

Sistem Pengenalan Wajah pada Keamanan Ruang Berbasis *Convolutional Neural Network*

Sunardi¹, Abdul Fadlil², Denis Prayogi^{3*}

^{1,2}Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta

³Program Studi Magister Informatika, Universitas Ahmad Dahlan, Yogyakarta

Penulis Korespondensi: *denis2107048007@webmail.uad.ac.id

Abstract

Face recognition is a biometric system that is widely applied in various fields especially in the security for identify and verify purposes. For every method of face recognition, they have a unique ways on the process with their advantages and disadvantages themselves. This study designs a face recognition system that is applied to a room security system using the Convolutional Neural Network (CNN). This method works by imitating the way nerve cells to communicate with interrelated neurons or rather mimics how artificial neural networks work in humans. The process of taking images as training data and the face recognition process using a webcam camera installed on a Raspberry pi-based device and python programming language with tensorflow library. Based on the results of research obtained using 875 data samples which were divided into 75% for training and 25% (or 219 data) for testing data produce predictions with 100% accuracy that means all data were successfully recognized.

Keywords: face recognition, security, convolutional neural network.

Abstrak

Pengenalan wajah merupakan salah satu sistem biometri yang banyak diterapkan khususnya pada bidang pengamanan untuk tujuan identifikasi dan verifikasi. Pada setiap metode pengenalan wajah memiliki cara tertentu dalam melakukan proses dengan kelebihan dan kekurangannya masing-masing. Penelitian ini merancang sistem pengenalan wajah yang diterapkan pada sistem keamanan pintu menggunakan metode Convolutional Neural Network (CNN). Metode ini bekerja dengan meniru cara sel saraf dalam berkomunikasi dengan neuron yang saling terkait atau lebih tepatnya meniru cara kerja jaringan syaraf tiruan pada manusia. Proses pengambilan citra sebagai data latih dan proses pengenalan wajah menggunakan kamera webcam yang terpasang pada perangkat berbasis Raspberry pi dan bahasa pemrograman python dengan library tensorflow. Berdasarkan penelitian dengan jumlah sampel sebanyak 875 data yang terbagi menjadi 75% untuk training dan 25% untuk data testing atau 219 data mendapatkan hasil prediksi dengan akurasi 100% yang berarti semua data dapat direkognisi dengan benar.

Kata kunci: pengenalan wajah, keamanan, convolutional neural network.

1. PENDAHULUAN

Beberapa tahun belakangan ini sistem biometri seperti retina (mata), sidik jari, suara, dan pengenalan wajah berkembang pesat dan banyak diterapkan di berbagai bidang yang salah satunya adalah pada pengamanan dalam hal identifikasi dan verifikasi. Wajah merupakan bagian tubuh manusia yang unik yang memiliki ciri tersendiri [1] sehingga dapat dijadikan sebagai identifikasi dan verifikasi seperti presensi, pembuatan Kartu Tanda Penduduk, dan sistem keamanan. Bagi manusia, mengenali wajah orang lain sangat mudah dilakukan hanya dengan melihat orang tersebut, akan tetapi

bagi mesin seperti komputer sangat sulit dilakukan dan membutuhkan program yang akurat dan teruji.

Bidang ilmu komputer yang mempelajari sistem pengenalan wajah adalah *computer vision*. *Computer vision* menggabungkan *artificial intellegent* dan *maching learning* [2]. Proses pengenalan wajah pada *computer vision* dilakukan dengan cara membandingkan suatu citra dengan citra wajah lain yang ada di dalam *database* sehingga mendapatkan kecocokan wajah [3]. Citra wajah dalam *database* sebelumnya dilakukan proses *training*, yaitu dengan memasukkan citra wajah seseorang sebanyak-banyaknya untuk mendapatkan akurasi lebih baik.

Beberapa metode klasifikasi dan ekstraksi citra yang dapat digunakan dalam proses pengenalan wajah diantaranya adalah *Support Vector Machine* (SVM), *Principal Component Analysis* (PCA), *Local Binary Pattern* (LBP), *Eigenface*, *Face Features*, dan *Convolutional Neural Network* (CNN). CNN merupakan salah satu metode dari model *deep learning* yang mampu mengenali wajah dengan akurasi yang tinggi secara statik maupun *real-time* [4].

Penelitian sebelumnya yang dilakukan oleh Al-Aidid dan Pamungkas meneliti tentang pengenalan wajah menggunakan *Haar Cascade Classifier* dengan *Local Binary Pattern Histogram*. Data latih wajah dengan dengan jarak 50-100 cm dapat melakukan deteksi wajah secara berkelompok ataupun sendiri dengan komputasi sistem yang cukup efektif [5]. Purwati dan Ariyanto melakukan penelitian terkait pengenalan wajah manusia menggunakan algoritma LBP. Dalam penelitian ini dilakukan pengembangan LBP dengan cara menggabungkan dengan Histogram Equalization, SVM, dan K-fold cross validation. Hasil penelitian ini yaitu pengenalan wajah yang optimal menggunakan dataset wajah dari JAFFE dengan resolusi yang diturunkan menjadi 92 x 112 piksel [6].

Peryanto, dkk melakukan penelitian menggunakan metode CNN yang diterapkan untuk mengklasifikasi citra dengan objek berupa bunga mawar, tulip, dan matahari. Hasil penelitian menunjukkan akurasi metode CNN tertinggi sebesar 80,36% dan rata-rata tertinggi yaitu 76,49% [7]. Siti Khotimul Wildah, dkk melakukan penelitian tentang pengenalan wajah menggunakan ekstraksi fitur pada citra wajah kualitas rendah menggunakan dataset dari *Yale Face* sebanyak 165 citra dengan 15 kelas individu. Penelitian ini memanfaatkan metode ekstraksi *Hu-Moments*, *Histogram of Oriented Gradients* (HOG) dan *Haralick*. Sedangkan klasifikasi menggunakan Logistic Regression (LR), *Linear Discriminant Analysis* (LDA), *Decision Tree Classifier*, *Random Forest Classifier*, *Gaussian NB*, KNN, dan SVM. Hasil penelitian didapatkan gabungan *Hu-Moment*, HOG, dan *Haralick* dengan klasifikasi algoritma LDA menghasilkan akurasi terbaik sebesar 79,71% [8].

Sriyati, dkk melakukan studi *literature review* terkait pengenalan wajah menggunakan CNN yang diaplikasikan pada kejujuran untuk identitas siswa. Hasil penelitian menunjukkan akurasi 98,8% pada citra wajah berukuran 64x64x3. Hasil akurasi sangat tergantung pada ukuran, kualitas, dan pola



citra yang mempengaruhi proses pengenalan wajah [9]. Andrianto Sukusvieri mengimplementasikan pengenalan wajah menggunakan metode Single Shot Detector (SSD) yang merupakan salah satu jenis CNN. Hasil penelitian didapatkan akurasi 100% pada sudut pandang hadap depan dan akurasi tertinggi 88% ketika wajah tidak hadap depan. Jarak akurasi yang baik menurut penelitian ini adalah 60-250 cm [10]. Pulung Adi Nugroho, dkk melakukan penelitian menggunakan CNN pada ekspresi manusia dengan klasifikasi senang, sedih, takut, jijik, netral, marah, dan kaget. Hasil percobaan total 35 ekspresi didapatkan 28 ekspresi dinyatakan benar dengan akurasi 80% [11]. Berdasarkan uraian di atas, penelitian pada studi ini merancang sistem pengenalan wajah menggunakan CNN yang diterapkan pada pintu untuk keamanan ruangan.

2. METODOLOGI PENELITIAN

2.1. Pengenalan Wajah

Deteksi wajah merupakan proses mengenali bentuk citra wajah pada manusia melalui pencocokan tekstur lekuk wajah dengan citra digital yang tersimpan pada *database*. Identifikasi wajah merupakan salah satu topik yang banyak diteliti pada bidang *computer vision* dengan penerapan di berbagai bidang seperti fitur keamanan pada aplikasi seperti media sosial [6], aplikasi berbasis keuangan, dan bidang lainnya yang menggunakan deteksi wajah. Terdapat tiga bagian dalam sistem pengenalan wajah [12], yaitu:

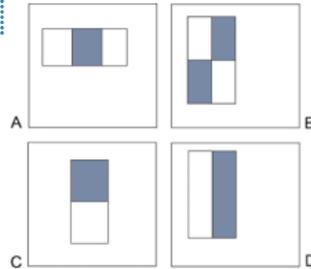
- a) Mendeteksi dan menormalisasi wajah
- b) Mengekstraksi dan normalisasi wajah dengan akurat
- c) Klasifikasi untuk hasil identifikasi

2.2. Haar Cascade Classifier

Haar Cascade atau dikenal juga dengan sebutan *Haar-like feature* merupakan algoritma yang digunakan untuk mendeteksi objek pada citra digital. Istilah ini merujuk pada fungsi matematika (*Haar Wavelet*) berbentuk kotak. Pada awalnya pengolahan citra hanya dengan melihat dari nilai RGB pada setiap titik piksel, akan tetapi ternyata metode ini tidaklah efektif. Viola dan Jones kemudian mengembangkan algoritma yang disebut *Haar-like feature* untuk memproses gambar dalam kotak-kotak dimana didalamnya terdapat beberapa piksel. Perbedaan nilai pada kotak tersebut menandakan adanya daerah yang gelap dan terang sehingga nilai inilah yang dijadikan dasar dalam memproses citra [5]. *Haar Cascade* merupakan salah satu algoritma yang mampu mendeteksi wajah manusia secara cepat, akurat, dan *realtime* karena sistem komputasinya hanya bergantung pada piksel persegi pada suatu citra [13].

Fitur Haar merupakan filter yang digunakan untuk mengekstraksi citra. Filter ini terdiri dari satu nilai interval tinggi dan rendah pada citra dua dimensi yang disebut daerah terang dan gelap. Kelebihan fitur ini adalah

kinerja komputasinya yang sangat cepat. Fitur *Haar* dapat dilihat pada Gambar 1 [14].

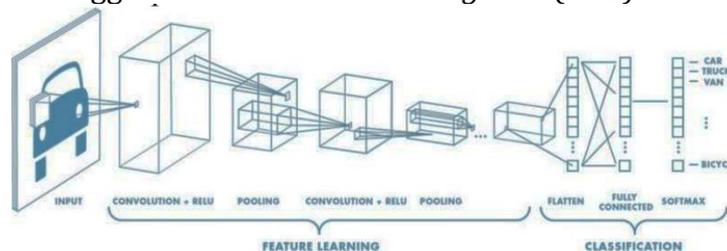


Gambar 1. Fitur Haar

Gambar 1 merupakan jenis-jenis *edge feature* pada *Haar* yang terdiri dari kotak hitam dan putih yang digunakan untuk memindai objek citra yang mengandung wajah. *Edge feature* ini akan memindai berdasarkan kondisi tingkat kecerahan berdasarkan nilai piksel dari objek *grayscale*. Fitur Haar akan memindai menggunakan 1a untuk mencari perbedaan tingkat kecerahan. Jika tidak ditemukan bentuk 1a maka fitur *Haar* akan menggunakan jenis 1b. Jika tidak ditemukan juga jenis 1b maka fitur akan mencoba memindai menggunakan fitur 1c dan 1d.

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan jenis *deep learning* yang didesain untuk mengolah data dua dimensi dengan kedalaman jaringan yang tinggi [9]. Jaringan yang dimaksud adalah jaringan syaraf tiruan yang digunakan untuk memproses citra dalam hal ini mengklasifikasi dan mengenali objek. CNN bekerja dengan meniru cara sel saraf dalam berkomunikasi dengan neuron yang saling terkait. CNN menggunakan operasi konvolusional yang menerapkan filter di setiap bagian input sebelumnya untuk mengekstraksi pola dan ini membuat CNN unik dibandingkan jaringan syaraf tiruan lainnya [7]. CNN menggunakan *Graphics Processing Unit* (GPU) untuk proses komputasi, dengan kata lain apabila menggunakan *platform* Nvidia Cuda maka pemrosesan dapat jauh lebih cepat dibandingkan menggunakan *Central Processing Unit* (CPU).



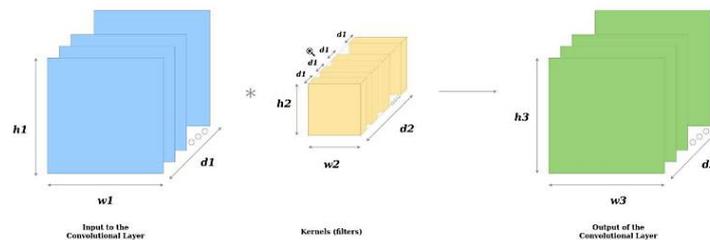
Gambar 2. Proses pada CNN

Gambar 2 menunjukkan tahapan proses pada CNN yang terdiri dari dua tahapan, yaitu *feature learning* dan *classification*. *Feature learning* terdiri dari

input layer sebagai data matriks dari suatu citra yang kemudian dilakukan abstraksi menjadi nilai *feature map convolution layer* dua dimensi. *Output* dari *convolution layer* berupa kernel akan dilakukan *pooling* untuk memperkecil atau mengurangi parameter dari *layer* tersebut. Hasil *pooling* akan menjadi *input layer* baru yang akan diproses sama seperti sebelumnya. Hal ini dapat dilakukan beberapa kali untuk meningkatkan banyaknya neuron dan beragam kombinasi variasi pada *layer*.

Lebih detail mengenai komponen utama pada CNN yaitu [15]:

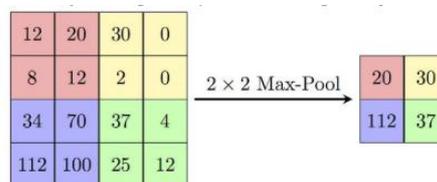
- 1) *Convolution Layer*, merupakan *backbone* dari CNN. *Layer* ini mengambil data citra, kemudian mengaplikasikan *filter* atau kernel diatas citra. Kernel atau *filter* ini mengekstraksi fitur dengan melakukan *dot-product*. Kemudian *dot-product* akan diserahkan ke *layer* berikutnya.



Gambar 3. Convolution layer

Gambar 3 merupakan proses *convolving scanning* yang terjadi pada *layer input* menggunakan kernel atau *filter* dari pojok kiri atas hingga kanan bawah pada *layer input*. Kernel merupakan representasi nilai dari *layer input* yang dilakukan proses perhitungan sehingga hasil nilainya diletakkan pada *layer output*. Antara *layer input* dan *output* memiliki dimensi yang sama antara lebar, tinggi, dan kedalaman atau banyaknya *channel*.

- 2) *Pooling Layer*, digunakan untuk mengurangi parameter dari tensor *input* sehingga dapat mengurangi perhitungan dan efisiensi. Gambar 4 mengilustrasikan cara kerja dari *pooling layer* pada *layer maximum pooling*.



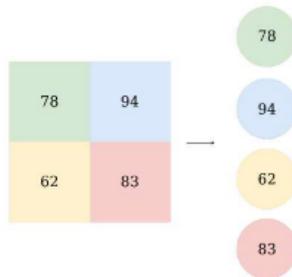
Gambar 4. Layer maximum pooling

Gambar 4 menunjukkan contoh *layer* ukuran 4x4 yang dilakukan reduksi atau mengurangi dimensi citra menjadi 2x2. Ada beberapa teknik yang digunakan pada *pooling* salah satunya adalah dengan mencari nilai maksimum pada matriks *layer*. Contoh pada Gambar 4 yang blok merah yaitu nilai pikels ukuran 2x2 kemudian dicari nilai



terbesar pada matriks tersebut yaitu nilai 20 sebagai nilai maksimum. Nilai *maxpolling* ini kemudian dimasukkan ke matriks yang baru berukuran 2x2. Begitu juga yang terjadi pada blok kuning, biru, dan hijau dicari nilai piksel terbesar pada blok tersebut sehingga didapat nilai *maxpolling* 20, 30, 122, dan 37 pada *layer* yang baru.

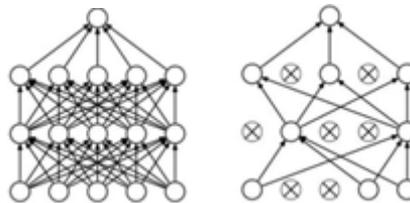
- 3) *Fully Layer*, hasil dari *convolutional layer* dan *final pooling* berupa matriks dua dimensi di-*flatened* kemudian dimasukkan ke dalam *fully connected layer*. *Flatened* merupakan proses merubah semua nilai menjadi *vector* seperti ditunjukkan pada Gambar 5.



Gambar 5. Proses *flatening*

Proses *flattening* yang mengubah *layer* dua dimensi menjadi satu dimensi berupa nilai vektor. Pada tahap ini nilai pada *fully layer* akan dijadikan urutan *array* satu dimensi yang dimulai dari sisi paling kiri atas hingga sisi paling kanan bawah pada matriks dua dimensi. Gambar 5 menunjukkan matriks $[[78,94], [62,83]]$ diubah menjadi matriks $[78,94,62,83]$.

- 4) *Dropout*, merupakan teknik di dalam jaringan syaraf tiruan untuk mencegah terjadinya *overfitting* dan mempercepat proses *learning*. *Dropout* menghilangkan sementara atau membuang neuron secara acak dalam jaringan. Setiap neuron diberikan probabilitas yang memiliki nilai nol dan satu. Contoh penerapan *dropout* dapat dilihat pada Gambar 6.



a) jaringan syaraf biasa b) Jaringan syaraf teknik *dropout*

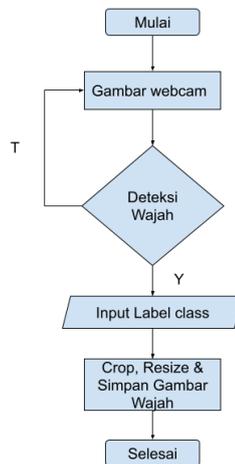
Gambar 6. Contoh penerapan *dropout*

Pada Gambar 6a terlihat contoh jumlah neuron terlalu banyak yang biasanya menyebabkan *overfitting* pada proses *training*. Teknik *dropout* dilakukan untuk membuat satu *layer* baru dan akan membuang neuron yang dianggap probabilitasnya kurang. Gambar 6b dilakukan proses *dropout* sehingga

neuron menjadi lebih sederhana. Pada CNN banyaknya neuron yang di-*dropout* dapat diatur sesuai dengan kebutuhan.

2.4. *Preprocessing* Data Wajah

Penelitian ini menggunakan dataset wajah yang diambil menggunakan kamera webcam. Sebelum melakukan proses *training* data, terlebih dahulu mendaftarkan citra wajah kedalam dataset. Ada total lima kelas wajah yang digunakan dengan masing-masing kelas terdiri dari 175 citra wajah sehingga total terdapat 875 data citra. Dari dataset ini kemudian dibagi menjadi dua bagian, yaitu data *training* menggunakan 656 (75%) dan data *testing* menggunakan 219 (25%) dari total sampel populasi dataset. Pembagian dataset diambil secara acak sehingga masing-masing kelas memiliki jumlah data *training* dan *testing* yang berbeda. Proses pengambilan dataset wajah lebih jelasnya seperti alur pada Gambar 7.



Gambar 7. Alur pengambilan dataset wajah

2.5. Proses *Training* Dataset

Proses *training* merupakan tahapan CNN untuk membentuk suatu model yang nantinya digunakan untuk proses klasifikasi. Proses ini melibatkan sampel data *training* kemudian parameter-parameter CNN yang harus diatur.

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan data sampel citra wajah sebanyak 875 yang terbagi dalam lima kelas dengan masing-masing kelas data berjumlah 175.

3.1. Hasil Klasifikasi

Proses *training* menggunakan 656 citra dengan ukuran layer 64x64, 128x128, dengan epoch 7, *BATCH_SIZE* 32 menggunakan activation relu dan *optimizer* adam menghasilkan model dari *training* data seperti Gambar 8. Proses *training* yang dilakukan menggunakan parameter yang telah diatur menghasilkan akurasi mencapai 100% dan *loss* sekitar 0,18%.

```

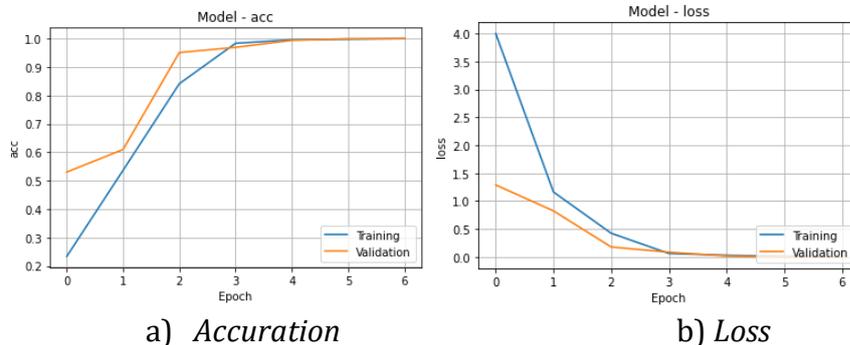
Epoch 1/7
16/16 [=====] - 12s 706ms/step - loss: 3.9967 - acc: 0.2337 - val_loss: 1.2892 - val_acc: 0.5305
Epoch 2/7
16/16 [=====] - 9s 577ms/step - loss: 1.1628 - acc: 0.5366 - val_loss: 0.8252 - val_acc: 0.6098
Epoch 3/7
16/16 [=====] - 9s 575ms/step - loss: 0.4252 - acc: 0.8415 - val_loss: 0.1797 - val_acc: 0.9512
Epoch 4/7
16/16 [=====] - 9s 595ms/step - loss: 0.0635 - acc: 0.9837 - val_loss: 0.0868 - val_acc: 0.9695
Epoch 5/7
16/16 [=====] - 9s 586ms/step - loss: 0.0312 - acc: 0.9959 - val_loss: 0.0178 - val_acc: 0.9939
Epoch 6/7
16/16 [=====] - 11s 699ms/step - loss: 0.0115 - acc: 0.9980 - val_loss: 0.0037 - val_acc: 1.0000
Epoch 7/7
16/16 [=====] - 10s 640ms/step - loss: 0.0018 - acc: 1.0000 - val_loss: 6.5523e-04 - val_acc: 1.0000
    
```

Gambar 8. Hasil proses *training* data

Dari konfigurasi *epoch* sebanyak tujuh kali, perlahan tingkat akurasi menanjak dari 0.2337 menjadi 1.0000 yang menandakan hasil *training* dengan keakuratan 100%. Sedangkan *loss* mengalami penurunan dari *epoch* kesatu 3.9967 menjadi 0.0018 menandakan tingkat kesalahan yang rendah pada saat *training*.

3.2. Evaluasi Model

Evaluasi model diperlukan untuk melihat tingkat akurasi dan *loss* dari model yang ditraining oleh CNN. Lebih jelas evaluasi model dari akurasi seperti pada Gambar 9.



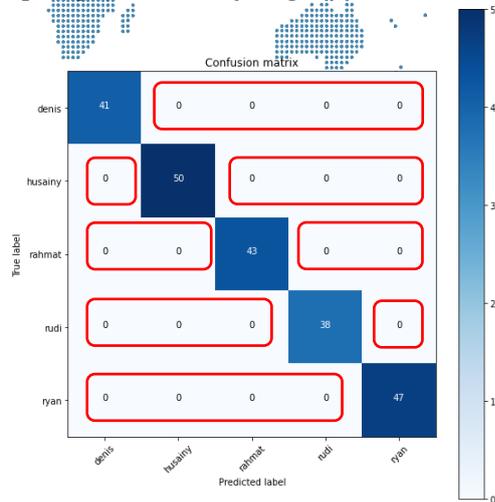
Gambar 9. Grafik evaluasi *acc* dan *loss* model *training*

Grafik pada Gambar 9a terlihat *accuracy* pada garis *training* dan *validation* sama-sama menanjak keatas dan mendekati nilai 1 atau 100%. Hal ini berarti bahwa model yang dihasilkan dari proses *training* sudah sangat baik karena untuk prediksi atau klasifikasi wajah memiliki akurasi yang baik. Gambar 9b menunjukkan *loss* dengan tingkat yang sangat rendah yaitu 0,18%. Garis *training* dan *validation* sama-sama menurun kebawah mendekati angka nol. Grafik antara *acc* dan *loss* berbanding terbalik menandakan proses *training* sudah berjalan dengan baik.

3.3. Pengujian Data menggunakan Confusion Matrix

Tahap selanjutnya adalah menguji model yang telah dilatih menggunakan data yang sebelumnya tidak pernah dilihat oleh model atau data wajah yang tidak ada di data *training*. Proses ini seperti memprediksi citra kemudian melihat dengan kemiripan yang ada pada model kemudian

menentukan citra tersebut ada di kelas mana. Gambar 10 menampilkan *confusion matrix* dari pengujian citra yang dilakukan.



Gambar 10. Pengujian data

Tabel 1. Hasil Prediksi

Kelas	Data <i>Testing</i>	Hasil Prediksi	Persentase
Denis	41	41	100
Husainy	50	50	100
Rahmat	43	43	100
Rudi	38	38	100
Ryan	47	47	100
Total Data	219	219	100

Confusion matrix pada Gambar 10 menunjukkan hasil prediksi masing-masing kelas data yang digunakan sebagai data *testing* dengan total data yang digunakan berjumlah 219 citra wajah. Dari *matrix* tersebut dapat dilihat bahwa jumlah sampel wajah dengan *True Label* “denis” sebanyak 41 citra dengan hasil prediksi juga 41 benar. Ini ditandai dengan nilai “0” pada *predicted label* di kelas yang lain. Nilai pada kotak warna merah menandakan bahwa terjadi kesalahan prediksi. Lebih jelasnya kesimpulan hasil prediksi ada pada Tabel 1.

3.4. Pengujian menggunakan webcam sebagai fitur keamanan ruangan

Implementasi pengenalan wajah menggunakan webcam dengan membandingkan antara objek wajah yang ditangkap kamera secara *real-time* kemudian dibandingkan dengan model yang telah dibuat dari proses *training* CNN. Pada implementasi ini menggunakan perangkat raspberry Pi.



a) Wajah dikenali b) Wajah terdeteksi jauh
Gambar 11. Implementasi pengenalan wajah

Dari ujicoba menggunakan video secara *real-time*, sistem mampu mendeteksi adanya citra wajah pada video dengan bantuan *Haar Cascade*. Setelah mendeteksi adanya objek wajah, kemudian sistem melakukan proses klasifikasi pengenalan wajah untuk mengenali objek seperti pada Gambar 11a yang termasuk dalam kelas “denis” dengan probabilitas kemiripan 99,93% dari model yang dibuat. Kemudian pada Gambar 11b sistem juga mendeteksi adanya objek wajah, namun terlalu jauh dari kamera sehingga menyulitkan proses pengenalan wajah untuk akses keamanan. Pada implementasi ini objek wajah hanya dapat dikenali di dalam kotak biru saja untuk menghindari deteksi objek lebih dari satu wajah.

4. SIMPULAN

Dari total data sebanyak 875 citra wajah yang terbagi dalam lima kelas wajah dengan porsi data *training* sebesar 656 (75%) dan data *testing* sebesar 219 (25%) menghasilkan akurasi dari proses *training accuracy* 100% dan *loss* 0,18%. Hasil prediksi menggunakan data testing sebanyak 219 data dengan tingkat akurasi prediksi 100%, karena semua gambar hasil uji coba bernilai benar berdasarkan *confussion matrix*. Ujicoba menggunakan webcam sebagai fitur keamanan mendeteksi dan mengenali wajah dengan baik pada posisi hadap depan berdasarkan model yang telah dilatih. Penelitian lanjutan dapat dilakukan dengan variasi data citra wajah yang lebih banyak dan berbagai variasi posisi, misal hadap kiri atau kanan. Hal ini diperlukan agar sistem dapat melakukan mendeteksi dan mengenali wajah dari berbagai sudut pandang. Penggunaan kamera untuk mengambil citra dan proses mengenali pada video dapat dilakukan menggunakan spesifikasi yang sama karena perangkat kamera yang berbeda menghasilkan nilai-nilai piksel yang berbeda antara data yang dilatih dengan objek wajah yang ditangkap secara *real-time*.

DAFTAR PUSTAKA

[1] N. Heryana, Rini Mayasari, and Kiki Ahmad Baihaqi, “Penerapan Haar

- Cascade Classification Model Untuk Deteksi Wajah, Hidung, Mulut, dan Mata Menggunakan Algoritma Viola-Jones,” *Techno Xplore J. Ilmu Komput. dan Teknol. Inf.*, vol. 5, no. 1; pp. 21–25, 2020, doi: 10.36805/technoxplore.v5i1.1064.
- [2] S. Yulina, “Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV,” *J. Politek. Caltex Riau*, vol. 7, no. 1, pp. 100–109, 2021.
 - [3] W. Anggraini, “Deep Learning untuk Deteksi Wajah yang Berhijab Menggunakan Algoritma Convolutional Neural Network (Cnn) dengan Tensorflow,” Universitas Islam Negeri Ar-Raniry Darussalam, 2020.
 - [4] V. D. WIN, “Pengenalan Wajah Menggunakan Convolutional Neural Network,” Institut Teknologi Sepuluh Nopember, 2018.
 - [5] S. Al-Aidid and D. Pamungkas, “Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram,” *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 62–67, 2018, doi: 10.17529/jre.v14i1.9799.
 - [6] R. Purwati and G. Ariyanto, “Pengenalan Wajah Manusia berbasis Algoritma Local Binary Pattern,” *Emit. J. Tek. Elektro*, vol. 17, no. 2, pp. 29–38, 2017, doi: 10.23917/emitor.v17i2.6232.
 - [7] A. Peryanto, A. Yudhana, and R. Umar, “Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation,” *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020, doi: 10.30871/jaic.v4i1.2017.
 - [8] S. K. Wildah, S. Agustiani, A. Mustopa, N. Wuryani, M. Nawawi, and R. A. Safitri, “Pengenalan Wajah Menggunakan Pembelajaran Mesin Berdasarkan Ekstraksi Fitur pada Gambar Wajah Berkualitas Rendah Face Recognition Using Machine Learning Based on Feature Extraction on Low Quality Face Images,” *INFOTECH J. Inform. Teknol.*, vol. 2, no. 2, pp. 95–103, 2021.
 - [9] Sriyati, A. Setyanto, and E. T. Luthfi, “Literature Review: Pengenalan Wajah Menggunakan Algoritma Convolutional Neural Network,” *J. Teknol. Inf. dan Komun.*, vol. 8, no. 2, 2020, doi: 10.30646/tikomsin.v8i2.463.
 - [10] A. Sucusvieri, “Implementasi Metode Single Shot Detector untuk Pengenalan Wajah,” Universitas Dinamika, 2020.
 - [11] P. A. Nugroho, I. Fenriana, and R. Arijanto, “Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia,” *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
 - [12] Y. R. W. Utami and T. Susyanto, “Unjuk Kerja Metode Klasifikasi Support Vector Machine (SVM) dengan Learning Vector Quantization (LVQ) pada Aplikasi Pengenalan Wajah,” *J. Ilm. SINUS*, pp. 9–18, 2012.
 - [13] S. Abidin, “Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab,” *J. Teknol. Elekterika*, vol. 15, no. 1, p. 21, 2018, doi: 10.31963/elekterika.v15i1.2102.
 - [14] A. R. Syafira and G. Ariyanto, “Sistem Deteksi Wajah Dengan Modifikasi

- Metode Viola Jones," *Emit. J. Tek. Elektro*, vol. 17, no. 1, pp. 26–33, 2017, doi: 10.23917/emit.v17i1.5964.
- [15] A. Peryanto, A. Yudhana, and R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Format J. Ilm. Tek. Inform.*, vol. 8, no. 2, p. 138, 2020, doi: 10.22441/format.2019.v8.i2.007.