

Pengecekan Foto Paspor Menggunakan Metode DNN dan *Facenet* Sebagai Pengenalan Wajah

Checking Passport Photos Using DNN And Facenet Methods as Facial Recognition

Winarno Sugeng¹⁾, Daniel Barus²⁾

^{1,2}Program Studi Informatika, Institut Teknologi Nasional Bandung

^{1,2}Jl. PH. H. Mustofa No. 23, Bandung 40124

winarno@itenas.ac.id¹⁾

Diterima : 11 Juli 2023 | | Revisi : 22 November 2023 | | Disetujui: 9 Desember 2023

Abstrak – Pemeriksaan foto paspor pada saat ini masih menggunakan sistem manual atau membutuhkan tenaga manusia. Dalam melakukan pemeriksaan tentunya tata cara pemeriksaan seperti ini memiliki keterbatasan yaitu *human error*, waktu operasional terbatas, serta kecepatan dan efisiensi. Tujuan dari penelitian ini adalah dapat mengenali wajah seseorang meskipun area wajah tertutup oleh objek atau aksesoris. Metode yang digunakan sebagai deteksi wajah yaitu *Deep Neural Network* dan metode untuk pengenalan wajah yaitu *FaceNet*. Untuk hasil pengujian *training model FaceNet* telah menghasilkan model terbaik dengan akurasi 97,48% terhadap 6107 *dataset* dan untuk *testing* gambar didapat akurasi gambar sebesar 97% dengan citra uji 412 dan untuk pengujian secara *realtime* didapat hasil untuk kondisi wajah normal sebesar 90%, kondisi wajah menggunakan kacamata sebesar 83%, kondisi wajah menggunakan masker sebesar 70%, kondisi wajah menggunakan topi sebesar 81% dan kondisi wajah menggunakan semua aksesoris sebesar 66%. Dengan begitu sistem telah berjalan sesuai dengan yang diharapkan dan metode *Facenet* telah dapat diimplementasikan secara *realtime* pada sistem pengenalan wajah dengan akurasi pengenalan wajah yang sangat baik.

Kata Kunci: *Deep Neural Network, FaceNet, pengenalan wajah*

Abstract – Currently, passport photos inspection still uses a manual system or requires human labor. This inspection procedure of course has limitations, namely human error, limited operational time, as well as speed and efficiency. The aim of this research is to be able to recognize a person's face even if the facial area is covered by objects or accessories. The method used as facial detection is *Deep Neural Network* and the method used for facial recognition is *FaceNet*. The testing results of the training model *FaceNet* has produced the best models with 97.48% accuracy to 6107 datasets and for image testing is obtained image accuracy of 97% with test images of 412, and for the realtime testing the result is 90% for normal facial conditions, facial condition wearing glasses is 83%, facial condition wearing mask is 70%, facial condition wearing hat is 81% and facial condition using all accessories is 66%. Therefore, the system is running as expected and *Facenet* method is able to be implemented in real time on a facial recognition system with the accuracy of facial recognition is very good.

Keywords: *Deep Neural Network, FaceNet, face recognition*

PENDAHULUAN

Pada era modern seperti saat ini perkembangan sangat cepat terutama dalam teknologi yang digunakan untuk menganalisis fisik dan kelakuan manusia (*biometric*) salah satunya yaitu sistem deteksi wajah (*face detection*). Deteksi wajah adalah *subset* dari pemrosesan gambar atau citra. Pemrosesan citra adalah utamanya sebuah teknik untuk mengkompresi, memperbaiki, atau mengekstrak informasi berharga dari suatu gambar (Hasan & Sallow, 2021). Tujuan deteksi wajah adalah menentukan keberadaan dan menandai posisi wajah, baik pada gambar maupun video, yang disebut dengan *bounding box* (Megawan &

Lestari, 2020). Deteksi wajah sering digunakan dan dimanfaatkan di berbagai perangkat teknologi digital pada zaman sekarang dengan memanfaatkan perkembangan teknologi komputer. Saat ini perkembangan aplikasi komersial yang sering menggunakan algoritma pendeteksi wajah maupun pengenalan wajah sangat pesat dalam perkembangannya (Jatmoko, Setiadi, Hartanto, & Kurniawan, 2020).

Saat ini banyak orang menggunakan aksesoris atau masker yang menutupi area wajah. Meskipun area wajah tertutupi oleh masker atau aksesoris lainnya, ciri khas dari wajah setiap orang masih dapat dikenali. Oleh

karena itu dengan adanya perkembangan teknologi, diperlukan sistem yang dapat membantu mengenali wajah seseorang meskipun wajah tersebut tertutup oleh objek atau aksesoris. Dengan adanya bantuan *deep learning*, saat ini terdapat beberapa metode untuk pengolahan citra seperti metode *Deep Neural Network* dan *FaceNet*.

Deep Neural Network (DNN) adalah algoritma jaringan saraf mendalam, yang paling umum diterapkan untuk menganalisis fitur atau variabel. DNN adalah *multilayer perceptron* yang setiap neuronnya terhubung ke semua neuron di lapisan berikutnya. DNN merupakan suatu sistem proses informasi yang memiliki karakteristik performa tertentu dalam jaringan saraf biologis secara mendalam (Errissya, Rudolf, & Pratama, 2020).

FaceNet merupakan salah satu model dengan jenis metode ekstraksi wajah yang memanfaatkan *Deep Convolutional Neural Network*. *FaceNet* dipertimbangkan sebagai model yang terbaik dikembangkan oleh *Google*. *FaceNet* menggunakan modul awal dalam blok untuk mengurangi jumlah parameter yang dapat dilatih (Efanntyo & Mitra, 2021). Salah satu fitur penting dari *FaceNet* yaitu *Triplet Los*. *Triplet Los* adalah sebuah algoritma yang diterapkan dalam *OpenFace* sebagai salah satu komponen dari proses *stochastic gradient descent* pada saat training (Weifeng Ge, Huang, Dong, & R. Scott, 2018).

Manusia merupakan makhluk individu yang tentunya memiliki ciri khas masing-masing, ciri tersebut dapat digunakan sebagai tanda pengenal bagi seseorang. Konsep identifikasi adalah mengenali seseorang dari komponen yang dimilikinya contohnya wajah, iris mata, dan sidik jari, atau dari kombinasi ketiganya (Mujib, Hidayatno, & Prakoso, 2018).

Sistem biometrika adalah teknologi pengenalan diri menggunakan bagian tubuh manusia seperti sidik jari, telinga, wajah, geometri tangan, telapak tangan, retina, gigi dan bibir (Wita & Liliana, 2022). Pengenalan wajah merupakan sebuah sistem identifikasi pribadi yang menggunakan karakteristik wajah seseorang (Sembiring, et al., 2019). Pengenalan wajah sendiri merupakan suatu cabang ilmu biometrik, yaitu suatu bidang keilmuan yang menggunakan karakteristik fisik dari seseorang untuk menentukan atau mengungkapkan identitasnya (S, Suma'inna, & Maulana, 2018). Karena wajah merupakan salah satu biometrika yang paling umum digunakan untuk mengenali seseorang. Pada sistem pengenalan wajah, citra wajah dari seorang individu akan diakuisisi sebagai citra masukan yang kemudian dicocokkan dengan sekumpulan citra wajah

yang sebelumnya telah disimpan di dalam *database* untuk mengenali citra wajah dari individu tersebut.

Berdasarkan penelitian sebelumnya oleh (Santoso & Ariyanto, 2018) dengan judul “Implementasi *Deep Learning* Berbasis Keras untuk Pengenalan Wajah”, wajah memiliki keunikan dan merupakan identitas bagi setiap manusia. Dalam pengembangannya, sistem pengenalan wajah masih memiliki permasalahan dalam faktor pencahayaan, ekspresi wajah dan perubahan atribut pada wajah, upaya yang dilakukan dalam penelitian ini adalah menggunakan *Convolutional Neural Network* (CNN) untuk mencoba mengatasi hal tersebut, Keras merupakan *framework* yang dibuat untuk mempermudah pembelajaran terhadap komputer, pengujian terhadap data *testing* memperoleh hasil yang baik dengan tingkat akurasi mencapai 98.57%. Penelitian lanjutan yang dilakukan deteksi wajah dengan menggunakan metode *Deep Neural Network* dan pengenalan wajah menggunakan *FaceNet*, serta bagaimana cara mengkombinasikan keduanya dalam melakukan pengenalan wajah meskipun wajah tertutup sebagian akibat tertutupi oleh objek atau aksesoris.

METODOLOGI PENELITIAN

FaceNet merupakan sistem pengenalan wajah yang dikembangkan oleh peneliti *Google*. *FaceNet* mengekstrak fitur wajah menjadi vektor menggunakan arsitektur *Deep Convolutional Neural Network*. *FaceNet* mengambil input berupa foto wajah dan akan mengeluarkan output berupa 128 nilai *vector* yang disebut *embedding* (Sakti, Murti, Kurniasari, & Rosid, 2022). Sehingga, jika wajah tersebut semakin mirip akan membuat nilainya semakin kecil dan sebaliknya jika wajah tersebut semakin berbeda akan menghasilkan nilai yang semakin besar dimana rumus yang digunakan untuk *FaceNet* adalah rumus *L2Norm* dan *Cosine Similarity* (Evelyn, Adipranata, & Gunadi, 2022). Berikut merupakan rumus perhitungannya:

$$\text{Cosine Similarity} = (A \cdot B) / (\|A\| * \|B\|)$$

Keterangan:

$A \cdot B$ mewakili hasil perkalian titik (*dot product*) dari vektor A dan B.

$\|A\|$ mewakili norma *Euclidean* (*magnitudo*) dari vektor A.

$\|B\|$ mewakili norma *Euclidean* (*magnitudo*) dari vektor B.

Inception V3 adalah arsitektur *CNN* yang dapat melakukan proses pengenalan citra termasuk proses

klasifikasi citra. *Inception V3* menggabungkan berbagai konvolusi dengan ukuran kernel yang berbeda-beda, karena dapat mengekstraksi fitur secara efisien. *Inception V3* juga menggunakan teknik regulasi dan reduksi untuk mencegah terjadinya *overfitting*. Arsitektur *Inception V3* ini memiliki struktur yang dalam dan kompleks. Model *Inception V3* adalah evolusi dari GoogleNet (*Inception V1*) yang didasarkan pada konvolusi 7x7 terfaktor dan dibagi menjadi 2 atau 3, 3x3 lapisan operasi konvolusi dengan tujuan meningkatkan kecepatan komputasi (Aini & Liliana, 2022).

Siamese Neural Network adalah salah satu pendekatan *Deep Learning* yang berisi bidang input untuk membandingkan dua pola dan menghasilkan satu output yang nilainya sesuai dengan kesamaan antara dua pola (Craparotta, Thomassey, & Biolatti, 2019).

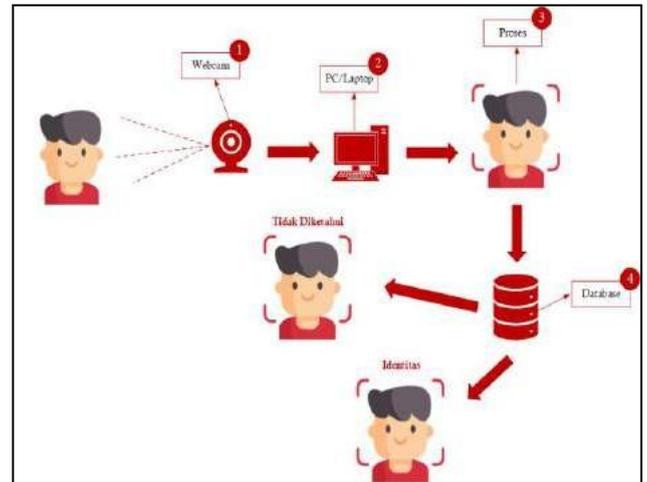
Siamese model memiliki arsitektur yang terdiri dari dua atau lebih jaringan yang memiliki struktur dan parameter yang sama. Pada saat pelatihan, kedua jaringan dijalankan pada input yang berbeda secara paralel, dan hasil keluarannya kemudian dibandingkan untuk menghasilkan output yang menunjukkan seberapa mirip kedua input tersebut.

Analisis Sistem

Metodologi pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model prototype. Pada penelitian dan perancangan dilakukan pada pembuatan sistem verifikasi foto wajah. Agar sistem yang digunakan dapat melakukan aktivitas dan difungsikan dengan baik.

Pengembang sistem mendefinisikan kebutuhan dari sistem yang akan dibangun dan mendefinisikan secara garis besar mengenai sistem yang akan dibuat. Dataset yang digunakan pada penelitian ini berupa citra wajah yang diperoleh dari *Kaggle.com* yang berjumlah 6107 citra dimensi dari citra wajah tersebut beragam dalam tiga channel warna RGB (*Red, Green, Blue*). Pada penelitian ini menggunakan 5190 citra wajah yang akan dimasukkan ke dalam proses pelatihan training dan 917 citra wajah untuk validasi dan 412 citra digunakan untuk citra uji.

Pada bagian ini dibahas mengenai rancangan yang dibangun untuk system pengecekan foto paspor. Berikut merupakan gambaran sistem beserta komponen-komponen yang dibutuhkan dapat dilihat pada Gambar 1.



Gambar 1 Gambaran Sistem

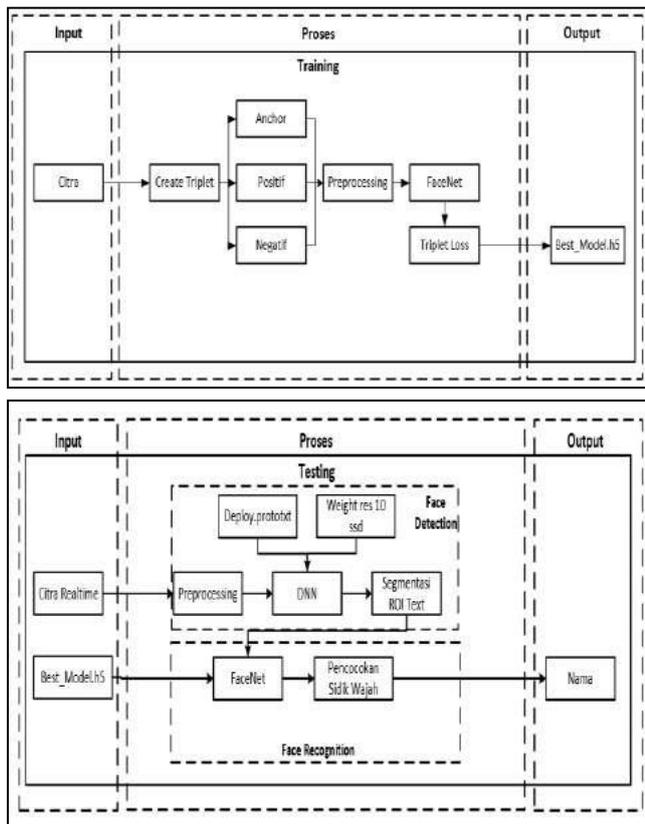
Pada Gambar 1 merupakan gambaran sistem yang dibangun. Berikut penjelasan dari beberapa komponen utama yang diperlukan dalam membangun sistem.

1. *Webcam*; *webcam* ini berfungsi untuk mengambil gambar pada wajah orang yang akan melakukan pengecekan foto paspor.
2. PC atau Laptop; PC atau laptop ini berfungsi untuk memproses gambar yang diperoleh dari *webcam* sebelumnya untuk mendeteksi wajah dan pengenalan wajah.
3. Pada proses dilakukan deteksi wajah menggunakan *Deep Neural Network* dan pengenalan wajah menggunakan *FaceNet*.
4. *Database*; *Database* ini berfungsi sebagai sistem penyimpanan ciri khas dari wajah untuk memvalidasi ciri khas tersebut dengan wajah yang telah di deteksi dan keluarannya informasi tidak diketahui atau identitas berupa nama dari wajah tersebut.

Desain Sistem

Pada desain sistem terdapat blok diagram sistem dan arsitektur model yang digunakan dalam pengenalan wajah, dan dataset yang digunakan untuk melatih model. Pada Gambar 2 dipaparkan blok diagram sistem mengenai proses *training* model dan *testing* model.

Blok diagram sistem merupakan tahapan pembuatan model *FaceNet* untuk melakukan pengenalan wajah. Selanjutnya dijelaskan dari masing-masing tahapan pada blok diagram sistem yang sudah dipaparkan.



Gambar 2 Blok Diagram Sistem

Pada tahapan *training* terdapat *input*, proses dan output dengan penjelasan berikut.

1. **Input**, dataset dipersiapkan terlebih dahulu sebagai data latih model, dataset ini terdiri dari beberapa citra wajah yang telah disegmentasi wajahnya beserta label yang berkaitan dengan masing-masing citra dan dataset yang digunakan tersebut didapatkan dari *kaggle.com*.
2. **Proses**, pada tahap proses citra data latih dilakukan proses pembuatan *triplet* untuk menghasilkan tiga masukan yaitu *anchor* sebagai citra masukan yang akan diprediksi, *positive* sebagai citra yang benar atau sesuai dengan citra *anchor*, dan negatif sebagai citra yang salah atau tidak sesuai dengan citra *anchor*. Kemudian ketiga masukan tersebut dilakukan *preprocessing* agar sesuai dengan masukan *inception V3* yaitu tinggi dan lebar citra sama dengan 299 dan *channel* nya (RGB). Selanjutnya citra yang telah dipreprocessing dimasukan ke model *FaceNet* untuk diproses agar mendapatkan prediksinya. Dasar dari model *FaceNet* yaitu *Inception V3*, oleh karena itu proses *preprocessing* dilakukan agar sesuai dengan masukan *Inception V3*. Pada model *FaceNet* juga terdapat model *siamase* yang digunakan sebagai model pembantu agar *Inception V3* dapat menerima tiga masukan. Arsitektur *siamese*

digunakan pada proses verifikasi wajah untuk menghitung jarak antara kedua fitur. Dan proses terakhir yaitu *triplet loss* untuk menyempurnakan hasil prediksi dari model *FaceNet*.

3. **Output**, pada output terdapat keluaran bobot dari model yang telah dilakukan training.

Weight dari model yang sudah di latih tersebut selanjutnya, dilakukan testing untuk memastikan *weight* model tersebut sudah dapat digunakan atau belum. Berikut penjelasan tahapan testing dari model yang sudah dilatih:

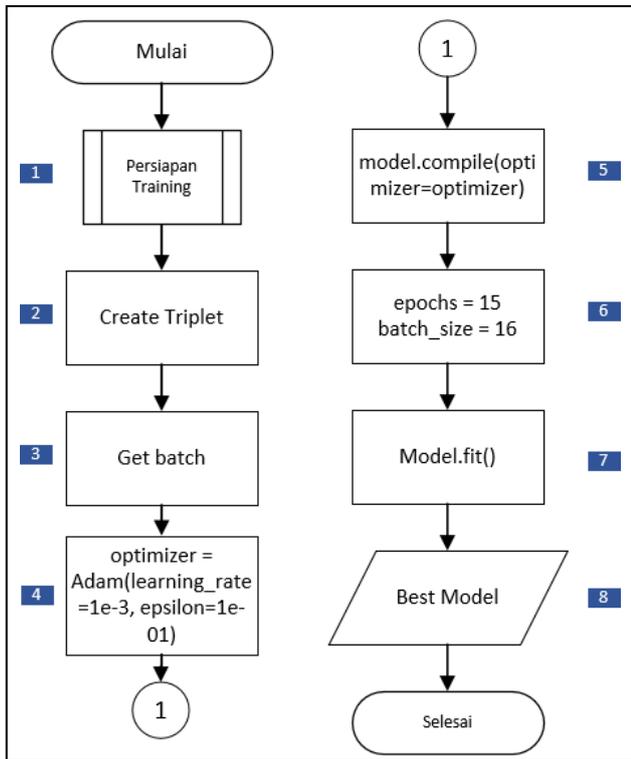
1. **Input**, pada tahap pertama terdapat input berupa citra yang didapat dari tiap frame pada *webcam* secara *realtime*.
2. **Proses**, pada tahap ini citra yang sudah diterima oleh laptop atau PC akan dilakukan proses *preprocessing* dahulu untuk diubah ukurannya dan *normalisasi*. *Normalisasi* adalah proses untuk mengganti jangkauan nilai intensitas piksel citra. Pada tahap *preprocessing* citra juga dilakukan proses konversi citra menjadi sebuah *array*. Citra yang telah di *preprocessing* tersebut diproses dengan *DNN* untuk mengetahui di mana tata letak wajah. Untuk melakukan proses metode *DNN* diperlukan *weight* yang sudah di *training* sebelumnya, terdapat 2 buah *weight* yang digunakan yaitu *deploy.prototxt* dan *res 10 ssd*. Setelah tata letak wajah diketahui dilakukan proses segmentasi *Region of Interest (ROI)* untuk memotong citra sesuai dengan ukuran wajah yang terdeteksi, selanjutnya citra tersebut dilakukan pengenalan wajah menggunakan *FaceNet* untuk diketahui ciri khas wajah dari citra tersebut. Ciri khas wajah yang telah diketahui selanjutnya dilakukan proses pencocokan dengan ciri khas wajah yang terdapat pada data *pickle* yang didapat pada proses *training*. *Pickle* adalah sebuah modul pada *standard library python*, yang dapat digunakan untuk menyimpan dan membaca data ke dalam /dari sebuah *file*.
3. **Output**, pada tahap ini terdapat keluaran berupa prediksi nama dari wajah yang dikenali dan jika wajah tidak dikenali maka akan memberikan informasi bahwa wajah tidak dikenali.

Prototyping

Pada pembentukan *prototyping* dibutuhkan beberapa proses dari awal hingga keputusan akhir sebagai keluaran sistem.

Flowchart Training FaceNet Model merupakan beberapa proses dari *training model* untuk pembuatan

model *FaceNet*. Pada *flowchart Training FaceNet Model* terdapat beberapa proses dan sub proses seperti pada Gambar 3.



Gambar 3 *Flowchart Training FaceNet Model*

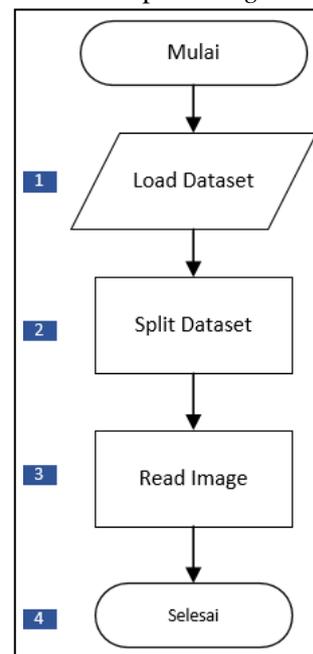
Penjelasan tahapan proses adalah sebagai berikut :

1. Pada tahap pertama persiapan *training* merupakan sebuah sub proses yang didalamnya terdapat beberapa proses yang dilakukan sebelum melakukan *training*. *Flowchart* dari proses persiapan *training* dapat dilihat pada Gambar 4.
2. *Create triplet* merupakan proses pembuatan *triplet* untuk menghasilkan tiga masukan yaitu *anchor* sebagai citra masukan yang akan diprediksi, *positive* sebagai citra yang benar atau sesuai dengan citra *anchor*, dan negatif sebagai citra yang salah atau tidak sesuai dengan citra *anchor*.
3. *Get batch* merupakan proses pengambilan citra masukan sesuai dengan jumlah *batch size* yang diatur untuk setiap satu *epoch*.
4. *Optimizer* merupakan proses optimalisasi untuk pelatihan model, pada tahap ini *optimizer* yang digunakan yaitu *adam* dengan *learning rate* $1e-4$ atau $0,0001$ dan *epsilon* sebesar $1e-01$ atau 1 . *Learning rate* diatur untuk menentukan seberapa dalam pelatihan model yang dilakukan dan *epsilon* diatur untuk menentukan seberapa besar jumlah step ketika memperbarui parameter model pada setiap iterasi.
5. *Model.compile* dilakukan untuk mengkonfigurasi

pelatihan model seperti mengatur *optimizer* yang digunakan.

6. Proses inialisasi *epochs* sama dengan 20 dan *batch_size* sama dengan 16.
7. *Model.fit()* merupakan fungsi untuk melakukan *training* sesuai dengan konfigurasi yang telah dilakukan sebelumnya
8. Terdapat keluaran yaitu model terbaik setelah *training* dengan 20 *epoch* selesai. Keluaran ini berupa file dengan ekstensi *.h5* karena *keras* dapat membaca hasil *training* model dalam bentuk *.h5*.

Flowchart sub proses persiapan *training* (Gambar 4) merupakan proses pembuatan fungsi-fungsi yang akan digunakan untuk tahap *training*.



Gambar 4 *Flowchart* sub Proses Persiapan Training

Penjelasan tahapan proses adalah sebagai berikut :

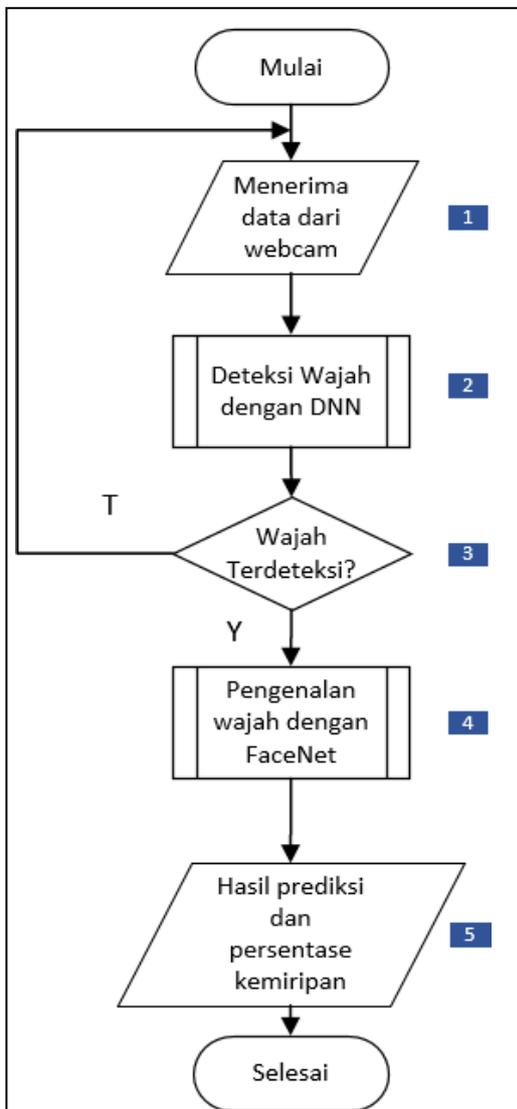
1. Memuat dataset yang telah *download* dari *kaggle.com* berupa citra wajah.
2. Proses *split dataset* untuk membagi *dataset* yang telah dimuat menjadi dua untuk *training* dan validasi
3. Membaca tiap citra yang telah dimuat

Flowchart utama sistem

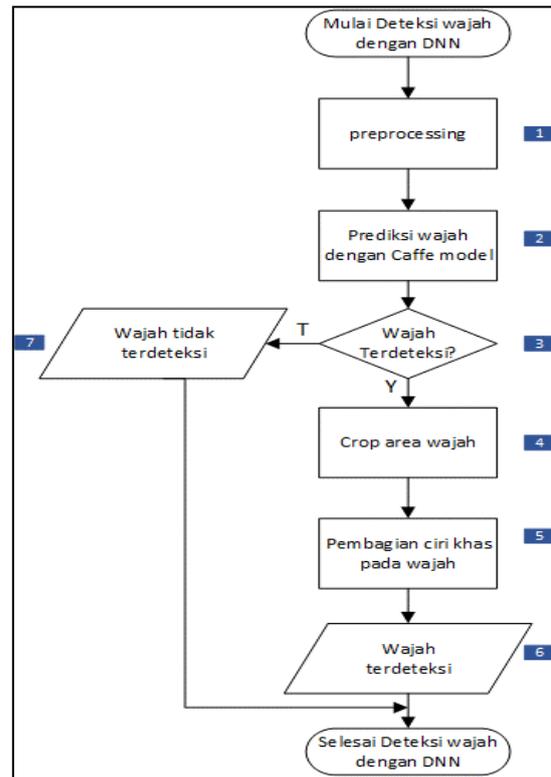
Pada Gambar 5 disampaikan *flowchart* dari sistem yang dibangun, dengan penjelasan sebagai berikut:

1. Menerima data dari *webcam*, sistem akan menerima *input* dari *webcam* berupa citra.
2. Deteksi wajah dengan *DNN*, pada tahap ini dilakukan deteksi wajah pada citra yang sudah dilakukan *preprocessing*.

3. Pada tahap ini terdapat kondisi jika wajah terdeteksi maka lanjut ke tahap 4, dan jika tidak kembali ke tahap 1.
4. Pengenalan *pickle*, pada tahap ini citra yang sudah di-crop pada area wajah, kemudian dilakukan pencocokan ciri khas wajah citra tersebut dengan ciri khas wajah yang telah disimpan pada *database* untuk pengenalan wajah.
5. *Output*, hasil informasi berupa nama identitas atau wajah tidak diketahui beserta persentasi kemiripannya.



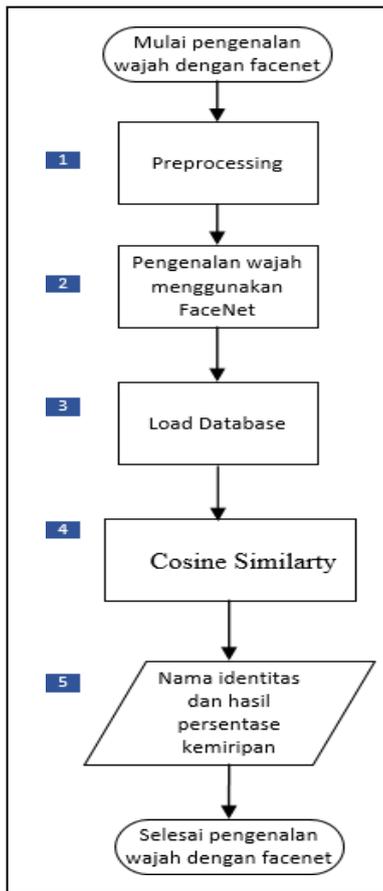
Gambar 5 Flowchart Utama Sistem



Gambar 6 Flowchart Sub Proses Deteksi DNN

Flowchart sub proses Pengenalan Wajah DNN model merupakan proses ketika DNN model bekerja. DNN model digunakan untuk mendeteksi wajah (Gambar 6). Pengenalan Wajah dengan FaceNet merupakan proses pengenalan wajah menggunakan FaceNet dengan penjelasan sebagai berikut:

1. *Preprocessing*, citra yang dideteksi dilakukan proses *preprocessing* terlebih dahulu. Pada tahap ini dilakukan perubahan ukuran citra menjadi 300x300 piksel dan dilakukan *normalisasi* agar dapat diproses oleh *Caffe Model*.
2. *Prediksi wajah dengan caffe model*, tahap ini merupakan proses prediksi wajah untuk mengetahui adanya wajah pada citra. *Caffe Model* merupakan *deep convolutional neural network* dengan menggunakan arsitektur *ResNet SSD* dan didalamnya terdapat beberapa layer konvolusi.
3. Pada tahap ini terdapat kondisi jika wajah terdeteksi maka lanjut ke tahap 4, jika tidak terdeteksi maka lanjut ke tahap 6.
4. *Crop area wajah*, tahap ini dilakukan pemotongan (*crop*) citra untuk area wajah.
5. *Wajah terdeteksi*, tahap ini memberikan keluaran bahwa terdapat wajah pada citra.
6. *Wajah tidak terdeteksi*, tahap ini memberikan keluaran bahwa tidak terdapat wajah pada citra.



Gambar 7 Flowchart Pengenalan Wajah FaceNet

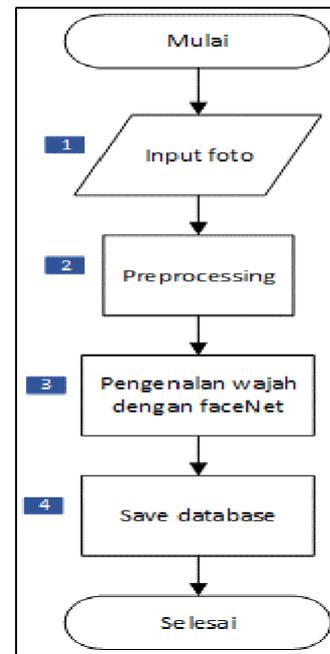
Pada Gambar 7 merupakan flowchart sub proses pengenalan wajah Facenet. Penjelasan dari Flowchart sub proses Pengenalan Wajah dengan FaceNet adalah sebagai berikut:

1. *Preprocessing*, pada tahap preprocessing dilakukan konversi citra ke RGB (*Red, Green, Blue*) dan dilakukan pengubahan ukuran pada citra menjadi 160x160 piksel. Kemudian citra tersebut dijadikan ke dalam bentuk *array* dan dilakukan *normalisasi* karena FaceNet menerima masukan yang sudah di *normalisasi*. Tahap *preprocessing* dan proses *load database* dilakukan secara bersamaan dan pada *database* yang telah di *load* dilakukan pengambilan nama identitas beserta *signature* wajah.
2. Pengenalan wajah dengan FaceNet, citra yang telah dilakukan *preprocessing* tersebut kemudian diproses oleh FaceNet untuk diketahui ciri khas dari wajah tersebut. Pada FaceNet merupakan *deep Convolutional Neural Network* yang didalamnya terdapat kumpulan layer konvolusi dan *fully connected layer*.
3. *Load database*, tahap ini dilakukan untuk memuat *database* yang sudah dibuat dalam

bentuk file *pickle*. Tujuannya untuk untuk menyimpan ciri khas wajah yang selanjutnya akan dilakukan proses pencocokan.

4. Proses *cosine similarity* dengan rumus $(A \cdot B) / (\|A\| * \|B\|)$. A merupakan hasil prediksi dari facenet dan B merupakan *signature* yang di dapat dari *database*. Proses ini dilakukan sebanyak jumlah *signature* yang terdapat pada *database* dan dicari selisih perbandingan kemiripannya.
5. *Output*, hasil dari selisih terkecil kemudian di jadikan persen dan di ambil nama identitas wajahnya.

Pada Gambar 8 Flowchart pembuatan *database* sebagai data foto paspor yang akan di cocokkan dengan wajah seseorang yang akan dikenali.



Gambar 8 Flowchart Pembuatan Database

Penjelasan dari flowchart pembuatan *database* adalah sebagai berikut:

1. *Input foto*, tahap awal pada pembuatan *database* ini dilakukan memasukan citra wajah yang akan digunakan untuk pencocokan wajah.
2. *Preprocessing*, pada tahap preprocessing dilakukan konversi citra ke RGB (*Red, Green, Blue*) dan dilakukan pengubahan ukuran pada citra menjadi 160x160 piksel. Kemudian citra tersebut dijadikan ke dalam bentuk *array* dan dilakukan *normalisasi* karna FaceNet menerima masukan yang sudah dinormalisasi
3. Pengenalan wajah dengan FaceNet, citra yang telah dilakukan *preprocessing* tersebut kemudian

diproses oleh *FaceNet* untuk diketahui ciri khas dari wajah tersebut. Pada *FaceNet* merupakan *deep Convolutional Neural Network* yang didalamnya terdapat kumpulan layer konvolusi dan fully connected layer.

4. *Save database*, pada tahap akhir, ciri khas wajah pada citra akan disimpan dalam bentuk file *pickle*.

HASIL DAN PEMBAHASAN

Dalam pembuatan sistem pengecekan foto paspor ini diperlukan beberapa bahasa pemrograman yaitu bahasa *python* yang digunakan untuk memprogram deteksi dan pengenalan wajah, *html* digunakan sebagai tampilan untuk deteksi dan pengenalan wajah.

Software yang digunakan untuk membangun sistem yaitu *Pycharm* dengan bahasa *python*. Adapun *hardware* yang digunakan yaitu laptop dan *webcam*.

Pengujian

Untuk sistem ini, diterapkan sistem pengujian alfa dan beta. Pengujian alfa yaitu teknik pengujian yang berfokus pada sisi fungsionalitas tiap sub komponen yang digunakan. Sedangkan pengujian beta yaitu pengujian keseluruhan sistem apakah sudah sesuai dengan yang diharapkan atau belum.

Pengujian Alfa

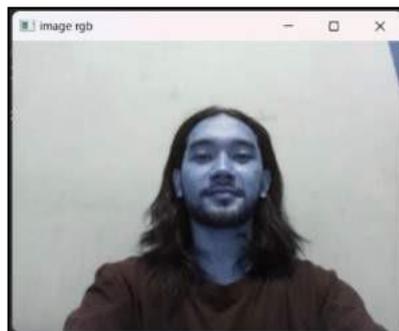
Pengujian *alfa* pada proses deteksi wajah dengan *DNN* dengan penjelasan sebagai berikut: Tahap *preprocessing*, citra yang dideteksi dilakukan proses *preprocessing* terlebih dahulu. pada tahap ini dilakukan perubahan ukuran citra menjadi 300x300 piksel dan dilakukan normalisasi agar dapat diproses oleh *Caffe Model*.



Gambar 9 Resize

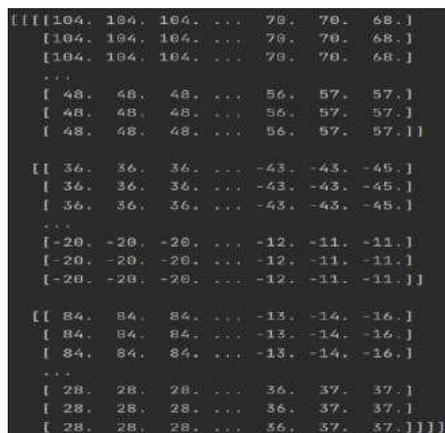
Selanjutnya pada Gambar 9 terdapat proses konversi citra yang telah di *resize* sebelumnya menjadi RGB (*Red, Green, Blue*) karena warna dari citra asli yaitu BGR (*Blue, Green, Red*). dapat dilihat pada

Gambar 10 merupakan hasil konversi citra menjadi RGB.



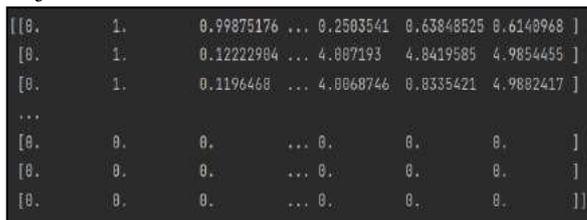
Gambar 10 Citra to RGB

Citra yang telah dikonversi ke RGB selanjutnya dilakukan normalisasi untuk mengurangi resolusi citra. Pada Gambar 11 merupakan hasil dari normalisasi citra RGB.



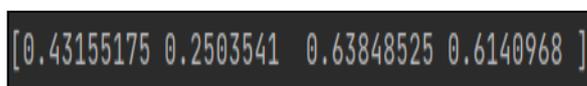
Gambar 11 Normalisasi

Pada proses *DNN* dilakukan prediksi wajah dengan menggunakan *caffe model*, citra yang diproses adalah citra yang sebelum sudah di *normalisasi*. Dapat dilihat pada Gambar 12 hasil dari prediksi wajah oleh *DNN*.



Gambar 12 Hasil Prediksi Wajah

Selanjutnya hasil prediksi tersebut dilakukan penyederhanaan dari hasil prediksi wajah untuk mengetahui tata letak wajah, Gambar 13 merupakan hasil dari penyederhanaan prediksi wajah untuk pembuatan box.



Gambar 13 Hasil Box Prediksi

Setelah itu dilakukan pencarian koordinat untuk segmentasi citra berdasarkan wajah yang dideteksi. Gambar 14 merupakan koordinat untuk segmentasi citra.



Gambar 14 Koordinat untuk ROI

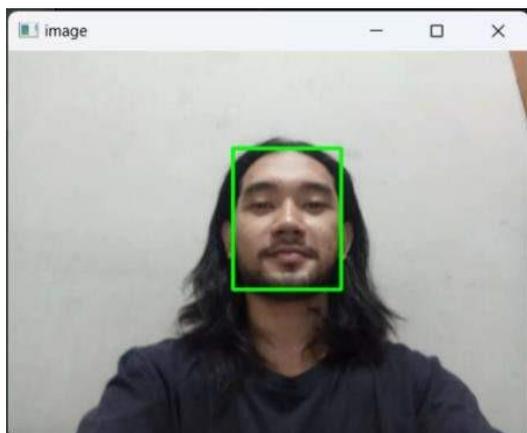
Terdapat kondisi jika adanya wajah pada citra maka dilakukan segmentasi citra, jika tidak kembali mencari area wajah pada citra.

1. Pada Gambar 15 dilakukan *Region of Interest* (ROI) atau segmentasi citra sesuai koordinat yang telah didapatkan sebelumnya untuk area wajah,



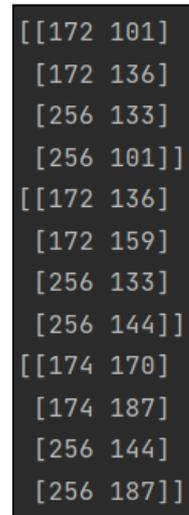
Gambar 15 Hasil *Region of Interest*

Selanjutnya pada Gambar 16 dilakukan penggambaran garis (*draw contours*) fungsinya untuk mengidentifikasi dan menandai kontur atau batas-batas pada area wajah.



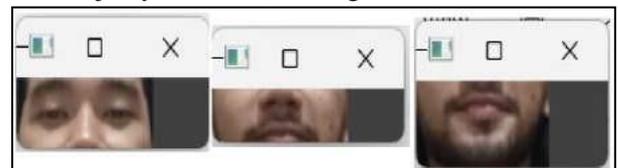
Gambar 16 Hasil *Bounding Box*

2. Selanjutnya wajah yang telah di dapat sebelumnya dilakukan proses pencarian ciri khas wajah dan didapat hasil koordinat seperti pada Gambar 17.



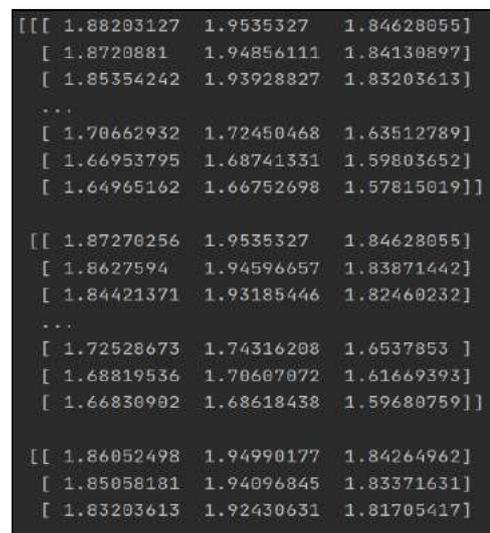
Gambar 17 Koordinat Ciri Khas Wajah

Pada Gambar 18 tahap koordinat yang telah di dapatkan sebelumnya dilakukan *crop* pada area ciri khas wajah yaitu mata, hidung dan mulut.



Gambar 18 Hasil *Crop Facial*

Pada tahap *preprocessing* dilakukan konversi citra ke RGB (*Red, Green, Blue*) dan dilakukan pengubahan ukuran pada citra dari 300x300 menjadi 160x160 piksel. Kemudian citra tersebut dijadikan ke dalam bentuk *array* dan dilakukan normalisasi. *FaceNet* menerima masukan yang sudah dinormalisasi. Pada Gambar 19 merupakan hasil *resize* normalisasi *FaceNet*.



Gambar 19 Hasil *Resize* Normalisasi

Selanjutnya terdapat proses pengenalan wajah, pada tahap ini dilakukan pengenalan dengan *FaceNet*, citra yang telah dilakukan *preprocessing* tersebut kemudian diproses oleh *FaceNet* untuk diketahui ciri khas dari wajah tersebut. Pada Gambar 20 merupakan hasil dari proses sidik wajah dan menghasilkan 128 angka yang didapat dari *signature* wajah.

```
eye_encode [ 0.00653458 -0.17846875 0.13687832 -0.09318151 -0.00819915 -0.03775473
-0.02110612 -0.03398728 -0.03677759 -0.02788531 0.08551217 -0.00659333
0.12913178 -0.04674381 -0.05259144 0.02978058 0.13016791 -0.14645004
-0.04262483 0.02132104 0.01794111 -0.00582767 -0.00550198 -0.02415233
0.05068891 -0.01239194 0.12347203 0.08628596 0.05878761 -0.08467449
-0.00674377 0.01022637 0.03884781 -0.08603593 0.02502663 -0.03073104
0.21535689 0.02276021 -0.01336062 0.06716739 0.07263171 -0.04520815
0.09661666 0.01154576 -0.20353524 -0.01446353 0.06478331 -0.06823717
0.05338336 0.04388276 -0.11026514 -0.01486781 0.06506912 -0.13488309
0.01244066 0.06886955 0.06420277 0.1089837 0.09242971 -0.0967571
-0.14323747 -0.01098796 -0.00546815 0.21681201 -0.00585847 0.16711992
0.05902215 -0.05978266 0.10367002 -0.10822874 -0.0108196 -0.16605474
0.0030943 -0.06238519 0.00109822 0.06337869 -0.00745358 -0.04771878
-0.01659746 -0.09684101 -0.07641941 -0.09048287 0.04927982 0.0081241
```

Gambar 20 Hasil Prediksi *FaceNet*

Gambar 21 hasil pengenalan wajah beserta nama pemilik dari wajah dan ciri khas yang terdiri dari mata hidung dan mulut.

```
Name ... Mouth
0 Daniel ... [-0.11731029, -0.10705087, -0.024059627, -0.03...
1 fakhrudin ... [-0.26363948, -0.048696745, 0.11973419, -0.062...
2 Galih ... [-0.089550786, -0.07016549, 0.0688809, -0.0326...
[3 rows x 5 columns]
```

Gambar 21 Upload Database

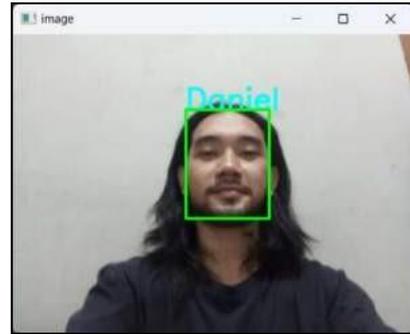
1. Pada pengujian ini dilakukan pencocokan ciri khas wajah secara *realtime*. Pada saat *webcam* mengambil ciri khas wajah seseorang kemudian dilakukan pencocokan value wajah yang terdapat pada *database*. Untuk hasil pencocokan tersebut di dapatkan dari selisih *signature* dan *value* $\sum_{i=1}^{28} \sqrt{(s - v)^2}$ dengan hasil nilai yang terendah. Gambar 22 merupakan hasil nilai pencocokan *signature* wajah, mata, hidung dan mulut.

```
0.10085630416870117
0.6253153383731842
0.47065597772598267
0.7738911807537079
```

Gambar 22 Hasil Selisih Database dan Prediksi Sebagai Pencocokan

Jika nilai selisih yang di dapat sebelumnya terendah maka *signature* wajah cocok dengan *value* yang terdapat pada *database*. Pada Gambar 23

merupakan kondisi dimana pencocokan sesuai dengan *database*.



Gambar 23 Hasil Pencocokan

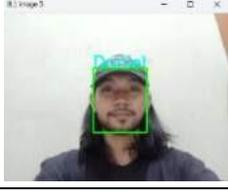
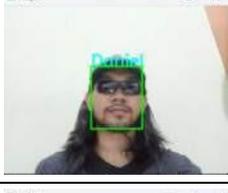
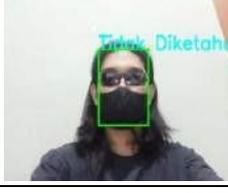
Pengujian Beta

Pada pengujian beta terdapat beberapa hasil pengujian secara *realtime* dan menggunakan citra. Untuk pengujian secara *realtime* terdapat 67 kondisi wajah yang berbeda seperti menggunakan masker, topi, kacamata dan tanpa aksesoris dan untuk pengujian menggunakan citra dilakukan dengan menggunakan 412 *dataset* wajah yang di dapat di *Kaggle.com*.

Pada Tabel 1 merupakan contoh hasil pengujian secara *realtime*, terdapat masing masing kelas yang didalamnya terdapat citra uji, kondisi wajah, kelas target, dan hasil prediksi benar atau salah. Hasil pengujian berdasarkan Tabel 1 dan Tabel 2 terdapat kesalahan prediksi pada kondisi menggunakan masker dan topi memiliki hasil tidak diketahui. Kesalahan terjadi karena kurangnya pencahayaan pada area wajah sehingga pengambilan citra dari *webcam* kurang jelas.

Tabel 2 Pengujian Lanjutan Pengenalan Wajah *Realtime*

Citra	Kelas Target	Kelas prediksi	Hasil
	Daniel	Daniel	Benar
	Daniel	Daniel	Benar
	Daniel	Daniel	Benar

Citra	Kelas Target	Kelas prediksi	Hasil
	Daniel	Daniel	Benar
	Daniel	Daniel	Benar
	Daniel	Daniel	Benar
	Daniel	Tidak diketahui	Salah
	Daniel	Tidak diketahui	Benar
	Daniel	Tidak diketahui	Benar
	Daniel	Daniel	Benar

Tabel 3 Hasil Persentase Pengujian Sistem dari Berbagai Kondisi

Kondisi	Jumlah Percobaan	Jumlah Benar	Persentase keberhasilan
Wajah normal tanpa aksesoris	10	9	90%
Wajah menggunakan aksesoris kaca mata	12	10	83%
Wajah menggunakan aksesoris masker	10	7	70%
Wajah menggunakan aksesoris topi	11	9	81%
Wajah menggunakan semua aksesoris	24	16	66%

Dapat dilihat dari hasil pengujian pada Tabel 3, untuk kondisi wajah menggunakan masker didapat persentase keberhasilan 70% dengan 10 kali percobaan, lalu untuk kondisi wajah menggunakan topi di dapat persentase keberhasilan 80% dengan 10 kali percobaan dan untuk kondisi wajah menggunakan semua aksesoris di dapat persentase keberhasilan 60% dengan 10 kali percobaan. Terjadi penurunan persentase keberhasilan pada kondisi wajah menggunakan aksesoris karena pada area wajah tertutupi oleh kacamata, masker dan juga topi sehingga pengambilan ciri khas pada wajah seseorang kurang akurat. Tabel 3 merupakan hasil persentase pengujian system dari berbagai kondisi. Untuk kondisi wajah normal tanpa aksesoris memiliki persentase keberhasilan 90% dengan 10 kali percobaan. Untuk kondisi wajah menggunakan kacamata di dapat persentase keberhasilan 80% dengan 10 kali percobaan.

KESIMPULAN

Pada penelitian ini telah dilakukan pelatihan model *FaceNet* untuk pengenalan wajah dengan menggunakan *dataset CelebA* yang berjumlah 6107 yang di dapat dari www.kaggle.com. Agar sistem dapat mengenali wajah yang tertutup aksesoris ataupun tidak maka diimplementasikan metode *Deep Neural*

Networks dan *FaceNet* sebagai pengenalan wajah. Dengan metode tersebut, sistem dapat melakukan pengenalan wajah secara *realtime* maupun gambar. Untuk pengujian secara *realtime* citra yang telah di dapat dari *webcam* kemudian dibandingkan dengan *database* dengan menggunakan perhitungan *cosine similarity*. Untuk pengujian secara *realtime* terdapat masalah pada saat posisi wajah tertentu seperti dari samping dan pengaruh cahaya dari *webcam*, masih kurang akurat dalam melakukan pengenalan wajah. Dengan menyampingkan masalah tersebut, proses *training model FaceNet* telah menghasilkan model terbaik yang memiliki akurasi sebesar 97,48% terhadap 6107 untuk pengujian secara realtime akurasi pengenalan wajah menggunakan *FaceNet* sudah cukup baik.

DAFTAR PUSTAKA

Aini, N., & Liliana, D. Y. (2022). Prediksi Gender Berdasarkan Citra Mata Menggunakan Metode Convolutional Neural Network, Inception dan MobileNet. *Buletin Poltanesa*, 226-232.

Craparotta, G., Thomassey, S., & Biolatti, A. (2019). A siamese neural network application for sales forecasting of new fashion products using heterogeneous data. *International Journal of Computational Intelligence Systems*, 1537-1546.

E. R., R. S., & Pratama, Y. (2020). Evaluasi Pembangunan Sistem Pakar Penyakit Tanaman Sawit dengan Metode Deep Neural Network (DNN). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 1206-1215.

Efanntyo, & Mitra, A. R. (2021). Perancangan Aplikasi Sistem Pengenalan Wajah Dengan Metode Convolutional Neural Network (CNN) Untuk Pencatatan Kehadiran Karyawan. *Jurnal Instrumentasi dan Teknologi Informatika*, 1-11.

Evelyn, Adipranata, R., & Gunadi, K. (2022). Sistem Presensi Mahasiswa Menggunakan Face Recognition Dengan Metode *FaceNet* Pada Android. *Jurnal Infra*, 2-8.

Hasan, R. T., & Sallow, A. B. (2021). Face Detection and Recognition Using OpenCV. *Journal of Soft Computing and Data Mining*, 86-97.

Jatmoko, C., Setiadi, D. M., Hartanto, D., & Kurniawan, A.

F. (2020). Uji Implementasi Algoritma Viola-Jones Dalam Pengenalan Wajah. *Dinamik*, 68-76.

Megawan, S., & Lestari, W. S. (2020). Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 261-267.

Mujib, K., Hidayatno, A., & Prakoso, T. (2018). ENGENALAN WAJAH MENGGUNAKAN LOCAL BINARY PATTERN (LBP) DAN SUPPORT VECTOR MACHINE (SVM). *Transient*, 123.

S, A. B., Suma'inna, & Maulana, H. (2018). Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA). *Jurnal Teknik Informatika*, 166-175.

Sakti, D. M., Murti, W. S., Kurniasari, A., & Rosid, J. (2022). Face Recognition Dengan Metode Haar Cascade dan *FaceNet*. *Indonesian Journal of Data and Science*, 30-34.

Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah. *Jurnal Teknik Elektro*, 15-21.

Sembiring, D. R., Aritonang, M. A., Pane, M. J., Laia, A. D., Waruwu, E., & Azmi, F. (2019). ANALISIS ACCURATE LEARNING BACKPROPAGATION NEURAL NETWORK PADA PENGENALAN WAJAH. *Jurnal Mantik Penusa*, 70-73.

Weifeng Ge, Huang, W., Dong, D., & R. Scott, M. (2018). Deep Metric Learning with Hierarchical Triplet Loss. *CVF*, 272-288.

Wita, D. S., & Liliana, D. Y. (2022). Klasifikasi Identitas Dengan Citra Telapak Tangan Menggunakan Convolutional Neural Network (CNN). *Jurnal Reayasa Teknologi Informasi (JURTI)*, 1.