

PENERAPAN BIOMETRIC FACE RECOGNITION MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK PADA APLIKASI BERBASIS ANDROID

Fahmi Fauzi Abdullah¹⁾, Soffiana Agustin²⁾

^{1,2)} Teknik Informatika, Fakultas Teknik Universitas Muhammadiyah Gresik

Jl. Sumatra 101 Gresik Kota Baru (GKB), Randuagung, 661121

e-mail: fahmifauziabdullah26@gmail.com¹⁾, soffiana@umg.ac.id²⁾

(Naskah masuk : 20 Desember 2022 Diterima untuk diterbitkan : 16 April 2024)

ABSTRAK

Sistem keamanan dengan tingkat akurasi tinggi sangat diperlukan untuk mencegah kebocoran data, salah satu cara mencegah kebocoran data adalah dengan menerapkan sistem keamanan biometrik face recognition pada sebuah sistem login. Pada penelitian ini, dilakukan penerapan face recognition pada sistem login aplikasi bengkel online (Mecha). Penerapan dilakukan dengan menggunakan metode Convolutional Neural Network yang diterapkan pada aplikasi berbasis android dengan menggunakan tensorflow lite melalui penambahan assets berekstensi .tflite pada Integrated Development environment (IDE) Android Studio serta library google ML kit yang digunakan sebagai proses pre- processing. Hasil penelitian menunjukkan bahwa sistem keamanan biometrik face recognition dapat diterapkan pada aplikasi berbasis android dan metode Convolutional Neural Network dapat digunakan untuk mengenali wajah meskipun dalam kondisi gelap pada aplikasi bengkel online (Mecha), penerapan metode Convolutional Neural Network pada aplikasi bengkel online (Mecha) memberikan rata – rata akurasi sebesar 88,25%. Dengan pengujian dari pengguna asli sebesar 99,25% dan pengujian dari bukan pengguna asli sebesar 77,25%.

Kata Kunci: face recognition, convolutional neural network, android

ABSTRACT

A security system with a high level of accuracy is needed to prevent data leaks. One way to prevent data leaks is by implementing a biometric facial recognition security system in a login system. This research purpose is to implement facial recognition in an online workshop application login system (Mecha). The implementation was carried out using the Convolutional Neural Network method which is applied to an Android-based application using TensorFlow Lite by adding assets with the .tflite extension to the Android Studio Integrated Development Environment (IDE) as well as the Google ML Kit library which was used as a pre-processing process. The results show that the biometric face recognition security system can be applied to Android-based applications and the Convolutional Neural Network method can be used to recognize faces even in dark conditions in the online workshop application (Mecha) well. By using testing data from real original users is 99.25% and from non-original users is 77.25% accuracy. The average result is 88.25% of accuracy.

Keywords: face recognition, convolutional neural network, android.

I. PENDAHULUAN

Sistem keamanan dengan tingkat akurasi tinggi sangat diperlukan untuk melindungi data pengguna mengingat banyaknya kasus kebocoran data yang terjadi pada masa kini [1]. Contohnya kasus kebocoran data yang dialami oleh Bank Indonesia pada tahun 2022 [2]. Kebocoran data ini dapat dicegah dengan cara menerapkan sebuah sistem keamanan dengan tingkat akurasi tinggi. Salah satu sistem keamanan dengan akurasi tinggi adalah sistem keamanan biometrik. Sistem keamanan biometrik merupakan sebuah sistem keamanan dengan menggunakan data biometrik sebagai cara untuk mengidentifikasi individu. Sistem keamanan biometrik digunakan karena bersifat akurat dan sulit untuk dipalsukan. Adapun data biometrik meliputi struktur wajah, retina, iris mata, sidik jari, telapak tangan, dan DNA. Salah satu teknologi biometrik yang ada yaitu face recognition.

Face recognition merupakan sebuah metode teknologi komputer yang dapat membandingkan

sebuah citra wajah dengan basis data wajah sehingga dapat ditemukan kecocokan antara basis data wajah dengan citra masukan [3]. Face recognition adalah teknik pengenalan wajah yang memiliki konsep sama seperti sidik jari, dimana hasil citra masukan akan dicocokkan dengan citra yang sudah ada pada basis data. Face recognition sering diterapkan pada sistem keamanan sebuah aplikasi khususnya untuk keperluan login.

Pada sebuah aplikasi android, umumnya terdapat sebuah sistem login sederhana yang hanya mengharuskan pengguna memasukkan informasi berupa email dan kata sandi tanpa verifikasi lebih lanjut. Tidak adanya verifikasi lebih lanjut menyebabkan kerentanan keamanan data pengguna pada aplikasi tersebut. Untuk mengatasi masalah ini, salah satu hal yang dapat dilakukan adalah menerapkan face recognition pada sistem login aplikasi android.

Convolutional Neural Network (CNN) merupakan salah satu varian algoritma yang dapat digunakan pada face recognition. CNN memiliki konsep utama dimana suatu citra akan diekstraksi setiap fituranya sehingga terbentuk beberapa pola yang lebih mudah untuk diklasifikasi. Penggunaan teknik ini dapat membuat proses pembelajaran gambar menjadi lebih efisien untuk diterapkan. CNN merupakan sebuah arsitektur jaringan syaraf tiruan yang efektif untuk klasifikasi citra dan digunakan dalam pengenalan dan pemrosesan gambar [4].

Jaringan syaraf tiruan (JST) merupakan sebuah sistem pemrosesan informasi yang mengadaptasi karakteristik jaringan syaraf biologi manusia [5]. Terdapat banyak varian JST seperti CNN [6], Backpropagation [7], Perceptron [8], Learning Vector Quantization [9] dll. Varian JST yang dapat diterapkan pada face recognition antara lain Backpropagation dan CNN, peneliti memilih CNN dikarenakan arsitektur CNN lebih sederhana sehingga prosesnya lebih ringan dan penerapannya pada aplikasi android lebih mudah serta hasil yang lebih akurat dan efisien [10].

Pada penelitian lain yang dilakukan oleh Khatina Sari (2022) yang berjudul Perancangan Sistem Absensi Facial Recognition Menggunakan CNN dan Liveness Detector pada BPR Central Dana Mandiri, hasilnya pengenalan wajah dengan CNN berhasil diterapkan dan dinilai efektif. Pada penelitian lain yang dilakukan oleh Indra Fransiskus Alam (2019) yang berjudul Implementasi deep learning dengan metode Convolutional Neural Network untuk identifikasi objek secara real time berbasis android, hasilnya sistem pengenalan wajah berhasil diterapkan menggunakan metode CNN pada android. Pada penelitian terdahulu yang dilakukan oleh Alfeus Adi Saputra (2019) yang berjudul implementasi Face Recognition pada sistem absensi berbasis android menggunakan metode Convolutional Neural Network (CNN) dan Lock Gps, hasilnya Face Recognition dapat dibuat dengan menerapkan salah satu varian JST yaitu Convolutional Neural Network.

Berdasarkan penelitian yang sudah pernah dilakukan terbukti bahwa CNN dapat diterapkan untuk Face Recognition sehingga pada penelitian ini penulis memilih metode CNN untuk Face Recognition pada sistem login aplikasi berbasis android.

II. METODE PENELITIAN

2.1 Analisis Sistem

Analisis sistem bertujuan untuk mendapatkan pemahaman secara penuh sistem yang akan dikembangkan. Proses login pada aplikasi android umumnya sangat sederhana, pengguna hanya diharuskan untuk memasukkan email dan kata sandi. Proses login semacam ini dapat menyebabkan resiko pembobolan akun, untuk itu diperlukan sebuah sistem yang dapat melakukan verifikasi tambahan guna meningkatkan keamanan proses login guna mencegah pembobolan akun.

2.2 Hasil Analisis

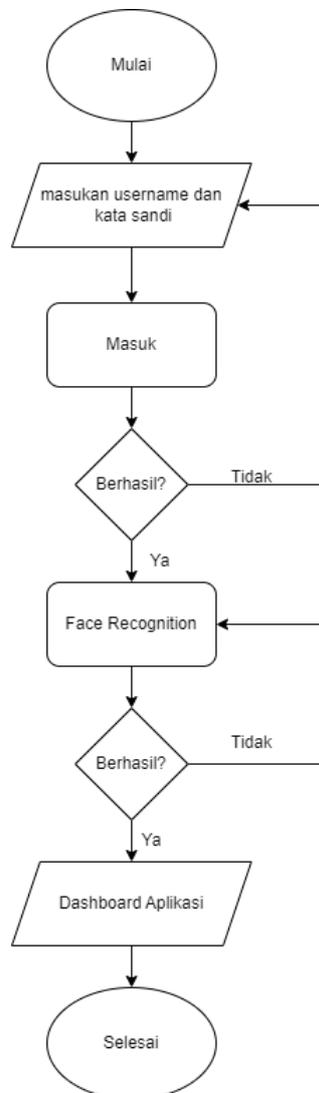
Penelitian ini dibuat untuk aplikasi android yang memberikan pelayanan privasi sehingga user bisa merasakan bahwa data yang diberikan lebih aman. Dalam hal ini akan dibuat suatu sistem verifikasi login dengan menggunakan Face Recognition dan menggunakan metode Convolutional Neural Network.

Adapun penerapan CNN umumnya menggunakan *library tensorflow* namun pada penerapannya di aplikasi android diperlukan *library* khusus yaitu *tensorflow lite* dan *library Google ML Kit* agar *face recognition* dapat diterapkan .

2.3 Perancangan Sistem

Sistem Login

Flowchart dibawah ini menjelaskan proses login pada aplikasi android



Gambar 2.1 Flowchart sistem login

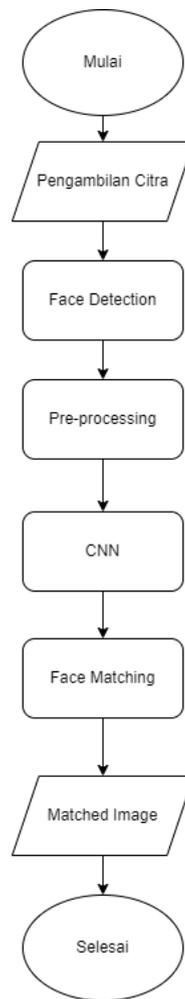
Gambar 2..1 menggambarkan flowchart dari *Face Recognition* pada aplikasi bengkel online (Mecha). Dimulai dari proses login dengan memasukkan username dan kata sandi, setelah proses login berhasil sistem beralih ke proses deteksi wajah.

Setelah wajah terdeteksi dan pengguna mengambil gambar dan menekan tombol kirim, sistem akan mengirim gambar yang sudah diambil ke basis data aplikasi bengkel online (Mecha). Kemudian sistem akan mengambil kembali gambar dan mengirimnya ke aplikasi lalu dilakukan proses *Face Recognition* dan jika berhasil maka pengguna akan masuk ke dashboard aplikasi bengkel online (Mecha).

Proses sistem login yang akan dijelaskan merupakan sistem login aplikasi bengkel online (mecha) yang ditambahkan *Face Recognition*. *Face Recognition* digunakan untuk verifikasi lanjut dari username dan kata sandi yang dimasukkan oleh pengguna. Setelah pengguna berhasil masuk maka sistem akan mengarahkan ke *Face Recognition* dan setelah proses *Face Recognition* berhasil maka pengguna akan dapat masuk ke dalam dashboard aplikasi.

2.4 Face Recognition

Flowchart di bawah ini menjelaskan bagaimana proses *Face Recognition*.



Gambar 2.2 Flowchart Face Recognition

Pada Gambar 2.2 **Error! Reference source not found.** menggambarkan proses *Face Recognition*, dimulai dari pengambilan citra kemudian dilakukan *face detection*, *face alignment* dan *pre-processing*, selanjutnya citra akan melalui proses *feature extraction* untuk selanjutnya dilakukan proses *classification* dan akhirnya dilakukan proses *face matching* sehingga akan menghasilkan luaran sebuah gambar yang sudah dicocokkan atau *matched image*.

Pada *Face Recognition* terdapat beberapa proses yaitu pengambilan citra, *face detection*, *face alignment*, *pre-processing*, *feature extraction*, *classification*, *face matching*. *Face Recognition* diawali dengan pengambilan citra lalu dilakukan proses *face detection*, *face alignment* dan *pre-processing*, selanjutnya citra akan melalui proses *feature extraction* untuk selanjutnya dilakukan proses *classification* dan akhirnya dilakukan proses *face matching* sehingga akan menghasilkan luaran sebuah gambar yang sudah dicocokkan atau *matched image*.

2.5 Pengambilan Citra

Proses pengambilan citra dilakukan menggunakan kamera depan perangkat android.

Face Detection

Pada tahap ini citra yang sudah ditangkap akan dilakukan deteksi wajah oleh *Google ML Kit* dengan mengonfigurasi detektor wajah dan mengaturnya menjadi `LANDMARK_MODE_NONE`,

PERFORMANCE_MODE_FAST, CLASSIFICATION_MODE_NONE. Sintaks di atas berfungsi untuk mengoptimalkan deteksi wajah dengan mematikan landmark dan klasifikasi wajah (tersenyum atau tidak) sehingga performanya lebih optimal.

Pre-processing

Pada proses ini akan dilakukan proses *cropping* pada bagian yang sudah terdeteksi sebagai wajah oleh *Google ML kit*. Ukuran citra diubah menjadi 112 pixel x 112pixel. Terdapat empat bitmap yang akan digunakan pada proses ini yaitu *rgbFrameBitmap*, *croppedBitmap*, *potraitBmp*, dan *faceBmp*.

Feature Extraction

Proses ini menggunakan metode CNN dimana terdapa *Convolutional Layer* dan *Pooling layer*, *output* dari proses ini adalah sebuah *feature map* yang nantinya akan digunakan pada proses *classification*.

Classification

Pada proses ini biasanya menggunakan *fully connected layer* atau *global average pooling layer*, sedangkan pada sistem ini digunakan *Global Depthwise Convolution layer (GDConv)* dimana kernel yang digunakan sesuai dengan ukuran input, *padding* = 0 dan *stride* = 1.

Face matching

Pada tahap ini citra yang sudah diklasifikasi akan dicocokkan dengan citra masukan. CNN akan mengambil masukan berupa wajah dan akan memberikan luaran berupa vektor 128D, vektor ini disebut dengan *embedding*. Jarak *eculidean* antar *embedding* digunakan untuk menghitung *similarity* antara citra yang sudah diklasifikasi dengan citra masukan. Untuk menghitung *similarity* digunakan rumus pada persamaan (1).

$$\begin{aligned} \text{Similarity}(F1, F2) &= ||\text{CNN}(F1) - \text{CNN}(F2)||_2 \\ &= \sqrt{\sum_{i=1}^D (E1_i - E2_i)^2} \end{aligned} \quad (1)$$

Keterangan :

F1 = Citra yang sudah diklasifikasi

F2 = Citra masukan

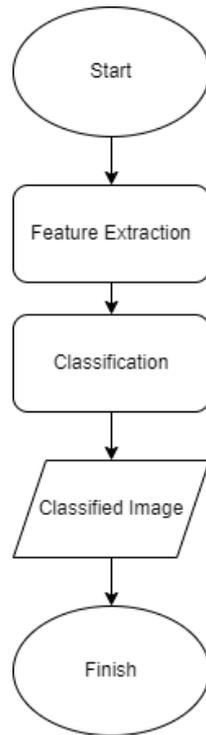
D = Vektor 128D

E = Jarak *Euclidean*

Pada *MobileFaceNet* perhitungan *similarity* berdasarkan pada nilai *confidence field* yaitu 0,01 sampai dengan 1, citra akan dikenali jika nilai *confidence field* dibawah 1, jadi tidak ada batasan persentase khusus untuk *similarity*. Setelah memenuhi threshold tersebut maka sistem akan mengenali citra dan Setelah ditemukan kesamaan citra wajah maka sistem akan beralih ke *dashboard* aplikasi.

2.6 CNN

Flowchart di bawah ini menjelaskan bagaimana metode CNN diterapkan dalam proses *face recognition*, Gambar 2.3 menggambarkan penerapan metode CNN pada *face recognition*. Proses ini diterapkan dengan cara menambahkan *assets mobile_face_net* berekstensi *.tflite*. Metode CNN digunakan pada proses *feature extraction* dan *classification*. Pada proses *feature extraction* terdapat *convolutional layer* dan *pooling layer*, sedangkan pada proses *classification* terdapat *Global Depthwise Convolution*.



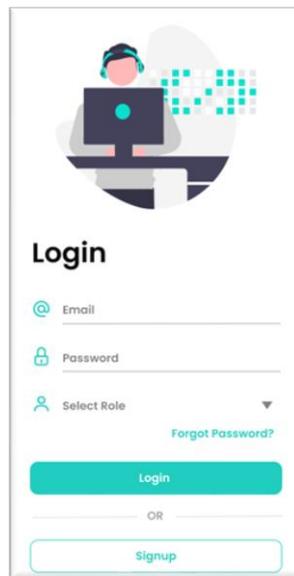
Gambar 2.3 Flowchart Sistem Login

III. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Sistem Login

Sistem login yang dibutuhkan untuk dapat menerapkan *face recognition* adalah sistem login yang memerlukan sebuah verifikasi lanjutan, pada penelitian ini sebagai percontohan digunakan sistem login pada aplikasi bengkel online (Mecha), untuk tampilan sistem login pada aplikasi bengkel online (Mecha) dapat dilihat pada Gambar 3.1.



Gambar 3.1 Antarmuka Halaman Login

Face Recognition

Pengambilan Citra

Pada tahap ini sistem membutuhkan izin untuk mengakses kamera perangkat yang berfungsi untuk mengambil citra. Citra yang akan diambil tidak harus berlatar polos, dapat dilihat pada Gambar bahwa citra tidak berlatar polos. Untuk kode akses kamera serta pengambilan citra dapat dilihat pada Kode Program 1. Akses Kamera, Kode Program 2. Pengambilan Citra.



Gambar 3.2 Antarmuka Pengambilan Citra

Kode Program 1. Akses Kamera

```
1 <uses-permission android:name="android.permission.CAMERA" />
```

Kode Program 2. Pengambilan Citra

```
1 setContentView(R.layout.tfe_od_activity_camera);
2 Toolbar toolbar = findViewById(R.id.toolbar);
3 setSupportActionBar(toolbar);
4 getSupportActionBar().setDisplayShowTitleEnabled(false);
5 if (hasPermission()) {
6     setFragment();
7 } else {
8     requestPermission();
9 }
```

Face Detection

Pada Face Detection digunakan library Google ML Kit untuk mendeteksi wajah pada citra yang akan diambil, untuk menggunakan Google ML Kit dapat dilakukan dengan menambah dependencies. Pada Gambar ditampilkan wajah yang telah terdeteksi dan ditandai dengan munculnya kotak berwarna biru, ukuran kotak ini mengikuti besar kecilnya wajah pengguna.

Kode Program 3. penambahan Google ML Kit pada dependencies.

```
1 implementation 'com.google.mlkit:face-detection:16.0.0'
```



Gambar 3.3 Antarmuka Face Detection*Kode Program 4. Face Detection*

```

1 FaceDetectorOptionsoptions=
2 newFaceDetectorOptions.Builder()
3 .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_FAST).setContourMode(FaceDetectorOptions.LAND
  MARK_MODE_NONE).setClassificationMode(FaceDetectorOptions.CLASSIFICATION_MODE_NONE)
4 .build();
5 FaceDetectordetector=FaceDetection.getClient(options);
6 faceDetector=detector;
7 }

```

Pre-Processing

Pada Proses *pre-processing* dilakukan *cropping* sehingga citra dipotong menjadi ukuran 112 pixel dan berfokus pada wajah yang telah terdeteksi oleh proses *face detection*. Citra yang telah dipotong ini nantinya akan digunakan untuk proses *feature extraction* dan *classification*.

**Gambar 3.4** Antarmuka Pre-processing*Kode Program 5. Pre-processing*

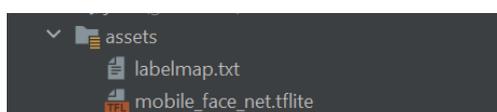
```

1 FaceDetectorOptionsoptions=
2 newFaceDetectorOptions.Builder()
3 .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_FAST).setContourMode(FaceDetectorOptions.LAND
  MARK_MODE_NONE).setClassificationMode(FaceDetectorOptions.CLASSIFICATION_MODE_NONE)
4 .build();
5 FaceDetectordetector=FaceDetection.getClient(options);
6 faceDetector=detector;
7 }

```

CNN

Pada tahap CNN ini terdapat dua proses yaitu *feature extraction* dan *classification*. Untuk menerapkan dua proses ini dilakukan dengan menambahkan *assets* *mobile_face_net* berekstensi *.tflite* pada Android Studio.

**Gambar 3.5** Assets *mobile_face_net*

Kode Program 6. Feature Extraction

```

1  FaceDetectorOptionsoptions=
2  newFaceDetectorOptions.Builder()
3  .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_FAST).setContourMode(FaceDetectorOptions.LAND
   MARK_MODE_NONE).setClassificationMode(FaceDetectorOptions.CLASSIFICATION_MODE_NONE)
4  .build();
5  FaceDetectordetector=FaceDetection.getClient(options);
6  faceDetector=detector;
7  }
    
```

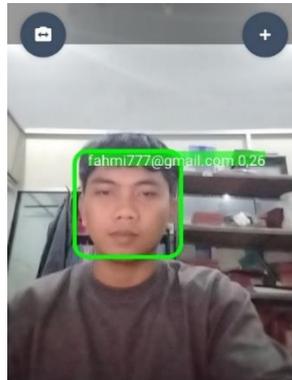
Kode Program 7. Classification

```

1  for i in range(len(issame_list)*2):
2  _bin = bins[i]
3  #img = mx.image.imdecode(_bin).asnumpy()
4  img = cv2.imdecode(np.fromstring(_bin, np.uint8), -1)
5  img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
6  img = mx.image.imdecode(_bin).asnumpy()
7  # img = cv2.imdecode(np.fromstring(_bin, np.uint8), -1)
8  # img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
9  #img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
10 img = img - 127.5
11 img = img * 0.0078125
    
```

Face Matching

Pada proses ini jarak euclidean antara citra data latih dan citra data uji dihitung menggunakan rumus persamaan 3.1. Semakin kecil jarak euclidean maka gambar akan semakin dianggap sama atau dikenali. Rentan jarak yang digunakan yaitu antara 0,01 hingga 1. Jika jarak euclidean melebihi 1 maka gambar tidak dianggap sama atau tidak dikenali. Jika kotak pendeteksi wajah berubah menjadi hijau atau wajah dikenali maka pengguna akan diarahkan masuk ke dashboard aplikasi.



Gambar 3.6 Antarmuka Face Matching

Kode Program 8. Face Matching

```

1  if(registered.size()>0){
2  finalPair<String,Float>nearest=findNearest(embeddings[0]);
3  if(nearest!=null){
4  finalStringname=nearest.first;
5  label=name;
6  distance=nearest.second;
7
8  LOGGER.i("nearest:" +name+"-distance:" +distance);
9  }}
    
```

3.2 Pengujian Sistem

Pengujian face recognition akan digunakan rumus pada persamaan **Error! Reference source not found.** Pada pengujian ini akan menghitung nilai akurasi face matching, face matching merupakan proses terakhir dari face recognition dengan CNN. Dalam penelitian ini digunakan citra sebanyak 122 yang terdiri dari 30 data latih dengan cahaya terang, 30 data latih dengan cahaya gelap, 30 data uji dengan cahaya terang, 30 data uji dengan cahaya gelap, 1 data uji bukan pengguna asli dengan cahaya

terang, 1 data uji bukan pengguna asli dengan cahaya gelap, Citra yang digunakan pada pengujian ini dapat dilihat pada lampiran.

Tabel 3.1 Rekapitulasi Hasil Pengujian Benar

	Data Uji Terang	Data Uji Gelap	Data Uji Bukan Pengguna Asli Terang	Data Uji Bukan Pengguna Asli Gelap
Data Latih Terang	30	30	7	5
Data Latih Gelap	29	30	5	10

Tabel 3.1 menampilkan rekapitulasi hasil pengujian, dimana data latih terang dibandingkan dengan data uji terang mendapatkan hasil 30 benar, data latih terang dibandingkan dengan data uji gelap mendapatkan hasil 30 benar, data latih gelap dibandingkan dengan data uji terang mendapatkan hasil 29 benar, data latih gelap dibandingkan dengan data uji gelap mendapatkan hasil 30 benar, data latih terang dibandingkan dengan data uji bukan pengguna asli terang mendapatkan hasil 7 benar, data latih gelap dibandingkan dengan data uji bukan pengguna asli terang mendapatkan hasil 5 benar, data latih terang dibandingkan dengan data uji bukan pengguna asli gelap mendapatkan hasil 5 benar, data latih gelap dibandingkan dengan data uji bukan pengguna asli gelap mendapatkan hasil 10 benar.

Tabel 3.2 Rekapitulasi Hasil Pengujian Salah

	Data Uji Terang	Data Uji Gelap	Data Uji Bukan Pengguna Asli Terang	Data Uji Bukan Pengguna Asli Gelap
Data Latih Terang	0	0	23	25
Data Latih Gelap	1	0	25	20

Tabel 3.2 menampilkan rekapitulasi hasil pengujian, dimana data latih terang dibandingkan dengan data uji terang mendapatkan hasil 0 salah, data latih terang dibandingkan dengan data uji gelap mendapatkan hasil 0 salah, data latih gelap dibandingkan dengan data uji terang mendapatkan hasil 1 salah, data latih gelap dibandingkan dengan data uji gelap mendapatkan hasil 0 salah, data latih terang dibandingkan dengan data uji bukan pengguna asli terang mendapatkan hasil 23 salah, data latih gelap dibandingkan dengan data uji bukan pengguna asli terang mendapatkan hasil 25 salah, data latih terang dibandingkan dengan data uji bukan pengguna asli gelap mendapatkan hasil 25 salah, data latih gelap dibandingkan dengan data uji bukan pengguna asli gelap mendapatkan hasil 20 salah.

Hasil Akurasi Tiap Pengujian

Tabel 2.3 Pengujian pengguna asli

No.	Pengujian	Hasil Pengujian
1.	Data latih terang dibandingkan dengan data uji terang	$\frac{30}{30} \times 100\% = 100\%$
2.	Data latih terang dibandingkan dengan data uji gelap	$\frac{30}{30} \times 100\% = 100\%$
3.	Data latih gelap dibandingkan dengan data uji terang	$\frac{29}{30} \times 100\% = 97\%$
4.	Data latih gelap dibandingkan dengan data uji gelap	$\frac{30}{30} \times 100\% = 100\%$
5.	Rata – rata pengujian pengguna asli	$\frac{100\% + 100\% + 97\% + 100\%}{4} = 99,25\%$

Tabel 3.3 Pengujian Bukan Pengguna Asli

No.	Pengujian	Hasil Pengujian
1.	Data latih terang dibandingkan dengan data uji bukan pengguna asli terang	$\frac{23}{30} \times 100\% = 76\%$
2.	Data latih gelap dibandingkan dengan data uji bukan pengguna asli terang	$\frac{25}{30} \times 100\% = 83\%$

3.	Data latih terang dibandingkan dengan data uji bukan pengguna asli gelap	$\frac{25}{30} \times 100\% = 83\%$
4.	Data latih terang dibandingkan dengan data uji bukan pengguna asli gelap	$\frac{20}{30} \times 100\% = 67\%$
5.	Rata – rata pengujian bukan pengguna asli	$\frac{76\% + 83\% + 83\% + 67\%}{4} = 77,25\%$

Berdasarkan hasil akurasi diatas, dapat diketahui rata – rata total akurasi pengujian dengan 60 data latih dan 62 data uji adalah 88,25%.

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah Metode Convolutional Neural Network dapat mengenali wajah meskipun dalam keadaan gelap, pada saat diterapkan pada aplikasi bengkel online (Mecha) face recognition menggunakan metode Convolutional Neural Network dapat berjalan dengan baik dan mendapatkan akurasi pengujian pengguna asli sebesar 99,25% serta akurasi pengujian bukan pengguna asli sebesar 77,25% dengan total akurasi sebesar 88,25%.

DAFTAR PUSTAKA

- [1] M. B. Satrio dan M. W. Widiatno, “Perlindungan Hukum Terhadap Data Pribadi Dalammediaelektronik (Analisis Kasus Kebocoran Datapenggunafacebook Di Indonesia),” *JCA of Law*, pp. 49-61, 2020.
- [2] A. C. Kusuma dan A. D. Rahmani, “Analisis Yuridis Kebocoran Data Pada Sistem Perbankan Di Indonesia (Studi,” *SUPREMASI: Jurnal Hukum 5.1*, pp. 46-63, 2022.
- [3] Munawir, L. Fitria dan M. Hermansyah, “Implementasi Face Recognition pada Absensi Kehadiran Mahasiswa Menggunakan Metode Haar Cascade Classifier,” *InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan*, pp. 314-320, 2020.
- [4] A. Kholik, “Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram,” *JDMSI*, pp. 10-20, 2021.
- [5] L. V. Fausett, *Fundamentals of neural networks: architectures, algorithms and applications.*, India: Pearson Education, 2006.
- [6] I. F. Alam, M. I. Sarita dan A. M. Sajidah, “Implementasi Deep Learning Dengan Metode Convolutional Neural Network Untuk Identifikasi Objek Secara Real Time Berbasis Android,” *semanTIK*, pp. 237-244, 2019.
- [7] M. Hendriani, Rais dan L. Handayani, “Penerapan Artificial Neural Network Terhadap Identifikasi Wajah Menggunakan Metode Backpropagation,” *Natural Science: Journal of Science and Technology*, pp. 203-208, 2019.
- [8] I. Mudzaki, R. Alfita dan M. Ulum, “Rancang Bangun Smart Urinoir Untuk Mendeteksi Status Dehidrasi Berbasis Image Processing Dengan Metode Jaringan Syaraf Tiruan Perceptron,” *Journal of Electrical Engineering and Computer (JEECOM)*, pp. 1-7, 2020.
- [9] A. Saleh, M. Harahap dan E. Indra, “Kombinasi Jaringan Learning Vector Quantization Dan Normalized Cross Correlation Pada Pengenalan Wajah,” *Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOM PRIMA)*, pp. 13-20, 2020.
- [10] A. A. Saputra, “Implementasi Face Recognition Pada Sistem Absensi Berbasis Android Menggunakan Metode Convolutional Neural Network (Cnn) Dan Lock Gps,” *Doctoral dissertation, Universitas Mercu Buana*, 2022.
- [11] B. T. Utomo, I. Fitri dan E. Mardiani, “Penerapan Face Recognition Pada Aplikasi Akademik Online,” *JURNAL INFORMATIK*, 2020.
- [12] S. Agustin, H. Tjandrasa dan R. H. Ginardi, “Deep Learning-based Method for Multi-Class Classification of Oil Palm Planted Area on Plant Ages Using Ikonos Panchromatic Imagery,” *Advanced Science Engineering Information Technology*, pp. 2200-2206, 2020.
- [13] I. W. S. E. P, A. Y. Wijaya dan R. Soelaiman, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *JURNAL TEKNIK ITS*, pp. 65-69, 2016.

- [14] E. Maiyana, “Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa,” *Research of Science and Informatic*, pp. 54-67, 2018.