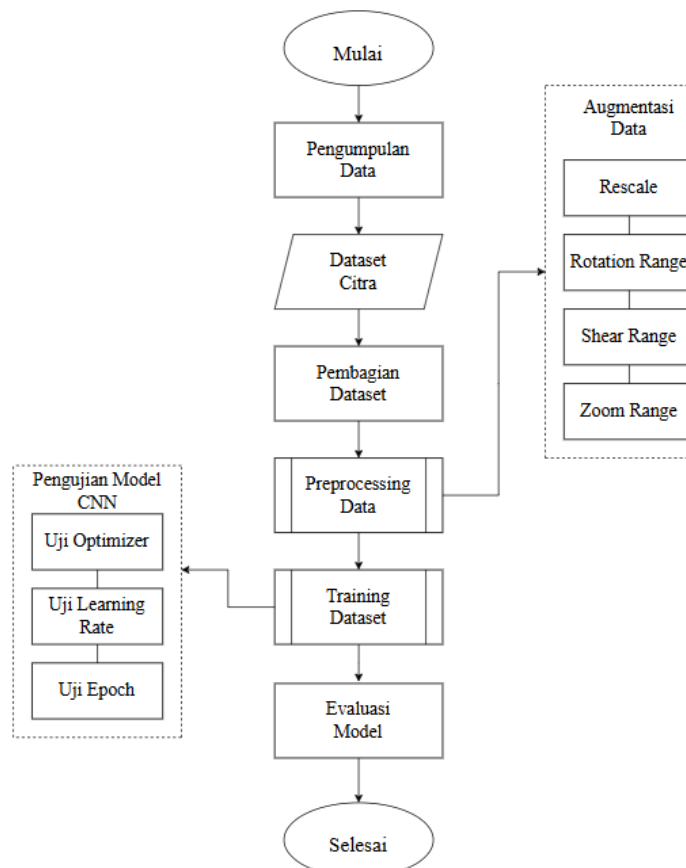
Tanaman tomat merupakan salah satu komoditas pertanian bernilai ekonomi tinggi di Indonesia. Namun, produktivitasnya sering menurun akibat serangan penyakit daun seperti *bacterial spot*, *late blight*, dan *yellow leaf curl virus* [1]. Deteksi dini terhadap penyakit ini menjadi langkah penting untuk menjaga kualitas dan hasil panen.

Perkembangan teknologi kecerdasan buatan, khususnya *Convolutional Neural Network (CNN)* [2], telah banyak dimanfaatkan dalam klasifikasi citra penyakit tanaman. Berbagai arsitektur CNN seperti *VGG16*, *VGG19*, dan *Xception* menawarkan kinerja yang beragam dalam mendeteksi objek visual secara otomatis [3]. Penelitian sebelumnya menunjukkan bahwa CNN mampu mencapai akurasi tinggi dalam klasifikasi daun tomat, melebihi metode konvensional.

Penelitian ini bertujuan untuk membandingkan performa ketiga arsitektur CNN tersebut dalam klasifikasi penyakit pada daun tomat menggunakan dataset dari *Kaggle* dan data pendukung dari lahan pertanian Universitas Islam Riau. Evaluasi dilakukan berdasarkan metrik akurasi, presisi, recall, dan f1-score untuk mengetahui model yang paling optimal.

2. METODE PENELITIAN

Metode penelitian adalah suatu cara yang digunakan untuk mendapatkan informasi yang dibutuhkan dalam menyusun penelitian. Dalam penyusunan ini, peneliti menggunakan beberapa metode untuk mendapatkan informasi yang dibutuhkan [4]. Pada metode penelitian ini digambarkan kedalam bentuk sebuah kerangka penelitian dimulai dari pengumpulan data sampai pada evaluasi model.



Gambar 2. 1 Tahapan Penelitian

2.1 Pengumpulan Data

Proses pengumpulan data pada penelitian ini dilakukan menggunakan data sekunder yang diperoleh dari *website Kaggle* dengan nama *Tomato Leaf Disease* yang nantinya akan digunakan sebagai *dataset* dalam proses pelatihan model[5]. Dalam proses ini diambil sampel data dan nantinya akan diklasifikasikan menjadi empat kelas.

Data yang digunakan berupa citra daun tomat yang sehat dan yang terinfeksi penyakit dengan total citra sebanyak 4000 dan dibagi menjadi 4(empat) kelas yaitu, 1000 citra daun tomat yang sehat (*Healthy*) , 1000 citra daun tomat busuk daun(*Late Blight*), 1000 citra daun tomat bercak bakteri(*Bacterial Spot*) dan 1000 citra daun tomat yang terkena virus kuning keriting(*Yellow Leaf Curl Virus*). Citra yang diambil dari dataset ini berukuran 512x512 pixel sehingga perlu dilakukan *preprocessing* pada citra untuk mendapatkan hasil yang maksimal.

2.2 Pembagian Dataset

Pada penelitian ini dilakukan pembagian data menjadi data *training*, data *testing*, dan data *validation*. *Dataset* dibagi dengan perbandingan 80% : 10% : 10% [6], dengan distribusi pada setiap kelas sebesar 800 citra untuk data *training*, 100 citra untuk data *testing* dan 100 citra dari setiap kelas yang didistribusikan untuk data *validation*.

2.3 Preprocessing

Setelah dilakukan pembagian data selanjutnya masuk ketahap *preprocessing* dengan cara melakukan augmentasi data, tahapan ini bertujuan untuk mencegah terjadinya *overfitting*[7]. Dalam penelitian ini, proses augmentasi dilakukan menggunakan fungsi “*ImageDataGenerator*” dari *library keras*. Keunggulan menggunakan *ImageDataGenerator* ini yaitu memungkinkan proses augmentasi citra secara *real-time* selama proses *training* sedang berlangsung. Beberapa teknik augmentasi data yang digunakan dalam penelitian ini yaitu, *rescale*, *rotation*, *width shift*, *height shift*, *shear*, *zoom*, *horizontal flip* dan *fill mode*.

2.4 Training Dataset

Proses ini digunakan untuk menemukan pola tertentu yang ada pada data. Tahapan ini dilakukan setelah tahapan *preprocessing* pada citra dan akan dibangun menggunakan beberapa model arsitektur yaitu *VGG16*[8], *VGG19* [9] dan *Xception* [10].

Pada tahap ini juga dilakukan beberapa kombinasi *scenario* yang nantinya digunakan untuk perbandingan model sehingga didapatkan model dengan hasil terbaik . Berikut adalah beberapa *parameter* dan *scenario* yang akan dilakukan pengujian diantaranya , *Epoch*, *Optimizer* , Dan *Learning Rate*[11].

2.5 Evaluasi Model

Setelah dilakukan berbagai skenario dengan menggunakan beberapa *hyperparameter* yang telah disediakan Langkah selanjutnya adalah dengan mengevaluasi model dengan melihat metrik nilai akurasi , presisi dan *recall* sehingga dapat menentukan mana model terbaik dalam hal klasifikasi penyakit pada daun tanaman tomat[12].

3. HASIL DAN PEMBAHASAN

Tahapan terakhir dalam sebuah penelitian yaitu melakukan analisis hasil yang memberikan suatu keluaran penelitian. Berdasarkan metodologi yang telah ditentukan, kegiatan penelitian dianalisis menjadi laporan sederhana yang mudah untuk dimengerti.

3.1 Hasil Pengujian Hyperparameter

Hyperparameter sangat memberikan dampak pada pelatihan serta kinerja model. Oleh karena itu, menetapkan *hyperparameter* terbaik merupakan langkah yang penting dalam membuat model yang akan menghasilkan akurasi dan prediksi yang baik. Dalam penelitian ini kombinasi *hyperparameter* yang digunakan diantaranya jumlah *epoch*, *learning rate* dan *optimizer*.

3.1.1 Epoch

Dalam penelitian ini, dilakukan pengujian terhadap beberapa variasi nilai *epoch* untuk memperoleh

hasil pelatihan yang optimal. Evaluasi dilakukan dengan memperhatikan nilai *training loss* dan *validation loss* pada setiap *epoch*, yang menjadi indikator penting dalam menentukan titik terbaik untuk menghentikan pelatihan model. Hasil pengujian ini dilakukan dengan menggunakan model *VGG16* dan menggunakan *hyperparameter* yang lain berupa *learning rate* 0.001 dan *optimizer adam*.

Berikut adalah hasil pengujian jumlah epoch :

Tabel 3. 1 Pengujian *Epoch*

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
10	98.22%	98.00%	0.0555	0.0744
15	98.44%	98.00%	0.0477	0.0687
20	98.66%	98.00%	0.0270	0.0686

Dari percobaan pengujian *epoch* berikut rata-rata *Precision*, *Recall* dan *F1-score* dimulai dari 10 *epoch* dengan rata-rata *Precision* 97%, *Recall* 97% dan *F1-score* 97%, 15 *epoch* dengan rata-rata *Precision* 98%, *Recall* 98% dan *F1-score* 98% dan 20 *epoch* dengan rata-rata *Precision* 98%, *Recall* 98% dan *F1-score* 98%.

Dengan hasil pengujian *epoch* ini dapat diambil kesimpulan bahwa jumlah *epoch* dapat mempengaruhi nilai akurasi dan loss dari model yang dilakukan pelatihan dimana nilai akurasi perlahan naik dan nilai loss perlahan-lahan mengalami penurunan. Hal ini sesuai dengan penelitian sebelumnya yang dilakukan oleh Ningsih (2024) dimana dengan penambahan jumlah *epoch* dapat mempengaruhi nilai akurasi dan loss menjadi lebih baik seiring pelatihan model dengan peningkatan jumlah *epoch*.

3.1.2 Learning Rate

Learning rate merupakan salah satu *hyperparameter* penting yang mengatur seberapa cepat model dalam memperbarui bobot selama proses pelatihan. Jika nilai *learning rate* terlalu tinggi, model dapat belajar dengan cepat namun berisiko mengalami ketidakstabilan. Sebaliknya, nilai yang terlalu rendah dapat menyebabkan proses pelatihan menjadi lambat dan konvergensi tidak optimal (Peng, 2024). Dalam tabel dibawah ini, merupakan hasil pengujian dilakukan menggunakan model *VGG16*, dengan kombinasi *hyperparameter* lain berupa jumlah *epoch* 20 dan *optimizer Adam*.

Tabel 3. 2 Pengujian *Learning Rate*

Learning Rate	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
0.1	97.91%	97.75%	2.4882	4.3341
0.01	97.56%	98.00%	0.2611	0.2839
0.001	98.66%	98.00%	0.0270	0.0686

Dari tabel diatas dengan pengujian menggunakan *learning rate* 0.1 mendapatkan nilai *loss* terendah pada saat *training* yaitu 2.4882, *learning rate* 0.01 memberikan hasil 0.2611 dan *learning rate* 0.001 dengan hasil *training loss* terendah 0.0270. Dapat diambil kesimpulan bahwa nilai dari *learning rate* disini mempengaruhi hasil pengujian baik dari segi akurasi maupun *loss*. Namun, dapat dilihat bahwa nilai dari *training loss* dan *validation loss* disini mengalami penurunan yang sangat signifikan sedangkan nilai akurasi naik perlahan menandakan bahwa jumlah *learning rate* sangat mempengaruhi nilai *loss* dari model yang dilakukan pengujian.

3.1.3 Optimizer

Pada penelitian ini, *optimizer* digunakan untuk mempercepat proses pelatihan dengan menyesuaikan bobot jaringan saraf guna meminimalkan *loss function*. Peran *optimizer* sangat penting karena menentukan seberapa cepat dan stabil model mencapai konvergensi. . Pengujian pada tabel dibawah ini dilakukan dengan menggunakan model *VGG16* dengan mengkombinasikan beberapa *hyperparameter* yaitu, *learning rate* 0.001 dan jumlah *epoch* 20.

Tabel 3. 3 Pengujian *Optimizer*

Optimizer	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
-----------	-------------------	---------------------	---------------	-----------------

Ardian Syahputra, Perbandingan Model Arsitektur CNN Dalam Klasifikasi Penyakit Pada Daun Tanaman Tomat

RMSprop	97.87%	98.00%	0.0646	0.0895
Adam	98.66%	98.00%	0.0270	0.0686

Dengan mengkombinasikan beberapa *hyperparameter* tadi dengan *optimizer RMSprop* dan *optimizer Adam* didapati bahwa *Adam* memberikan hasil yang lebih baik dari segi akurasi maupun *loss* dalam pelatihan model. Dapat disimpulkan juga disini bahwa *optimizer* yang digunakan mempengaruhi hasil dari pelatihan model dan dapat memberikan nilai yang lebih baik jika digunakan dengan kombinasi *hyperparameter* yang tepat.

3.2 Hasil Evaluasi Pembuatan Model

Tabel 3. 4 Hasil Evaluasi Pembuatan Model

No.	Model	Accuracy (%)	Precision	Recall	F1-Score
1	VGG16	98.25	0.98	0.98	0.98
2	VGG19	97.50	0.97	0.97	0.97
3	Xception	95.99	0.95	0.95	0.95

Tabel diatas menyajikan perbandingan hasil evaluasi dari tiga model *CNN*, yaitu *VGG16*, *VGG19*, dan *Xception*, berdasarkan empat metrik utama: *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Metrik-metrik ini dihitung berdasarkan hasil pengujian pada data uji, dan masing-masing nilai merupakan rata-rata dari keempat kelas yang digunakan dalam klasifikasi citra daun tanaman tomat.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa *VGG16* merupakan model yang paling optimal dalam melakukan klasifikasi citra daun tanaman tomat pada penelitian ini, baik dari sisi akurasi maupun kualitas prediksi secara keseluruhan.

3.3 Hasil Klasifikasi Citra

Pada tahap ini dilakukan proses pengujian klasifikasi citra melalui aplikasi berbasis web yang telah diintegrasikan dengan model *CNN* yang telah dilatih sebelumnya. Aplikasi web ini dibangun menggunakan *microframework* Python, yaitu *Flask*, dan dijalankan secara lokal. *Output* yang dihasilkan merupakan kelas dengan probabilitas tertinggi dan nilai kepercayaan dari masing-masing model terhadap hasil klasifikasi tersebut.

Selanjutnya dilakukan proses pengujian menggunakan data baru (diluar dari dataset) dengan scenario data sebanyak 52 citra dengan masing-masing kelas berjumlah 13 citra daun tanaman tomat.

3.3.1 VGG16

Berikut adalah hasil pengujian model *VGG16*:

Tabel 3. 5 Testing Model *VGG16* Via Website

No	Class	Total Image	TP	TN	FP	FN
1	bacterial spot	13	12	39	1	0
2	healthy	13	13	39	0	0
3	late blight	13	13	39	0	0
4	yellow leaf curl virus	13	13	39	0	0
	TOTAL	52	51	156	1	0

Dari hasil *confusion matrix* tersebut didapatkan hasil perhitungan akurasi *precision*, *recall* dan *f1-score* setiap kelas sebagai berikut:

Tabel 3. 6 Classification Report *VGG16*

No	Class	Precision	Recall	F1 Score	Support
1	bacterial spot	0.923	1.00	0.96	13

2	healthy	1.00	1.00	1.00	13
3	late blight	1.00	1.00	1.00	13
4	yellow leaf curl virus	1.00	1.00	1.00	13
		0.98	1.00	0.99	52

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \frac{51+156}{51+156+1+0} = \frac{207}{208} = 0.9952$$

Berdasarkan tabel 3.6 didapatkan hasil klasifikasi model *VGG16* yang sangat baik pada saat dilakukan pengujian menggunakan sebanyak 52 data uji. Akurasi yang diperoleh yaitu sebesar 99.52%.

3.3.2 VGG19

Berikut adalah hasil pengujian model *VGG19*:

Tabel 3. 7 Testing Model VGG19 Via Website

No	Class	Total Image	TP	TN	FP	FN
1	bacterial_spot	13	11	39	2	0
2	healthy	13	12	39	1	0
3	late blight	13	12	39	1	0
4	yellow leaf curl virus	13	13	39	0	0
	TOTAL	52	48	156	4	0

Dari hasil *confusion matrix* tersebut didapatkan hasil perhitungan akurasi *precision*, *recall* dan *f1-score* setiap kelas sebagai berikut:

Tabel 3. 8 Classification Report VGG19

No	Class	Precision	Recall	F1 Score	Support
1	bacterial_spot	0.846	1.00	0.916	13
2	healthy	0.923	1.00	0.96	13
3	late blight	0.923	1.00	0.96	13
4	yellow leaf curl virus	1.00	1.00	1.00	13
		0.923	1.00	0.96	52

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \frac{48+156}{48+156+4+0} = \frac{204}{208} = 0.9808$$

Berdasarkan tabel diatas didapatkan hasil klasifikasi model *VGG19* yang sangat baik pada saat dilakukan pengujian menggunakan sebanyak 52 data uji. Akurasi yang diperoleh yaitu sebesar 98.08%.

3.3.3 Xception

Berikut adalah hasil pengujian model *Xception*:

Tabel 3. 9 Testing Model Xception Via Website

No	Class	Total Image	TP	TN	FP	FN
1	bacterial_spot	13	12	39	1	0
2	healthy	13	11	39	2	0
3	late blight	13	10	39	3	0
4	yellow leaf curl virus	13	13	39	0	0
	TOTAL	52	46	156	6	0

Dari hasil *confusion matrix* tersebut didapatkan hasil perhitungan akurasi *precision*, *recall* dan *f1-score* setiap kelas sebagai berikut:

Tabel 3. 10 Classification Report Xception

No	Class	Precision	Recall	F1 Score	Support
1	bacterial_spot	0.923	1.00	0.9600	13
2	healthy	0.846	1.00	0.9167	13

3	late blight	0.769	1.00	0.8696	13
4	yellow leaf curl virus	1.00	1.00	1.00	13
		0.884	1.00	0.921	52

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \frac{46+156}{46+156+6+0} = \frac{202}{208} = 0.9712$$

Berdasarkan tabel diatas didapatkan hasil klasifikasi model *Xception* yang sangat baik pada saat dilakukan pengujian menggunakan sebanyak 52 data uji. Akurasi yang diperoleh yaitu sebesar 97.12%.

4. KESIMPULAN

Berdasarkan rangkaian eksperimen dan analisis yang telah dilakukan, penelitian ini memberikan sejumlah temuan penting terkait penerapan model *Convolutional Neural Network (CNN)* dalam klasifikasi penyakit pada daun tanaman tomat. Hasil pengujian yang dilakukan menghasilkan tingkat akurasi tertinggi pada model *VGG16* sebesar 99,52%. Sementara itu, model *VGG19* memperoleh akurasi 98,05%, dan *Xception* memperoleh 97,12%. Secara keseluruhan, *VGG16* dinilai sebagai model paling efektif dan konsisten dalam mengklasifikasikan penyakit pada daun tanaman tomat.

DAFTAR PUSTAKA

- [1] Bastari, A. J., & Cherid, A. (2023). Klasifikasi Penyakit Tanaman Tomat Menggunakan Convolutional Neural Network Dan Implementasi Model H5 Pada Aplikasi Desktop. *Jurnal Sistem Informasi dan Sistem Komputer*, 8(2), 199-207.
- [2] Ningsih, N. P., Suryadi, E., Bakti, L. D., & Imran, B. (2022). Klasifikasi Penyakit Early Blight Dan Late Blight Pada Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode Cnn Berbasis Website. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 1(3), 27-35.
- [3] L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in *IEEE Access*, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [4] Putra, J. V. P., Ayu, F., & Julianto, B. (2023, January). Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN. In *Seminar Nasional Teknologi & Sains* (Vol. 2, No. 1, pp. 155-162).
- [5] Mazhar, S. A., Anjum, R., Anwar, A. I., & Khan, A. A. (2021). Methods of data collection: A fundamental tool of research. *Journal of Integrated Community Health*, 10(1), 6-10.
- [6] Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, 10, e2245.
- [7] Prasetyo, S. Y. (2024). Overcoming Overfitting in CNN Models for Potato Disease Classification Using Data Augmentation. *Engineering, Mathematics and Computer Science Journal (EMACS)*, 6(3), 179-184.
- [8] Kusumawati, W. I., & Noorizki, A. Z. (2023). Perbandingan Performa Algoritma VGG16 dan VGG19 Melalui Metode CNN untuk Klasifikasi Varietas Beras. *Journal of Computer, Electronic, and Telecommunication*, 4(2).
- [9] Husodo, K., Lubis, C., & Rusdi, Z. (2023). Klasifikasi Tanaman Anggrek Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19. *Simtek: jurnal sistem informasi dan teknik komputer*, 8(2), 253-258.
- [10] Akram, M. N., Yaseen, M. U., Waqar, M., Imran, M., & Hussain, A. (2023). A Double-Branch Xception Architecture for Acute Hemorrhage Detection and Subtype Classification. *Computers, Materials & Continua*, 76(3).
- [11] Sipper, M. (2022). High per parameter: A large-scale study of hyperparameter tuning for machine

- learning Algorithms. *Algorithms*, 15(9), 315.
- [12] Cabot, J. H., & Ross, E. G. (2023). Evaluating prediction model performance. *Surgery*, 174(3), 723-726.
- [13] Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. *African Journal of Biomedical Research*, 4023-4031.

BIOGRAFI PENULIS

First author's

Ardian Syahputra memperoleh gelar sarjana teknik dari Universitas Islam Riau pada tahun 2025
