

SISTEM PENGENALAN DAN VERIFIKASI WAJAH MENGUNAKAN TRANSFER LEARNING BERBASIS RASPBERRY PI

Muhammad Farhan Aditama⁽¹⁾, Munnik Haryanti ST, MT⁽²⁾
Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas
Dirgantara Marsekal Suryadarma
email : farhanaditwork14@gmail.com

ABSTRACT

Technology is currently developing rapidly, especially in artificial intelligence technology or AI. One part of artificial intelligence is deep learning. Deep learning is very powerful and capable of solving various problems with big data such as images, text and sound so that deep learning is widely used in research and industry needs. Face recognition and verification is an example of the application of artificial intelligence in the field of computer vision. In making facial recognition and verification using deep learning with the transfer learning method or pre-trained models. Basically, the architecture of the model can be made by yourself (scratch). However, in making a model architecture that is made by yourself, it takes a lot of time to find a suitable architecture in solving certain problems, the model that is made by yourself also does not necessarily get a better level of accuracy and the inference process also takes a long time and requires sufficient data. so that the computer can recognize the image properly. The model architecture to be used is MobileNetV1 for face recognition and Facenet for face verification. Tests were carried out using a Raspberry PI with a camera-to-face distance of 40 cm, 70 cm and 100 cm and the closest distance obtained had high accuracy and an average accuracy rate of 83% with a response time of 1.26 seconds.

Keywords : *Transfer Learning, Raspberry PI, MobileNetV1, dan Facenet.*

ABSTRAK

Teknologi saat ini sudah berkembang pesat terutama pada teknologi kecerdasan buatan atau Artificial Intelligence (AI). Salah satu bagian dari kecerdasan buatan adalah deep learning. Deep learning sangat kuat dan mampu untuk memecahkan berbagai permasalahan dengan data yang besar seperti citra, text dan suara sehingga deep learning banyak digunakan pada penelitian dan kebutuhan di dalam industri. Pengenalan dan verifikasi wajah merupakan salah satu contoh penerapan dari kecerdasan buatan di bidang computer vision. Dalam pembuatan pengenalan dan verifikasi wajah menggunakan deep learning dengan metode transfer learning atau pre-trained model. Pada dasarnya, arsitektur pada model dapat dibuat sendiri (scratch). Namun, dalam pembuatan arsitektur model yang dibuat sendiri membutuhkan waktu yang banyak untuk mencari arsitektur yang cocok dalam menyelesaikan masalah tertentu, model yang dibuat sendiri juga belum tentu mendapatkan tingkat akurasi yang lebih bagus dan waktu proses inferensinya juga membutuhkan waktu yang lama serta dibutuhkan data yang cukup banyak agar komputer dapat mengenali gambar tersebut dengan baik. Arsitektur model yang akan digunakan yaitu MobileNetV1 untuk pengenalan wajah dan Facenet untuk verifikasi wajah. Pengujian yang dilakukan menggunakan Raspberry PI dengan jarak kamera ke wajah 40 cm, 70 cm dan 100 cm dan diperoleh jarak terdekat memiliki akurasi yang tinggi serta rata – rata tingkat akurasi 83% dengan respon time 1,26 detik.

Kata kunci : *Transfer Learning, Raspberry PI, MobileNetV1, dan Facenet.*

1. PENDAHULUAN

Teknologi saat ini sudah berkembang pesat terutama pada teknologi kecerdasan buatan atau *Artificial Intelligence* (AI). AI adalah kecerdasan yang dimiliki oleh mesin yang dapat bertindak seperti layaknya manusia. Salah satu bagian dari kecerdasan buatan adalah *deep learning*. *Deep learning* merupakan metode *learning* yang memanfaatkan *Artificial Neural Network* (ANN) yang berlapis-lapis (*multi layer*) dan *deep learning* dapat dipandang sebagai gabungan *machine learning* dengan ANN (*Artificial Neural Network*).^[7] *Deep learning* sangat kuat dan mampu untuk memecahkan berbagai permasalahan dengan data yang besar seperti citra, text dan suara sehingga *deep learning* banyak digunakan pada penelitian dan kebutuhan di dalam industri.

Pengenalan dan verifikasi wajah merupakan salah satu contoh penerapan dari kecerdasan buatan di bidang *computer vision* untuk dapat mengenali dan memverifikasi wajah seorang. Wajah merupakan salah satu bagian dari manusia yang memiliki ciri-ciri berbeda. Karena wajah manusia mempresentasikan sesuatu yang kompleks, sehingga pengembangan model komputasi yang ideal untuk pengenalan wajah adalah sesuatu hal yang sangat penting.^[3] Saat ini

wajah merupakan salah satu ciri pada manusia yang dapat digunakan sesuai dengan kebutuhan. Salah satunya kebutuhan pada bidang keamanan untuk identifikasi data diri seorang, mengisi absensi dengan menggunakan pemindai wajah dan akses untuk membuka pintu menggunakan wajah.

Pada penelitian ini, pengenalan dan verifikasi wajah akan dibuat menggunakan *deep learning* dengan metode *transfer learning* atau *pre-trained model*. Pada dasarnya, arsitektur pada model dapat dibuat sendiri (*scratch*). Namun, dalam pembuatan arsitektur model yang dibuat sendiri membutuhkan waktu untuk mencari arsitektur yang cocok dalam menyelesaikan masalah tertentu, model yang dibuat sendiri juga belum tentu mendapatkan tingkat akurasi yang lebih baik dan waktu proses inferensinya juga membutuhkan waktu yang lama serta dibutuhkan data yang cukup

banyak agar komputer dapat mengenali gambar tersebut dengan baik.

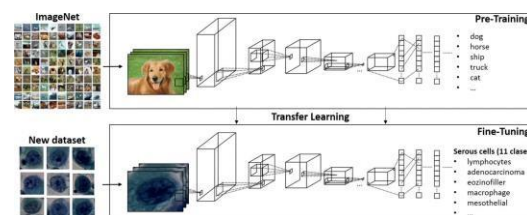
2. LANDASAN TEORI

2.1 Haar Cascade Classifier

Haar Cascade Classifier merupakan metode yang digunakan dalam pendeteksian obyek. Metode ini juga dikenal dengan metode *viola jones* dikarenakan diperkenalkan oleh Paul Viola dan Michael Jones untuk mendeteksi wajah. Metode ini mendeteksi objek dalam gambar dengan menggabungkan empat kunci utama yaitu *haar feature*, *integral image*, *adaptive boosting* dan *cascade classifier*.

2.2 Transfer Learning

Transfer learning adalah suatu metode yang menggunakan model yang sudah ada sebagai dasar untuk membuat model baru untuk menyelesaikan masalah yang lain dengan cara menjadi starting point, memodifikasi, dan mengupdate parameter yang di perlukan sehingga terbentuk sebuah model baru.^[1]



Gambar 1. Ilustrasi Transfer Learning

Model yang sudah terlatih ini disebut dengan *pre-trained model*. *Pre-trained model* biasanya sudah dilatih pada dataset yang besar dan merupakan dataset benchmark, sehingga kualitas *pre-trained model* harusnya sudah sangat baik. Secara umum ada dua cara menggunakan model *pre-trained* untuk *transfer learning* yaitu *fixed feature extraction* dan *fine tuning*.

2.3 MobileNetV1

MobileNetV1 merupakan model yang memiliki ukuran kecil baik dari jumlah parameter maupun ukuran model yang dihasilkan. MobileNetV1 adalah sebuah arsitektur model yang dikembangkan untuk pengembangan aplikasi pada perangkat mobile ataupun perangkat lain yang memiliki keterbatasan sumber daya

perangkat keras dengan mengurangi ukuran dan kompleksitas model menggunakan depthwise separable convolutions. Penggunaan depthwise separable convolutions pada MobileNetV1 mengurangi jumlah parameter lebih dari 7 kali lipat dari penggunaan convolution standar, dengan penurunan akurasi hanya 1% pada ImageNet.^[9]

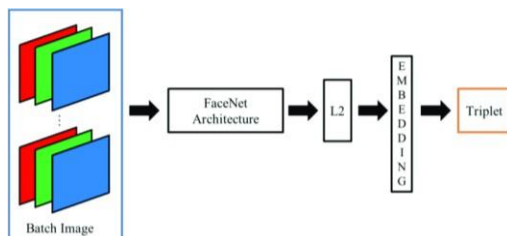
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Gambar 2. Arsitektur MobileNetV1

2.4 Facenet

Facenet merupakan sebuah model yang dikembangkan oleh Schroff, et al. Facenet dapat digunakan untuk face recognition, verification dan clustering. FaceNet termasuk kedalam deep convolutional neural network dengan 22 layer yang akan menghasilkan output embedding 128-D. Loss function yang digunakan pada FaceNet biasa disebut dengan 'triplet loss'.



Gambar 3. Blok Diagram Facenet

Pada Gambar 3, arsitektur CNN yang digunakan pada facenet merupakan

arsitektur 'Inception-ResNet-v1' dari GoogleNet. Arsitektur dan parameter pada setiap layer jaringan Googlenet dapat dilihat pada Gambar 4.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2	64	192					112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×512	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×512	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Gambar 4. Arsitektur dan Parameter GoogleNet

Selama proses training, deep network akan melakukan ekstraksi dan menghasilkan fitur yang ada pada wajah yang akan dikonversi dalam bentuk embedding 128-D. Gambar wajah yang sama atau mirip akan memiliki nilai embedding yang mendekati sama dan wajah yang berbeda akan memiliki nilai embedding yang berbeda. Untuk mencari tingkat kemiripan dari embedding yang dihasilkan oleh wajah maka digunakan matriks persamaan Cosine Similarity. Berikut dibawah ini merupakan persamaan dari Cosine Similarity.

$$Similarity = \cos(\theta) = \frac{A \cdot B}{||A|| ||B||} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

$$Distance = Dc(A, B) = 1 - Sc(A, B)$$

2.5 Raspberry PI

Raspberry PI atau disingkat dengan nama Raspi, merupakan *micro computer* multifungsi yang berukuran kecil. Raspberry PI dikembangkan oleh Raspberry Pi Foundation dan berawal bertujuan untuk modul pembelajaran ilmu komputer, Nama Raspberry terinspirasi dari nama buah-buahan dan PI kepanjangan dari Python. Raspberry PI juga merupakan papan tunggal (single board circuit; SBC).

2.6 Webcam

Webcam atau web camera adalah sebuah kamera video digital kecil yang dihubungkan dengan komputer melalui port USB atau port COM. Pada umumnya, webcam dapat digunakan untuk menangkap citra atau merekam video secara real-time. Biasanya di dalam webcam tidak hanya terdapat kamera saja. Webcam mempunyai fitur seperti mikrofon yang sudah built-in, sensor gerakan, serta lampu indikator yang akan menyala saat kamera sedang aktif.

2.7 TensorFlow

Tensorflow merupakan framework atau library open-source secara gratis dalam pengembangan Machine Learning (ML) atau Kecerdasan Buatan khususnya banyak digunakan pada pelatihan atau inferensi jaringan saraf (Deep Neural Networks). Tensorflow juga merupakan platform end-to-end yang mudah digunakan oleh pemula atau ahli didalam pembuatan ML dan deploy ML.

2.7.1 TensorFlow Lite

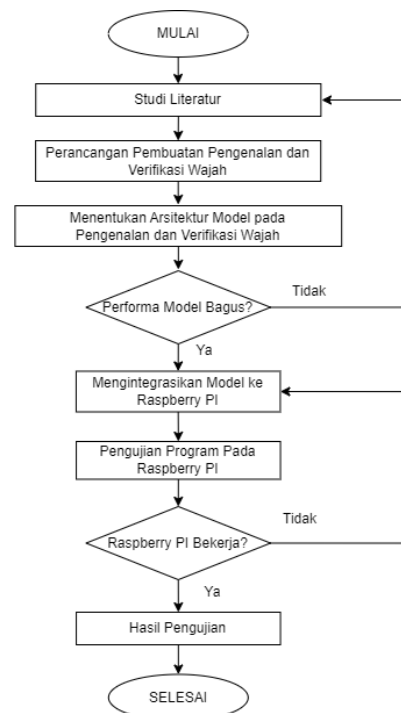
TensorFlow Lite atau yang disebut TF Lite merupakan library atau framework tensorflow versi ukuran kecil. TF Lite biasanya mengkonversi pre-trained model menjadi format khusus yang dapat mengoptimalkan kecepatan dan ukuran pada model tersebut. TF Lite yang berbentuk format khusus biasanya digunakan untuk deploy pada perangkat cross-platform seperti mobile (Android atau iOS) dan edge devices yang berbasis mikrocontrollers atau mini-computer yang berbasis linux. Pada dasarnya, perangkat kecil memiliki memori atau daya komputasi yang terbatas. Berbagai optimasi dapat diterapkan pada model sehingga dapat dijalankan dalam batasan tersebut. Selain itu, beberapa pengoptimalan memungkinkan penggunaan perangkat keras khusus untuk inferensi yang dipercepat.

2.8 DeepFace

DeepFace adalah pustaka pengenalan wajah dan analisis atribut wajah paling ringan untuk Python. Pustaka DeepFace open-source mencakup semua model AI terdepan untuk pengenalan wajah dan secara otomatis menangani semua

prosedur untuk pengenalan wajah di latar belakang. DeepFace mempunyai fitur untuk melakukan face recognition, face verify dan facial attribute analysis seperti usia, jenis kelamin, emosi dan ras. Kemudian, terdapat beberapa model untuk digunakan di dalam prosesnya seperti VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib and Sface.

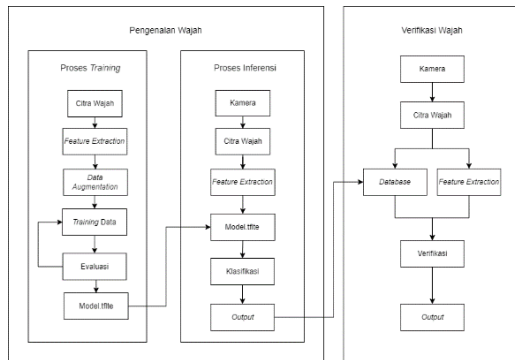
3. METODOLOGI PENELITIAN



Gambar 5. Flowchart Penelitian

4. PERANCANGAN DAN PENGUJIAN

Pada perancangan penelitian ini terdapat dua perancangan yang terdiri dari perancangan pengenalan wajah dan perancangan verifikasi wajah yang berisikan blok diagram serta diagram alir program. Setelah itu, akan dilakukan sebuah pengujian dengan beberapa skenario. Berikut ini merupakan blok diagram dari proses pengenalan dan verifikasi wajah pada **Gambar 6**.



Gambar 6. Blok Diagram Proses Keseluruhan

4.1 Perancangan Pengenalan Wajah

Pada perancangan ini merupakan tahapan pertama untuk proses pembuatan dan pelatihan model untuk pengenalan wajah. Tahapan pembuatan model dilakukan pada Google Colaboratory dengan menggunakan library TensorFlow. Tahapan ini diawali dengan akuisisi data, kemudian Data Augmentation menggunakan Image Data Generator, training data, evaluasi hingga konversi model dan menyimpan model dalam bentuk format TensorFlow Lite (.tflite) yang akan di integrasikan kedalam Raspberry PI. Selanjutnya, model yang telah di format akan dimasukan ke dalam program inferensi untuk menjalankan sistem pengenalan wajah. Berikut ini merupakan langkah – langkah yang harus dilakukan dalam pembuatan sistem pengenalan wajah dari awal hingga akhir :

4.1.1 Akuisisi Data

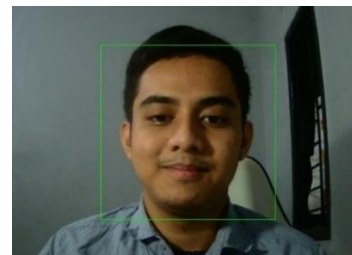
Akuisisi data atau citra dilakukan secara *realtime* dengan menggunakan kamera webcam dan rekaman video yang kemudian akan dipecah dari per *frame* menjadi gambar menggunakan bahasa python yang telah dibuat dan seluruh citra akan disimpan ke dalam bentuk format (.jpg). Data citra yang diambil terdiri dari 5 dataset wajah orang yaitu Farhan, Eca, Nizam, Rizkyellyasa dan Zakaria. Pada **Tabel 1** dapat dilihat jumlah citra yang di dapatkan dari keseluruhan dataset citra yang akan digunakan dan dalam proses pelatihan akan dibagi menjadi 3 data yaitu, data training, data validation dan data testing.

Tabel 1. Ukuran dan Jumlah Dataset

Data	Ukuran	Jumlah
Training	70%	750
Validation	20%	250
Testing	10%	100
Total	100%	1100

4.1.2 Feature Extraction

Setelah melakukan akuisisi data citra, citra tersebut akan dilakukan feature extraction. Feature extraction merupakan proses untuk mendapatkan seluruh fitur wajah, wajah pada citra akan di deteksi terlebih dahulu menggunakan Haar Cascade Classifier yang sudah di sediakan oleh library OpenCV dengan menambahkan file script XML yang sudah terdapat algoritma pola wajah manusia.



Gambar 7. ROI Wajah dan Bounding Box

4.1.3 Data Augmentation

Sebelum data citra dimasukan kedalam pelatihan model *neural network* untuk dilatih, citra akan terlebih dahulu masuk ke tahap data augmentation. Hal ini bertujuan agar data citra memiliki variasi data untuk menghasilkan data pelatihan tambahan dan data yang telah disesuaikan dengan kebutuhan.

Tabel 2. Data Augmentation

1	rotation_range	Perputaran citra wajah sampai 10 derajat	
2	shear_range	Memiringkan citra wajah sampai 10 derajat	
3	horizontal_flip	Pembalikan citra wajah secara horizontal	

4.1.4 Training Data

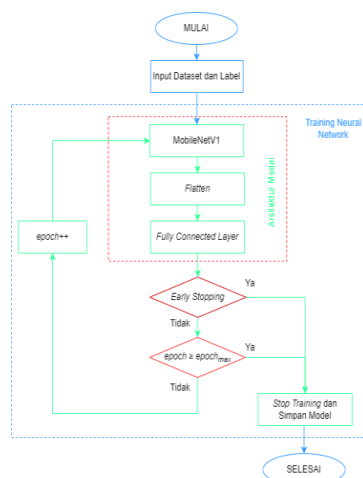
Pada tahap pelatihan data, arsitektur model yang akan digunakan yaitu MobileNetV1 dengan menggunakan metode transfer learning. Arsitektur MobileNetV1 memiliki kedalaman 55 lapisan dan memiliki 3,4 juta parameter. Model ini akan menerima inputan citra RGB dengan ukuran 224 x 224 piksel. Model ini merupakan pre-trained model yang sudah dilatih menggunakan dataset ImageNet yang memiliki 1000 kelas atau label maka untuk lapisan teratas tidak digunakan dan yang akan digunakan hanya lapisan ekstraksi fitur yaitu sebelum lapisan *flatten*.

Tabel 3. Ringkasan Arsitektur MobileNetV1

Layer	Shape	Params
Input Layer	(224, 224, 3)	0
MobileNetV1 (base model)	(7, 7, 1024)	3,228,864
Flatten	(50176)	0
Dense	1024	51,381,248
Dense	5	5125
Total Parameter		51,388,421

Tabel 4. Pengaturan Untuk Proses Training

Pengaturan	Keterangan
Epoch	150
Batch Size	32
Early Stopping	10 Epoch



Gambar 8. Diagram Alir Training Neural Network

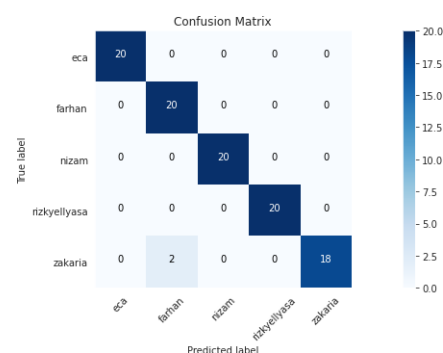
4.1.5 Evaluasi

Evaluasi diperlukan untuk mengetahui hasil dari proses *training* pada *neural network* dengan menggunakan arsitektur MobileNetV1. Proses *training* terhenti pada epoch ke-45 karena *Early Stopping* dijalankan setelah memenuhi syarat bahwa tidak ada perubahan atau sudah terjadi konvergensi dan menghindari peningkatan pada *validation loss*. Dalam hal ini, epoch ke-36 merupakan epoch yang terbaik dari keseluruhan *epoch* karena memiliki nilai *validation loss* terendah.



Gambar 9. Grafik Accuracy dan Loss Pada MobileNetV1

Pada **Gambar 9** menunjukkan grafik dari model MobileNetV1 dalam melakukan proses training, dari grafik tersebut menunjukkan juga model tersebut tidak mengalami overfitting dan underfitting yang berlebih. Dari proses training tersebut memperoleh akurasi 96%. Kemudian, saat mengevaluasi model tersebut dengan data testing diperoleh akurasi sebesar 98% dan *confusion matrix* nya dapat dilihat pada **Gambar 10**.



Gambar 10. Confussion Matrix Pada Data Testing

Tabel 5. Classification Report

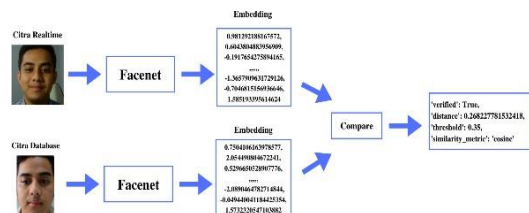
	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Eca	1.00	1.00	1.00
Farhan	0.91	1.00	0.91
Nizam	1.00	1.00	1.00
Rizkyellyasa	1.00	1.00	1.00
Zakaria	1.00	0.90	0.95
<i>Accuracy avg</i>			0.98

4.2 Perancangan Verifikasi Wajah

Perancangan ini merupakan tahap kedua setelah model pengenalan wajah dibuat dan verifikasi wajah akan menggunakan *library* DeepFace. Pada awalnya, tahap ini akan membutuhkan sebuah database yang berisikan citra dengan masing – masing satu citra wajah dari setiap orang atau label dengan jumlah 5 citra dalam database. Citra yang akan digunakan di dalam database sudah dilakukan feature extraction untuk mendapatkan fitur wajah.

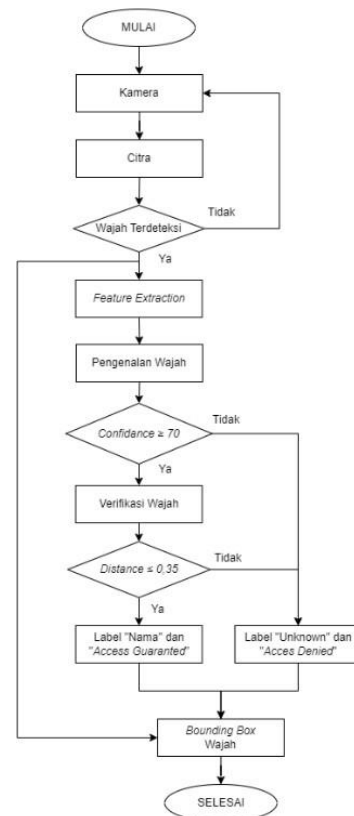
**Gambar 11. Citra Pada Database Untuk Verifikasi Wajah**

Model yang akan digunakan dalam proses verifikasi yaitu “Facenet”, model ini akan menghasilkan *embedding* dari citra, kemudian untuk mengetahui jarak perbedaan antar *embedding* menggunakan cosine similarity dengan *threshold* 0,35. Dibawah ini merupakan blok diagram dari proses verifikasi wajah menggunakan *library* DeepFace.

**Gambar 12. Blok Diagram Verifikasi Wajah**

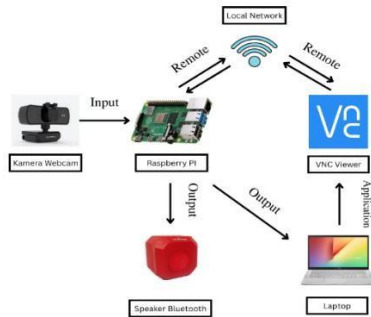
4.3 Diagram Alir Program

Dibawah ini merupakan diagram alir (*flowchart*) program pengenalan dan verifikasi wajah secara keseluruhan yang akan dijalankan pada Raspberry PI dengan menggunakan bahasa Python.

**Gambar 13. Diagram Alir Program Dalam Raspberry PI**

4.4 Pengujian

Pada tahap ini akan dilakukan pengujian masing - masing model pengenalan dan verifikasi wajah dan secara keseluruhan dengan berdasarkan jarak. Pengujian dilakukan menggunakan citra wajah secara realtime dengan kamera webcam sebagai input untuk di proses dalam Raspberry PI yang akan di remote melalui laptop menggunakan aplikasi VNC Viewer. Pengujian ini dilakukan untuk mengetahui akurasi dan waktu proses dari sistem yang telah dibuat untuk model pengenalan dan verifikasi wajah secara realtime berdasarkan masing – masing wajah yang di kenal dengan jarak 40 cm, 70cm, dan 100 cm.



Gambar 14. Blok Diagram Alat Pengujian

4.4.1 Pengujian Model Pengenalan Wajah

Tabel 6. Hasil Pengujian Akurasi Pada Model Pengenalan Wajah Berdasarkan Jarak 40cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	Total Data Benar	30	30	27	27	28	142
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{142}{150} \times 100\% = 95\%$						

Tabel 7. Hasil Pengujian Akurasi Pada Model Pengenalan Wajah Berdasarkan Jarak 70cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
70	Total Data Benar	29	27	25	26	26	133
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{133}{150} \times 100\% = 89\%$						

Tabel 8. Hasil Pengujian Akurasi Pada Model Pengenalan Wajah Berdasarkan Jarak 100cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
100	Total Data Benar	25	21	24	25	22	117
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{117}{150} \times 100\% = 78\%$						

Dari hasil pengujian model pengenalan

wajah dari tabel diatas, rata – rata keseluruhan akurasi berdasarkan jarak 40 cm, 70 cm, 100cm didapatkan rata-rata akurasi:

$$\text{Akurasi} = \left(\frac{95 + 89 + 78}{3} \right) \% = 87\%$$

Tabel 9. Hasil Pengujian Waktu Pada Model Pengenalan Wajah

Jarak (cm)	Label Wajah					Rata - Rata
	Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	0,13 detik	0,12 detik	0,12 detik	0,11 detik	0,12 detik	0,12 detik
70	0,13 detik	0,11 detik	0,11 detik	0,12 detik	0,13 detik	0,12 detik
100	0,13 detik	0,13 detik	0,12 detik	0,12 detik	0,12 detik	0,12 detik
Rata - Rata						0,12 detik

4.4.2 Pengujian Model Verifikasi Wajah

Tabel 10. Hasil Pengujian Akurasi Pada Model Verifikasi Wajah Berdasarkan Jarak 40cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	Total Data Benar	30	28	27	28	28	141
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{141}{150} \times 100\% = 94\%$						

Tabel 11. Hasil Pengujian Akurasi Pada Model Verifikasi Wajah Berdasarkan Jarak 70cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
70	Total Data Benar	27	27	25	24	26	129
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{129}{150} \times 100\% = 86\%$						

Tabel 12. Hasil Pengujian Akurasi Pada Model Verifikasi Wajah Berdasarkan Jarak 100cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
100	Total Data Benar	24	24	23	24	25	120
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{120}{150} \times 100\% = 80\%$						

Dari hasil pengujian model verifikasi wajah dari tabel diatas, rata – rata keseluruhan akurasi berdasarkan jarak 40 cm, 70 cm, 100cm didapatkan rata-rata akurasi:

$$\frac{94 + 86 + 80}{3}$$

$$\text{Akurasi} = \left(\frac{94 + 86 + 80}{3} \right) \% = 87\%$$

Tabel 13. Hasil Pengujian Waktu Pada Model Verifikasi Wajah

Jarak (cm)	Label Wajah					Rata - Rata
	Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	1,12 detik	1,17 detik	1,13 detik	1,14 detik	1,12 detik	1,14 detik
70	1,14 detik	1,17 detik	1,13 detik	1,15 detik	1,12 detik	1,14 detik
100	1,13 detik	1,13 detik	1,14 detik	1,16 detik	1,10 detik	1,13 detik
Rata - Rata						1,14 detik

4.4.3 Pengujian Model Pengenalan dan Verifikasi Wajah Secara Keseluruhan

Tabel 14. Hasil Pengujian Akurasi Pada Model Secara Keseluruhan Berdasarkan Jarak 40cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	Total Data Benar	30	28	26	27	27	138
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{138}{150} \times 100\% = 92\%$						

Tabel 15. Hasil Pengujian Akurasi Pada Model Secara Keseluruhan Berdasarkan Jarak 70cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
70	Total Data Benar	26	24	24	25	22	121
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{121}{150} \times 100\% = 81\%$						

Tabel 16. Hasil Pengujian Akurasi Pada Model Secara Keseluruhan Berdasarkan Jarak 100cm

Jarak (cm)	Kategori	Label Wajah					Total
		Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
100	Total Data Benar	25	24	21	24	22	116
	Total Seluruh Data	30	30	30	30	30	150
	Akurasi = $\frac{\sum \text{Data Prediksi Benar}}{\sum \text{Total Data}} \times 100\% = \frac{116}{150} \times 100\% = 77\%$						

Dari hasil pengujian model secara keseluruhan dari tabel diatas, rata – rata keseluruhan akurasi berdasarkan jarak 40 cm, 70 cm, 100cm didapatkan rata-rata akurasi:

$$92 + 81 + 77$$

$$\text{Akurasi} = \left(\frac{\quad}{3} \right) \% = 83\%$$

Tabel 17. Hasil Pengujian Waktu Pada Model Secara Keseluruhan

Jarak (cm)	Label Wajah					Rata - Rata
	Farhan	Eca	Nizam	Rizkyellyasa	Zakaria	
40	1,25 detik	1,29 detik	1,25 detik	1,26 detik	1,22 detik	1,25 detik
70	1,28 detik	1,29 detik	1,24 detik	1,27 detik	1,24 detik	1,26 detik
100	1,26 detik	1,25 detik	1,27 detik	1,28 detik	1,22 detik	1,26 detik
Rata - Rata						1,26 detik

5. KESIMPULAN

Berdasarkan hasil perancangan dan pengujian dari pengenalan dan verifikasi wajah menggunakan *transfer learning* berbasis raspberry pi dapat disimpulkan sebagai berikut:

1. Setelah dilakukan pengujian menggunakan kamera *webcam* berbasis Raspberry PI terhadap pengenalan wajah menggunakan model “MobileNetV1” pada citra *realtime* sebanyak 30 percobaan didapat bahwa rata – rata akurasi yang didapatkan yaitu 87%, sedangkan untuk pengujian verifikasi wajah menggunakan model “Facenet” didapatkan rata – rata akurasi yaitu 87% dan rata – rata akurasi yang didapatkan dari model pengenalan dan verifikasi wajah secara keseluruhan yaitu 83%.
2. Pengujian yang dilakukan berdasarkan jarak 40 cm, 70 cm dan 100 cm diperoleh akurasi dengan jarak terdekat memiliki akurasi tinggi sedangkan jarak terjauh memiliki akurasi rendah.
3. Pengujian terhadap waktu atau *response time* dari model pengenalan wajah menggunakan “MobileNetV1” membutuhkan rata – rata waktu 0,12 detik sedangkan dari model verifikasi wajah menggunakan “Facenet” membutuhkan rata – rata waktu 1,14 detik dan rata – rata waktu yang diperoleh dari model pengenalan dan verifikasi wajah secara keseluruhan sebesar 1,26 detik sehingga masing –

masing pada model yang dipilih mempunyai performa berbeda sehingga dapat mempengaruhi waktu pemrosesan.

DAFTAR PUSTAKA

- [1] Afrizal, Setia Juli Irzal Ismail, dan Gandeva Bayu Satrya, “Perancangan Sistem Keamanan Rumah Menggunakan Deteksi Wajah Berbasis Machine Learning Menggunakan Tensorflow”, Universitas Telkom, Bandung, 2022.
- [2] Adhi Setiawan, “Augmentasi Data Pada Computer Vision”, 2021, Diakses pada tanggal 21 September 2022 di situs <https://medium.com/data-folks-indonesia/augmentasi-data-pada-computer-vision-45c5ebe10e8f>
- [3] D. Suprianto, “Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, EigenFace PCA & MySQL”, 2013.
- [4] Julpan, Erna Budhiarti Nababan, dan Muhammad Zarlis, “Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar Dalam Algoritma Backpropagation Pada Prediksi Kemampuan Siswa”, 2015.
- [5] Jeff Heaton, “Introduction to the Math of Neural Network”. 2005.
- [6] Munir, R. “Pengolahan Citra Digital”, Informatika, Bandung, 2004
- [7] Pulung Adi Nugroho, Indah Fenriana, dan Rudy Arijanto, M.Kom, “Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia”, 2020.
- [8] Pulung Nurtantio Andono, T. Sutojo, dan Muljono, “Pengolahan Citra Digital”, Penerbit ANDI, Yogyakarta, 2017.
- [9] Putri Nada Zakiya, Ledy Novamizanti, dan Syamsul Rizal, “Klasifikasi Patologi Makula Retina Melalui Citra OCT Menggunakan Convolutional Neural Network Dengan Arsitektur Mobilenet”, Universitas Telkom, Bandung, 2021
- [10] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks”, 2018, Diakses pada tanggal 20 September 2022 di situs <https://towardsdatascience.com/a-comprehensive-guide-toconvolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [11] Suartika E. P, Arya Yudhi Wijaya Wijaya, dan Rully Soelaiman. “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101.” 2016.
- [12] Sarirotul Ilahiyah, dan Agung Nilogiri, “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network”, Universitas Muhammadiyah Jember, 2018.