



Pengembangan Model *Deep Learning* Untuk Pengenalan Wajah pada Sistem Keamanan

Husain Ahmad Faiq*¹, Hary Sabita²

^{1,2} Institut Informatika dan Bisnis Darmajaya; Jl. Z.A. Pagar Alam N0 93 Gedong Meneng, Kec. Rajabasa, Bandar Lampung, Lampung, 35141, (0721) 787214

Jurusan Teknik Informatika, Institut Informatika dan Bisnis Darmajaya, Lampung

*Email Penulis Korespondensi: hahmadfaiq.1911010165@mail.darmajaya.ac.id

Abstrak

Penelitian ini bertujuan untuk mengembangkan sebuah model Deep Learning menggunakan Convolutional Neural Network (CNN) guna meningkatkan sistem keamanan berbasis sistem pengenalan wajah pada Program Studi Teknik Informatika di Institut Informatika dan Bisnis Darmajaya. Keamanan ruangan atau akses terhadap suatu area tertentu menjadi isu yang sangat penting seperti di gedung perkantoran, pusat data, atau laboratorium riset untuk mencegah terjadinya kebocoran data maupun akses yang tidak sah yang dilakukan oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan teknologi yang canggih dan akurat seperti sistem pengenalan wajah untuk memastikan akses yang aman ke dalam ruangan tersebut. Metode Convolutional Neural Network (CNN) telah terbukti efektif dalam tugas pengenalan pola kompleks seperti wajah manusia. Dengan mengimplementasikan teknik-teknik canggih dalam arsitektur Convolutional Neural Network (CNN), penelitian ini menghasilkan sebuah model yang mampu mengenali wajah dengan akurasi tinggi dan respons yang cepat. Hasil akhir dari penelitian ini berupa model Deep Learning dengan akurasi tinggi yang diintegrasikan dengan Graphical User Interface (GUI) yang nantinya dapat digunakan pada perangkat seluler atau edge devices seperti devices IoT (Internet of Things), kamera pengawas, dan perangkat lainnya sesuai dengan kemampuan komputasi yang dibutuhkan.

Kata kunci— *Pengenalan wajah, deep learning, Convolutional Neural Network, sistem keamanan, akurasi.*

Abstract

This research aims to develop a Deep Learning model using Convolutional Neural Network (CNN) to enhance the security system based on face recognition in the Computer Engineering Study Program at Darmajaya Institute of Informatics and Business. Security of rooms or access to specific areas is a crucial issue in office buildings, data centers, or research laboratories to prevent data leaks or unauthorized access by unauthorized individuals. Therefore, advanced and accurate technology such as face recognition systems is needed to ensure secure access to these rooms. The Convolutional Neural Network (CNN) method has proven effective in complex pattern recognition tasks such as human faces. By implementing advanced techniques in Convolutional Neural Network (CNN) architecture, this research produces a model capable of recognizing faces with high accuracy and fast response. The final result of this research is a Deep

Learning model with high accuracy integrated with a Graphical User Interface (GUI) that can be used on mobile devices or edge devices such as IoT (Internet of Things) devices, surveillance cameras, and other devices according to the required computational capabilities.

Keywords— *Face recognition, deep learning, Convolutional Neural Network, security systems, accuracy.*

1. PENDAHULUAN

Keamanan ruangan atau akses terhadap suatu area tertentu menjadi isu yang sangat penting dalam lingkungan yang memerlukan pengawasan ketat, seperti di gedung perkantoran, lembaga pemerintahan, pusat data, atau laboratorium riset. Sistem keamanan tradisional yang mengandalkan kartu akses atau kata sandi mungkin kurang efektif karena dapat disalahgunakan atau dicuri oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan teknologi yang lebih canggih dan akurat, seperti pengenalan wajah untuk memastikan akses yang aman ke dalam ruangan tersebut.

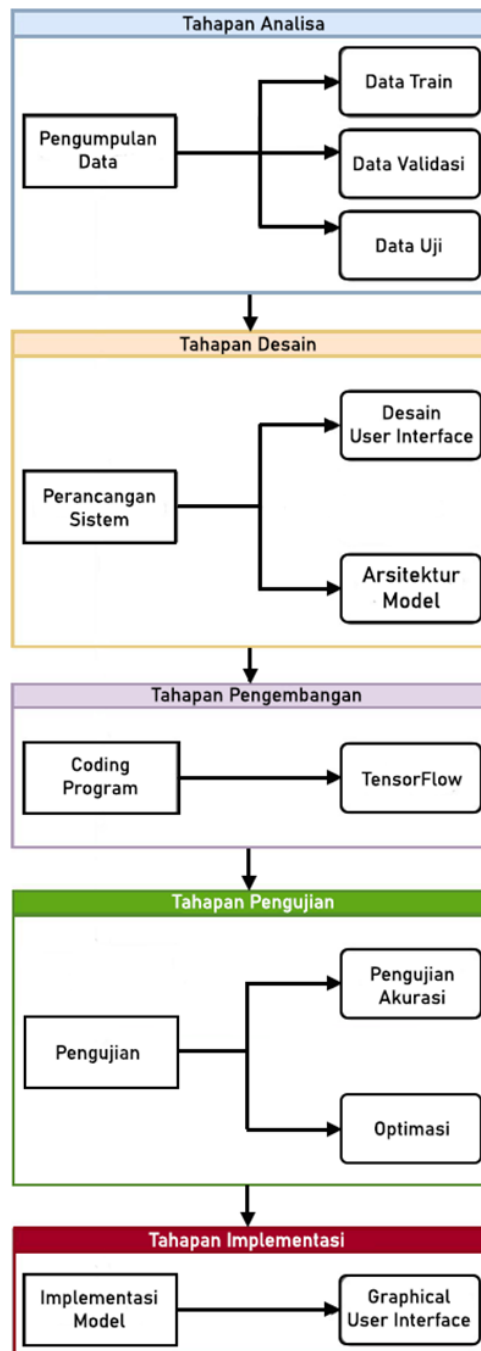
Sistem biometrik semakin mendominasi berbagai aspek kehidupan manusia berkat kemajuan komputasi yang semakin tinggi. Konsep sistem ini merujuk pada identifikasi individu melalui ciri-ciri fisik atau anggota tubuh unik, seperti sidik jari, retina mata, suara, rantai DNA, dan struktur wajah. Uniknya, ciri-ciri fisik ini tidak dapat dijumpai dalam bentuk yang identik antar-individu. Kelebihan teknologi biometrik ini sangatlah signifikan. Tidak hanya cenderung sulit hilang atau terlupakan, namun juga sulit untuk dipalsukan. Pembedaannya yang jelas antara satu individu dengan yang lain menjadi salah satu poin utama. Salah satu metode yang diterapkan dalam sistem biometrik adalah pengenalan wajah, yang sering disebut sebagai *face recognition* [1].

Model *deep learning* adalah salah satu jenis model prediktif yang digunakan dalam pembelajaran mesin, yang menggunakan arsitektur jaringan saraf tiruan dengan banyak lapisan. Dalam konteks pengenalan wajah, model *deep learning* dapat belajar secara mandiri untuk mengidentifikasi ciri-ciri unik dari wajah manusia, seperti bentuk wajah, struktur mata, hidung, mulut, dan sebagainya. [2]

Namun, sistem pengenalan wajah juga masih menghadapi beberapa tantangan. Misalnya, variasi dalam ekspresi wajah, perubahan pencahayaan, maupun variasi suhu lingkungan yang dapat mempengaruhi akurasi pengenalan wajah. Oleh karena itu, perlu dilakukan penelitian dan pengembangan lebih lanjut dalam mengatasi tantangan-tantangan tersebut untuk meningkatkan akurasi model sehingga dapat digunakan secara optimal pada sistem keamanan ruangan.

2. METODE PENELITIAN

Untuk menerapkan Metode *Convolutional Neural Network (CNN)*, terdapat beberapa tahapan yang dilakukan. Tahapan tersebut yaitu [3]: (1) Tahapan Analisa. Pada tahap ini, pengumpulan dataset dilakukan dengan mempersiapkan data gambar yang akan digunakan untuk pelatihan model dan membagi dataset menjadi 3 bagian, data train, data validasi, dan data uji; (2) Tahapan Desain. Setelah persiapan data, langkah berikutnya adalah perancangan desain *User Interface (UI)* yang akan digunakan pada *Graphical User Interface (GUI)* dan dengan merancang arsitektur model; (3) Tahapan Pengembangan. Pada tahap ini pengembangan model dilakukan menggunakan *framework TensorFlow* yang digunakan untuk membangun dan melatih model; (4) Tahapan Pengujian. Langkah berikutnya adalah melakukan pengujian pada data yang belum pernah dilihat sebelumnya untuk menguji kemampuan model dalam memprediksi; dan (5) Tahapan Implementasi. Pada langkah terakhir, model yang telah dikembangkan dan dioptimasi diterapkan pada sistem keamanan menggunakan *Graphical User Interface (GUI)*.



Gambar 1 Tahapan penelitian

2.1 Tahapan Analisa

Pada Tahapan Analisa, data yang digunakan berupa dataset *Labeled Faces in the Wild (LFW)* yang dipublikasikan oleh University of Massachusetts Amherst yang merupakan salah satu *research university* terletak di Amherst, Massachusetts, Amerika Serikat. *Labeled Faces in the Wild (LFW)* merupakan database foto wajah yang dirancang untuk mempelajari masalah pengenalan wajah. Kumpulan data ini berisi lebih dari 13.000 gambar wajah yang dikumpulkan dari web. Setiap wajah telah diberi label sesuai dengan nama orang yang digambarkan. Pada dataset ini terdapat 1680 orang dalam foto memiliki dua atau lebih foto yang berbeda dalam dataset. [4]

2.1.1 Pemilihan Sampel Gambar dari Dataset

Dari dataset *Labeled Face in the Wild (LFW)* pada tahapan sebelumnya, dilakukan pengambilan sampel gambar yang akan digunakan untuk pelatihan dan pengujian model. Pada penelitian ini, saya juga menambahkan beberapa label gambar baru menggunakan teknik pengambilan gambar pada OpenCV. Untuk teknik pemilihan sampel gambar pada dataset, label yang dipilih berdasarkan banyaknya jumlah gambar pada setiap label. Label yang diambil adalah label yang memiliki lebih dari 70 gambar. Proses pemilihan ini menghasilkan 11 label terpilih seperti yang ditampilkan pada Gambar 2, dengan total 1726 sampel gambar yang kemudian diubah menjadi *grayscale* dan dengan ukuran 50x50 piksel.

```

##### (77 samples)      label : Ariel_Sharon
##### (236 samples)     label : Colin_Powell
##### (121 samples)     label : Donald_Rumsfeld
##### (530 samples)     label : George_W_Bush
##### (109 samples)     label : Gerhard_Schroeder
##### (71 samples)      label : Hugo_Chavez
##### (197 samples)     label : Husain_Ahmad_Faiq
##### (80 samples)      label : Muhammad_Saifuddin_Mahfudz
##### (150 samples)     label : Najib_Abdullah_Muqsith
##### (97 samples)      label : Pak_Hary_Sabita
##### (144 samples)     label : Tony_Blair

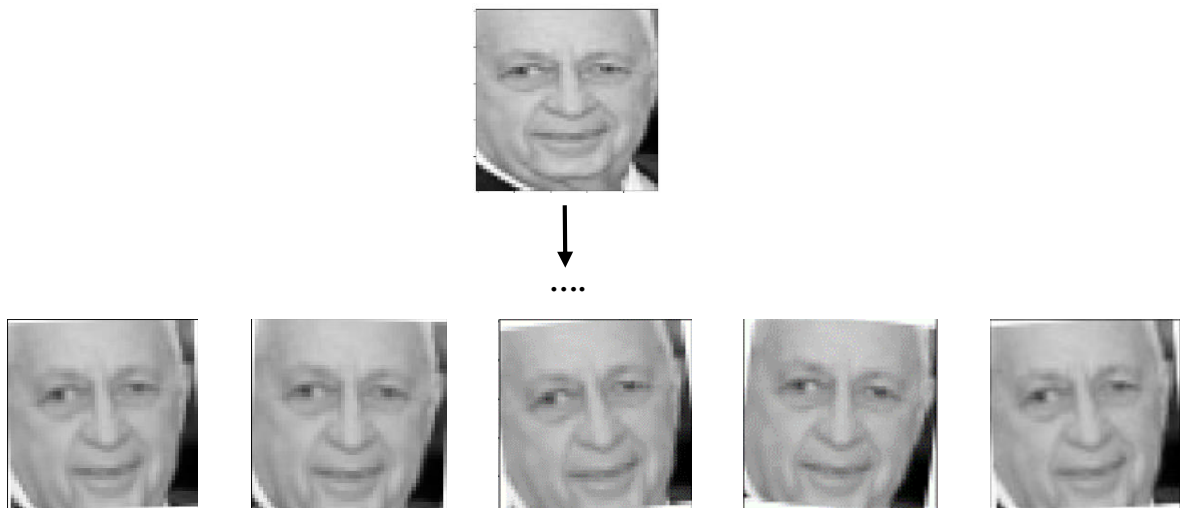
                number of samples : 1726

```

Gambar 2 Sampel yang akan digunakan untuk pelatihan model

2.1.2 Augmentasi dan Balancing Data

Pada proses ini, augmentasi data dilakukan dengan mengambil setiap 1 sampel gambar pada data yang telah dipilih sebelumnya, kemudian diaugmentasikan sehingga setiap 1 gambar pada label menghasilkan 50 sampel gambar baru dengan variasi berbeda. Contoh hasil dari proses augmentasi data ditampilkan pada Gambar 3.



Gambar 3 Beberapa contoh hasil dari proses augmentasi data

Pada proses augmentasi data ini, 1726 sampel gambar menghasilkan 86300 sampel gambar baru, sehingga total keseluruhan gambar yang akan digunakan pada proses pelatihan model sebanyak 88026 sampel gambar.

```

('Ariel_Sharon', 3876)
('Colin_Powell', 11934)
('Donald_Rumsfeld', 6171)
('George_W_Bush', 26928)
('Gerhard_Schroeder', 5508)
('Hugo_Chavez', 3570)
88026 → ('Husain_Ahmad_Faiq', 7599)
('Muhammad_Saifuddin_Mahfudz', 4080)
('Najib_Abdullah_Muqsith', 6069)
('Pak_Hary_Sabita', 4947)
('Tony_Blair', 7344)

```

Gambar 4 Persentase pembagian dan jumlah sampel gambar pada setiap label

Setelah augmentasi data dilakukan, proses selanjutnya yaitu *balancing data* dengan menyamaratakan jumlah sampel gambar pada tiap label. Tujuannya untuk menghindari terjadinya *overfitting* yang menyebabkan masalah pada generalisasi saat mendeteksi wajah [5]. Proses *balancing data* ini dilakukan dengan mengambil sebanyak 3500 sampel gambar pada tiap label, hingga diperoleh total keseluruhan sampel gambar dari 11 label yaitu sebanyak 38500 sampel gambar.

2.1.3 Label Encoding

Encoding adalah salah satu tahap *preprocessing data* dengan merubah tipe data kategori ke numerik sebelum diproses dengan algoritma *machine learning*. Algoritma *machine learning* klasifikasi tidak dapat memproses data bertipe kategori sehingga data harus diubah menjadi berbentuk bilangan. Proses ini disebut dengan encoding [6]. Pada penelitian ini, *label encoding* menggunakan teknik *One-Hot Encoding*. *One-Hot Encoding* menggunakan vektor renggang di mana satu elemen disetel ke 1 dan semua elemen lainnya disetel ke 0 dan merupakan teknik yang umum digunakan untuk merepresentasikan *string* yang memiliki sekumpulan nilai terbatas. Dengan menggunakan *One-Hot Encoding*, kardinalitas tinggi akan menghasilkan vektor fitur berdimensi tinggi [7]. Ada 11 label yang akan digunakan pada proses ini.

```

number of class : 11
['Ariel_Sharon' 'Colin_Powell' 'Donald_Rumsfeld' 'George_W_Bush'
 'Gerhard_Schroeder' 'Hugo_Chavez' 'Husain_Ahmad_Faiq'
 'Muhammad_Saifuddin_Mahfudz' 'Najib_Abdullah_Muqsith' 'Pak_Hary_Sabita'
 'Tony_Blair']

```

Kemudian, label tersebut diubah menjadi *array*.

```
[ 0 0 0 ... 10 10 10]
```

Setelah itu, label diubah menjadi *one-hot encoding* seperti Gambar 5.

```

[[1. 0. 0. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 0. 0. 1.]]

```

Gambar 5 *One-Hot Encoding*

2.1.4 Pembagian Dataset

Proses berikutnya adalah pemisahan data menjadi 3 bagian, yaitu: set pelatihan (*data train*), set validasi (*data validation*), dan set pengujian (*data test*)[8]. Pada penelitian ini, pembagian dataset dari 38500 sampel gambar yang telah diolah dilakukan dengan persentase data

pelatihan dan data validasi 90% (34650), dan data pengujian: 10% (3850 gambar) dari data validasi.

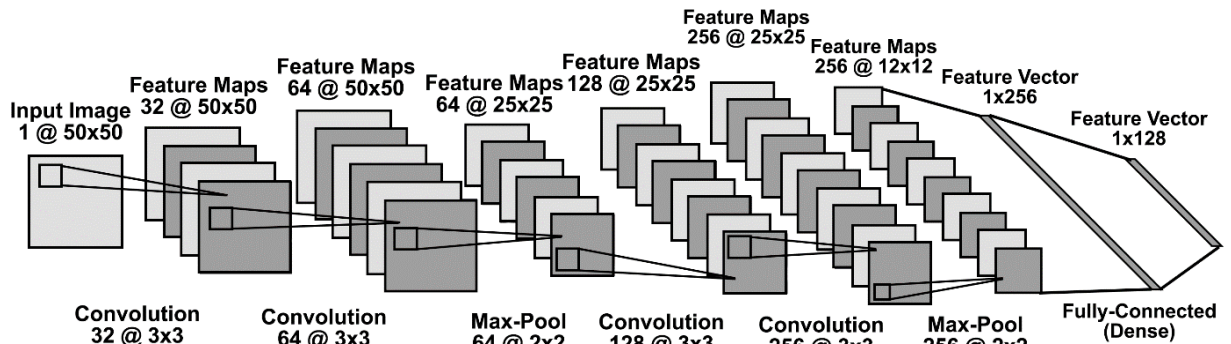
2.2 Tahapan Desain

Tahapan Desain ini dilakukan untuk merancang sistem yang akan digunakan untuk membangun model. Tahap ini terdiri dari perancangan desain *User Interface (UI)* dan arsitektur model yang akan digunakan.

2.2.1 Desain User Interface

Pada *Graphical User Interface (GUI)* akan menggunakan 3 halaman, yaitu: Halaman Utama yang menjadi halaman awal *Graphical User Interface (GUI)*, Halaman Pengenalan Wajah yang digunakan untuk mendeteksi wajah, dan Halaman Tentang yang berisi tentang proyek yang telah dikembangkan.

2.2.2 Arsitektur Model



Gambar 6 Arsitektur Model

Seperti yang ditampilkan pada Gambar 6, model *Convolutional Neural Network (CNN)* ini, menggunakan beberapa lapisan (*layer*), yaitu [9]:

- Convolution Layer 1*. Rumus untuk menghitung jumlah parameter dalam lapisan konvolusi yaitu: $((m * n * d) + 1) * k$,
m = lebar filter;
n = tinggi filter;
d = jumlah filter di lapisan sebelumnya;
k = jumlah filter di lapisan saat ini.
 Jadi, jumlah parameter pada lapisan ini adalah $((3 * 3 * 1) + 1) * 32 = 320$.
- Convolution Layer 2*. Parameter pada lapisan ini adalah $((3 * 3 * 32) + 1) * 64 = 18496$.
- Max- Pooling 1*. Lapisan ini tidak memiliki parameter yang dapat dipelajari. Jadi jumlah parameter pada lapisan ini adalah 0.
- Convolution Layer 3*. Parameter pada lapisan ini adalah $((3 * 3 * 64) + 1) * 128 = 73856$.
- Convolution Layer 4*. Parameter pada lapisan ini adalah $((3 * 3 * 128) + 1) * 256 = 295168$.
- Max- Pooling 2*. Parameter pada lapisan ini adalah 0.
- Flatten*. Fungsi *flatten* digunakan untuk mengubah output dari lapisan konvolusi sebelumnya menjadi satu vektor (*array 1D*) yang akan menjadi input untuk lapisan-lapisan *Fully-Connected (Dense) Layer*. Untuk menghitung jumlah input yang akan digunakan pada *dense layer* yaitu:
 $m * n * k$,
m = lebar filter lapisan sebelumnya;
n = tinggi filter lapisan sebelumnya;

k = jumlah filter lapisan sebelumnya.

Jadi, jumlah input yang akan digunakan pada *dense layer* adalah $12 \times 12 \times 256 = 36864$.

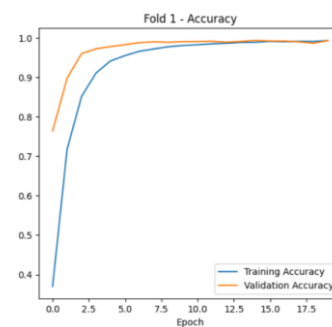
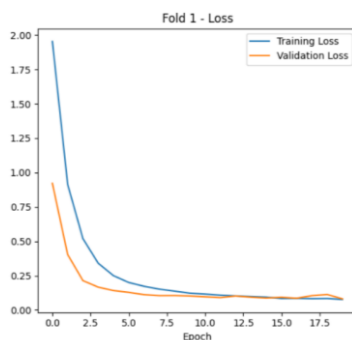
- h. *Fully-Connected (Dense) Layer 1*. Rumus untuk menghitung jumlah parameter dalam lapisan *dense* yaitu: $((c * p) + (1 * c))$,
 c = jumlah filter pada lapisan saat ini;
 p = jumlah neuron di lapisan sebelumnya.
 Jadi jumlah parameter pada lapisan ini adalah $((256 \times 36864) + (1 \times 256)) = 9437440$.
- i. *Fully-Connected (Dense) Layer 2*. Jumlah parameter pada lapisan ini adalah $((32 \times 256) + (1 \times 32)) = 8224$.
- j. *Fully-Connected (Dense) Layer 3*. Jumlah parameter pada lapisan ini adalah $((11 \times 32) + (1 \times 11)) = 363$.

Total parameter dari keseluruhan lapisan adalah $320 + 18496 + 73856 + 295168 + 9437440 + 8224 + 363 = 9833867$ parameter.

2.3 Tahapan Pengembangan

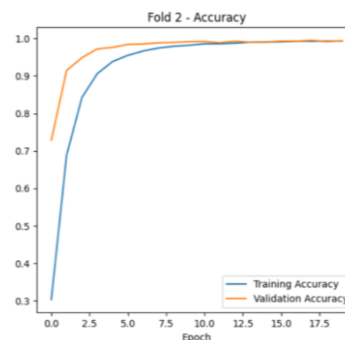
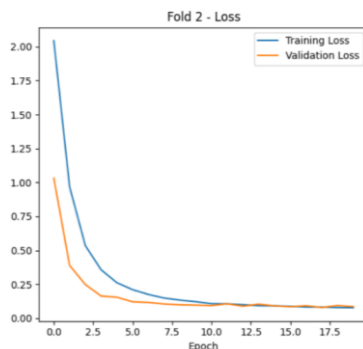
Setelah melalui tahap persiapan data, *pre-processing data*, pembagian dataset, dan perancangan arsitektur, langkah selanjutnya yaitu Tahapan Pengembangan model dengan melakukan proses pelatihan pada model. Pada model ini, dilakukan *L2 Regularization* [10] dengan tingkat regularisasi sebesar 0,01 pada setiap lapisan, menambahkan lapisan *Dropout* [11] pada lapisan *Fully-Connected (dense) layer* dengan probabilitas 0.1 (10%), dan menambahkan *learning rate* [12] pada *optimizer* sebanyak 0.0001. Pada penelitian ini, *Cross-Validation* dilakukan menggunakan 5-Fold dengan masing-masing data validasi yang digunakan diambil dari 20% data training. Berikut adalah hasil 5 model dari 5-Fold yang dihasilkan dengan masing-masing menggunakan 20 epoch.

Fold 1

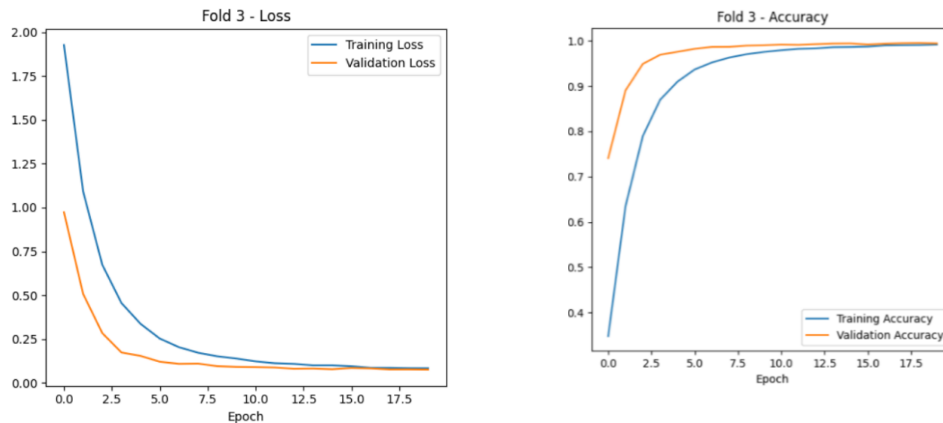


Fold 1 - Loss: 0.0951789990067482, Accuracy: 0.9927272796630859

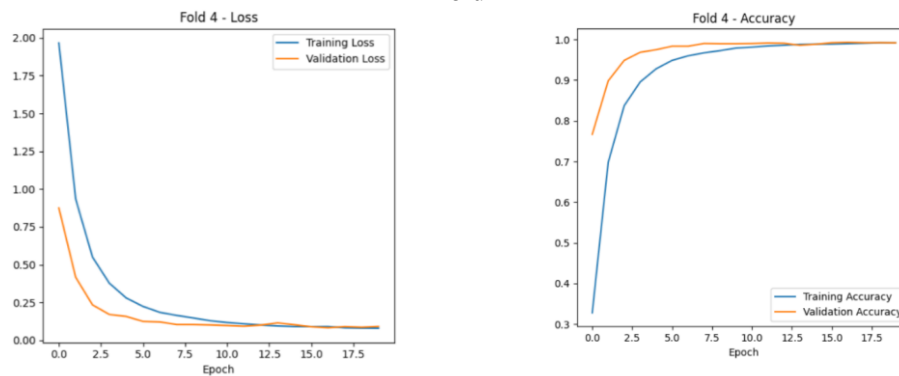
Fold 2



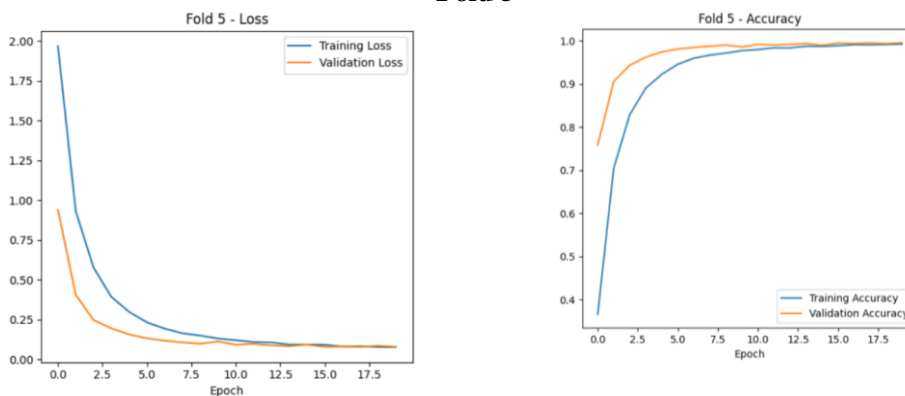
Fold 2 - Loss: 0.09402814507484436, Accuracy: 0.9927272796630859

Fold 3

Fold 3 - Loss: 0.0824093222618103, Accuracy: 0.9942857027053833

Fold 4

Fold 4 - Loss: 0.09219150245189667, Accuracy: 0.9893506765365601

Fold 5

Fold 5 - Loss: 0.09251967072486877, Accuracy: 0.991428554058075

Gambar 7 Hasil 5-Fold dari *Cross-Validation*

Teknik *Cross-Validation* melibatkan pembagian data menjadi beberapa subset atau lipatan (folds), di mana setiap lipatan digunakan sebagai data pelatihan dan data validasi bergantian. Untuk pengambilan model terbaik dari *Cross-Validation* yang dilakukan, yaitu dengan mengambil nilai tertinggi dari nilai akurasi kelima model yang dihasilkan [13], sehingga terpilih Fold 3 sebagai model terbaik dengan nilai akurasi yang ditampilkan pada Gambar 8.

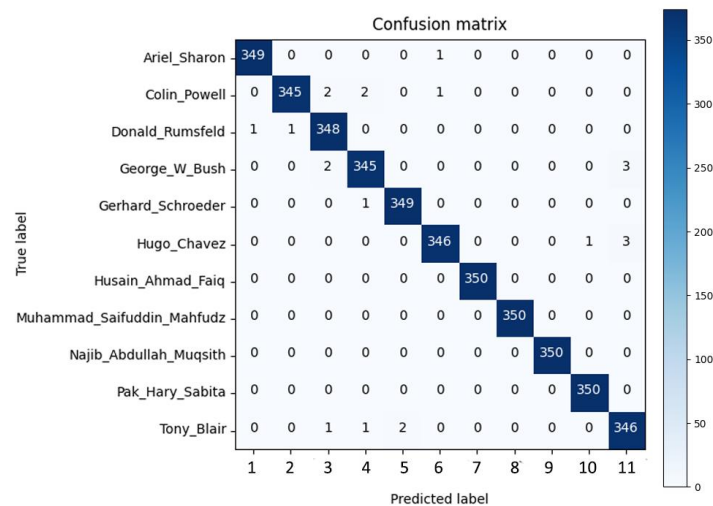
Fold 3 - Loss: 0.0824093222618103, Accuracy: 0.9942857027053833

Gambar 8 Hasil model terbaik dari metode *Cross-Validation*

2.3.1 Perhitungan Akurasi Menggunakan Confusion Matrix

Confusion matrix adalah suatu metode evaluasi performa model klasifikasi yang menyajikan hasil prediksi model dalam bentuk tabel, membandingkan prediksi dengan nilai sebenarnya. [14]

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah keseluruhan data}} \quad (1)$$



Gambar 9 Confusion Matrix

Keterangan:

- | | | |
|---------------------|-------------------------|------------------------|
| 1 = Ariel Sharon | 5 = Gerhard Schroeder | 9 = N Abdullah Muqsith |
| 2 = Colin Powell | 6 = Hugo Chavez | 10 = Pak Hary Sabita |
| 3 = Donald Rumsfeld | 7 = Husain Ahmad Faiq | 11 = Tony Blair |
| 4 = George W Bush | 8 = M Saifuddin Mahfudz | |

Perhitungan nilai akurasi yang dihasilkan:

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah keseluruhan data}}$$

$$\text{Akurasi} = \frac{3828}{3850}$$

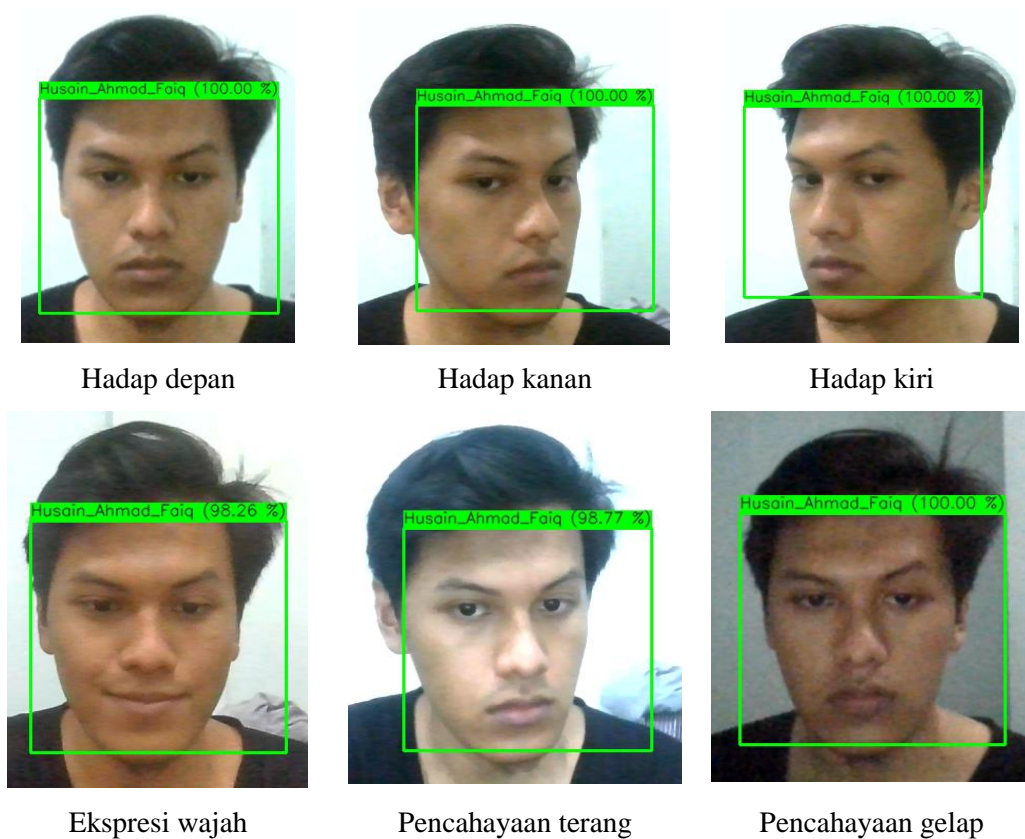
$$\text{Akurasi} = 0.9942857$$

$$\text{Akurasi} = 99\%$$

3. HASIL DAN PEMBAHASAN

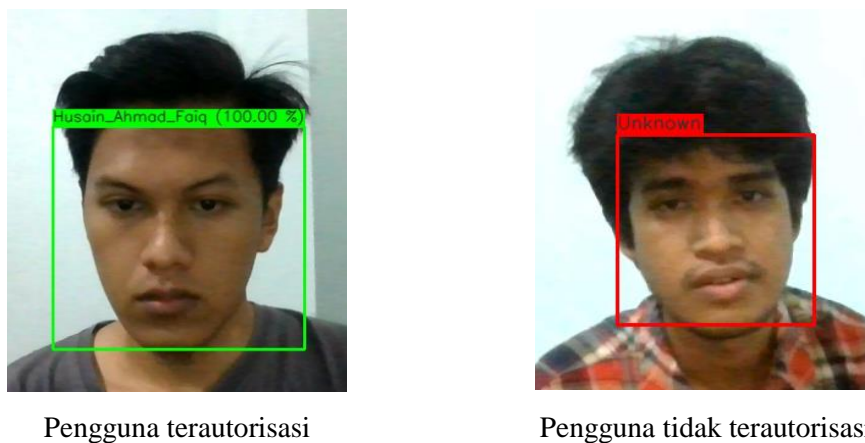
3.1 Tahapan pengujian

Tahap akhir dalam pengembangan model *deep learning* yaitu tahap pengujian, dimana dalam tahap ini, model diuji untuk melakukan prediksi pada data yang belum pernah dilihat sebelumnya. Model diuji menggunakan *video frame* untuk menguji akurasi secara *real-time*. Hasil pengujian ditampilkan pada Gambar 10.



Gambar 10 Pengujian variasi pencahayaan dan pose wajah pada *video frame*

Setelah itu, model juga diuji dengan wajah pengguna yang tidak terdapat pada dataset seperti pada Gambar 10.



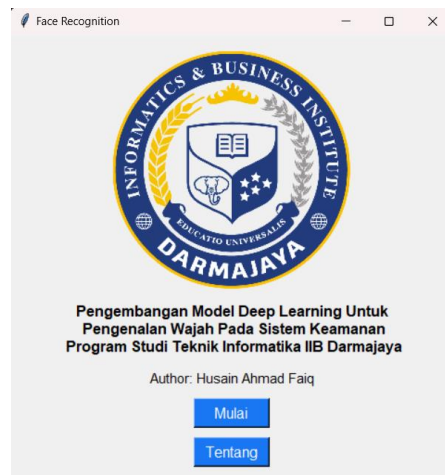
Pengguna terotorisasi

Pengguna tidak terotorisasi

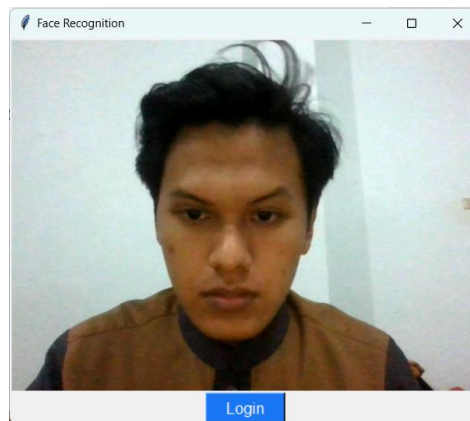
Gambar 11 Deteksi otorisasi menggunakan *video frame*

3.2 Tahapan Implementasi

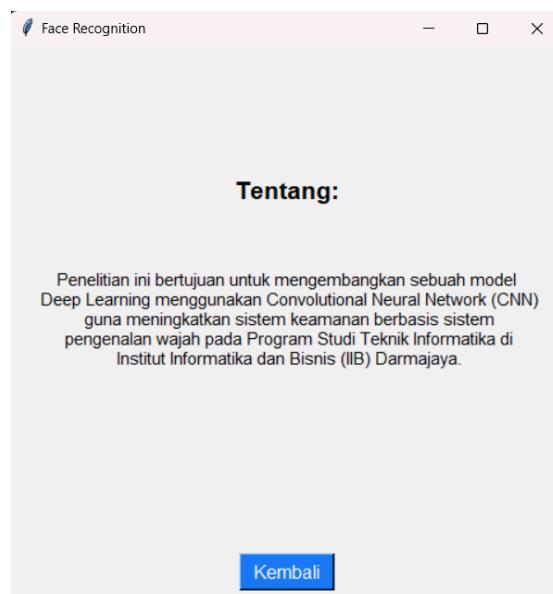
Pada tahapan ini, model yang telah dikembangkan dan dioptimalkan diterapkan pada *Graphical User Interface (GUI)* yang dikembangkan menggunakan *Python*. Adapun pada penelitian ini, *Graphical User Interface (GUI)* yang dikembangkan terdiri dari 3 halaman, yaitu: Halaman Utama, Halaman Pengidentifikasi Wajah, dan Halaman Tentang.



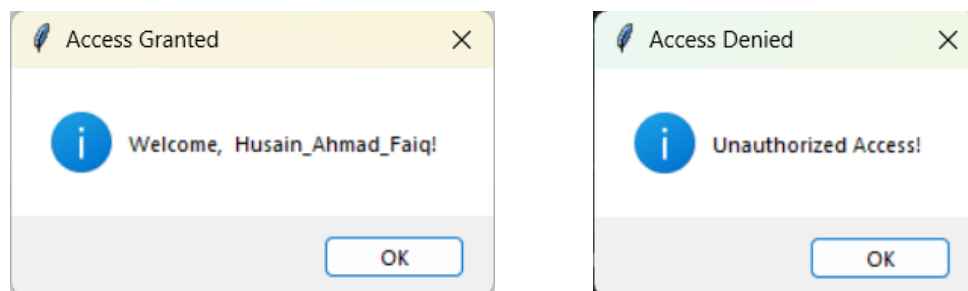
Gambar 12 Halaman Utama



Gambar 13 Halaman Pengidentifikasi Wajah



Gambar 14 Halaman Tentang



Gambar 15 Dialog yang dimunculkan pada *Graphical User Interface (GUI)*

4. KESIMPULAN

Dilihat dari hasil akurasi yang ditampilkan pada Gambar 8, dengan nilai akurasi 99% model ini dapat membantu meningkatkan sistem keamanan pada Program Studi Teknik Informatika dengan menggunakan sistem pengenalan wajah yang dapat mengidentifikasi wajah dengan baik untuk mengatasi akses yang tidak sah pada ruangan. Hasil penelitian ini berbentuk file dengan format *Hierarchical Data Format 5 (HDF5 atau H5)* [15] yang dapat digunakan pada perangkat seluler atau *edge devices* seperti *devices IoT (Internet of Things)*, kamera pengawas, dan perangkat lainnya sesuai dengan kemampuan komputasi yang dibutuhkan.

5. SARAN

Penelitian ini merupakan penelitian awal dan masih diperlukan penelitian lebih lanjut sehingga model yang dikembangkan sesuai dengan kebutuhan Program Studi Teknik Informatika. Untuk itu, berikut ini adalah saran-saran yang dapat diterapkan untuk penelitian berikutnya.

1. Model ini hanya dapat mengidentifikasi identitas wajah, sehingga perlu ditambahkan model *liveness detection* untuk membantu mencegah serangan palsu dengan memastikan bahwa yang diidentifikasi adalah wajah asli dan bukan hanya foto atau rekaman video.
2. Untuk mempermudah pengembangan model, gunakan metode *transfer learning* dalam penambahan dataset ataupun tugas baru. Dengan menggunakan metode ini, model dapat lebih mudah mengenali data baru tanpa harus dilakukan pelatihan model dari awal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknika Politeknik Negeri Sriwijaya yang telah memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

DAFTAR PUSTAKA

- [1] Arkiansyah, Y., & Setiawan, D. Y. 2015. Realisasi CCTV Cerdas Berbasis Mikrokontroler dan Real Time 3D Face Recognition. *Jurnal Informatika*.
- [2] Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. MIT Press.
- [3] Arsal M, Agus Wardijono B, Anggraini D. 2020. "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN". *Jurnal Nasional Teknologi dan Sistem Informasi*.
- [4] University of Massachusetts. 2018. LFW Face Database. <http://vis-www.cs.umass.edu/lfw/>.

-
- [5] Yao, Y., Sullivan IV, T., Yan, F., Gong, J., & Li, L. (2022). Balancing data for generalizable machine learning to predict glass-forming ability of ternary alloys. *Scripta Materialia*, 209, 114366.
- [6] Kristiawan, K., & Widjaja, A. (2021). Perbandingan Algoritma Machine Learning dalam Menilai Sebuah Lokasi Toko Ritel. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(1).
- [7] Bagui, S., Nandi, D., Bagui, S., & White, R. J. (2021). Machine learning and deep learning for phishing email classification using one-hot encoding. *Journal of Computer Science*, 17, 610-623.
- [8] Kahloot, K. M., & Ekler, P. (2021). Algorithmic splitting: A method for dataset preparation. *IEEE Access*, 9, 125229-125237.
- [9] Vasudev, R. 2019. Understanding and Calculating the Number of Parameters in Convolution Neural Networks (CNNs). *Towards Data Science*.
<https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d>.
- [10] Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). L2 regularization for learning kernels. arXiv preprint arXiv:1205.2653.
- [11] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
- [12] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265.
- [13] Chollet, F. 2021. *Deep Learning With Python, Second Edition*. Minning.
- [14] Karimi, Z. 2021. *Confusion Matrix*. ResearchGate.
- [15] Koranne, S., & Koranne, S. (2011). Hierarchical data format 5: HDF5. *Handbook of open source tools*, 191-200.