

## DESAIN SISTEM PENGENALAN WAJAH MENGGUNAKAN RASPBERRY PI 3

### DESIGN OF FACE RECOGNITION SYSTEM USING RASPBERRY PI 3

Muh Ihsan S<sup>1</sup>, Nyoman Bogi Aditya Karna<sup>2</sup>, Raditiana Patmasari<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

<sup>1</sup>[muhihans33@gmail.com](mailto:muhihans33@gmail.com), <sup>2</sup>[nyoman.bogi@gmail.com](mailto:nyoman.bogi@gmail.com), <sup>3</sup>[raditiana@telkomuniversity.ac.id](mailto:raditiana@telkomuniversity.ac.id)

---

#### Abstrak

Tugas akhir ini menganalisis desain sistem pengenalan wajah yang menggunakan *Raspberry Pi 3 B+* sebagai pusat sistem. Sistem pada Tugas Akhir ini menggunakan *Raspberry Pi 3 B+* dan *Raspberry Pi Camera Module V2.1*. Adapun sistem *face recognition* yang digunakan merupakan sistem *face recognition* dari *pyimagesearch* yang diperuntukkan untuk *Raspberry Pi 3*. Terdapat program *encode\_face.py* untuk melakukan proses training *image* dari lima orang subjek penelitian. Terdapat pula program *pi\_face\_recognition.py* yang akan dijalankan dan diuji terhadap empat orang yang wajahnya sudah ada pada *database* yang telah di *training* sebelumnya dan terhadap seseorang yang wajahnya tidak ada pada *database* sistem *face recognition*. Metode yang digunakan untuk *face recognition* yaitu *Deep Metric Learning* dengan *triplet training step*. Sistem *face recognition* pada tugas akhir ini berdasarkan pada *pi\_face\_recognition* yang berasal dari *pyimagesearch* oleh Adrian. *Face recognition* tersebut menggunakan *network architecture* bernama *dlib* milik David King dan modul *face recognition* milik Adam Geutgey. Sedangkan *face detection* memanfaatkan *haar cascade frontal face default* berupa file *xml*. *Dataset* terdiri dari 5 orang dengan jumlah foto wajah perorang yaitu 30, sehingga totalnya yaitu 150 foto. Kemudian *dataset* tersebut di *training* menggunakan *encode\_face.py* sehingga menghasilkan berkas *TUGASAKHIR-5subjek.pickle*. Pengujian sistem *face recognition* dilakukan pada empat kondisi jarak pengujian yang berbeda yaitu 1,5 meter, 2 meter, 2,5 meter, dan 3 meter. Terdapat tiga macam parameter pengujian yaitu parameter *size*, parameter *scale factor*, dan parameter *neighbourhood*. Variasi nilai dari parameter *size* yaitu 20×20, 25×25, 30×30, dan 35×35. Variasi nilai dari parameter *scale factor* yaitu 1.1, 1.2, 1.3, dan 1.4. Variasi nilai dari parameter *neighbourhood* yaitu 3, 4, 5, dan 6. Hasil pengujian menunjukkan nilai *Accuracy* tertinggi yaitu 80% dan *True Positive Rate* mencapai 100% dengan parameter terbaik yaitu parameter *size* 20×20, parameter *scale factor* 1,1, dan parameter *neighbourhood* bernilai 3.

**Kata Kunci:** *Raspberry Pi 3 B+*, *face recognition*, *Deep Metric Learning*

---

#### Abstract

This final project analyzes the design of a face recognition system that uses *Raspberry Pi 3 B +* as the center of the system. The system in this final project uses *Raspberry Pi 3 B +* and *Raspberry Pi Camera Module V2.1*. The face recognition system used is a face recognition system from *Pyimagesearch* which is intended for *Raspberry Pi 3*. There is an *encode\_face.py* program to carry out the image training process of five research subjects. There is also the *pi\_face\_recognition.py* program which will be run and tested on four people whose faces have been in a database that has been previously trained and on someone whose face is not in the face recognition system database. The method used for face recognition is *Deep Metric Learning* with step training triplets. The face recognition system in this thesis is based on *pi\_face\_recognition* originating from *pyimagesearch* by Adrian. *Face recognition* uses a network architecture called David King's *Dlib* and Adam Geutgey's face recognition module. Whereas face detection utilizes a default frontal face haar cascade in the form of an *xml* file. The dataset consists of 5 people with 30 photos per face, so that the total is 150 photos. Then the dataset is trained using *encode\_face.py* to produce the *TUGASAKHIR-5subjek.pickle* file. Face recognition system testing is carried out on four different test conditions, namely 1.5 meters, 2 meters, 2.5 meters and 3 meters. There are three types of testing parameters, namely size parameters, scale factor parameters, and neighborhood parameters. Variations in the value of the size parameter are 20x20, 25x25, 30x30, and 35x35. Variations in the values of the scale factor parameters are 1.1, 1.2, 1.3, and 1.4. Variation values of the neighboring parameters are 3, 4, 5, and 6. The test results show the highest *Accuracy* value is 80% and the *True Positive Rate* reaches 100% with the best parameters, namely the size parameter 20 × 20, scale factor parameters 1,1, and parameters neighborhood is worth 3.

**Keywords:** *Raspberry Pi 3 B+*, *face recognition*, *Deep Metric Learning*

---

## 1. Pendahuluan

Kebutuhan papan yaitu kebutuhan untuk memiliki sebuah rumah merupakan salah satu kebutuhan primer manusia. Rumah harus memberikan rasa aman dan nyaman bagi para penghuninya. Akan tetapi, ada sebuah permasalahan yang mengancam keamanan dan keselamatan para penghuninya. Permasalahan tersebut adalah pencurian yang terjadi di rumah kediaman masyarakat. Berdasarkan laporan publikasi Statistik Kriminal 2017 dari Badan Pusat Statistik (BPS) yang bersumber dari data Markas Besar Polisi Republik Indonesia (MABES POLRI) [1], pada tahun 2017, tercatat 26.636 kasus kejahatan pencurian dan 36.467 kasus kejahatan pencurian dengan pemberatan seperti pencurian di rumah seseorang. Informasi tersebut menjadi dasar utama perlunya dilakukan penelitian tentang desain sebuah sistem pengenalan wajah yang dapat diterapkan pada sistem keamanan rumah. Sebelumnya sudah pernah dilakukan beberapa penelitian sejenis. Penelitian tersebut dilakukan oleh Irfan Kurniawan [2] tentang sistem keamanan rumah berbasis *Raspberry Pi* dengan memanfaatkan aplikasi *Telegram Messenger* namun terbatas pada *monitoring* saja. Selain itu, sistem *face recognition* belum digunakan pada penelitian tersebut. Pada penelitian ini, penulis menambahkan sistem *face recognition* akan digunakan untuk membedakan antara pemilik rumah dan orang asing. Sistem *face recognition* ini dapat digunakan sebagai elemen tambahan pada sistem keamanan rumah yang sudah ada. Diharapkan dari penelitian ini, dihasilkan sistem *face recognition* yang akurat dalam membedakan antara orang yang wajahnya sudah ada pada *database* maupun yang wajah orang yang tidak ada pada *database* (orang asing).

## 2. Tinjauan Pustaka

### A. *Raspberry Pi 3 B+*

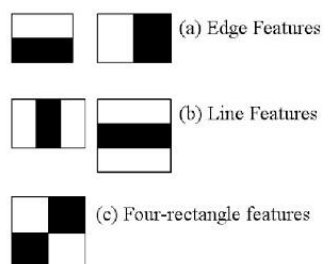
*Raspberry Pi* merupakan sebuah *board* elektronik dengan dimensi panjang dan lebar seperti kartu kredit. Meskipun ukurannya kecil namun perangkat ini sudah dapat digunakan sebagaimana komputer pada umumnya. *Raspberry Pi* dihubungkan terlebih dahulu dengan catu daya 5 Volt 2 Ampere agar dapat digunakan. Pemanfaatan *board* ini sangat luas, mulai dari menulis dokumen, *programming*, memainkan lagu atau video, bermain *game* kategori ringan, menjelajahi *internet*, menjadikannya sebagai *web server*, bahkan dapat dimanfaatkan dalam teknologi *Internet of Things* (IoT) berkat dukungan terhadap penggunaan sensor-sensor melalui pin *General-Purpose Input/Output* (GPIO). *Raspberry Pi* sangat populer berkat harganya yang murah, ukurannya yang kecil, dan penggunaan daya listrik yang juga kecil. Banyak sekali *project* atau hal yang dapat diwujudkan menggunakan perangkat ini seperti pengolahan citra pada robot, pengendalian alat elektronik rumah tangga dari jarak jauh, mengendalikan *quadcopter*, dan masih banyak lagi. Semua tergantung dari kreativitas dan ide yang dimiliki oleh orang-orang yang menggunakan *Raspberry Pi* [3].

### B. Modul *Raspberry Pi camera V2*

*Raspberry Pi camera module (PiCam)* merupakan modul tambahan bagi *Raspberry Pi* untuk mengambil gambar ataupun video dengan resolusi yang tinggi. Tugas Akhir ini memanfaatkan *PiCam* sebagai komponen penting dari sistem *face recognition*. *PiCam* terbaru yaitu *Raspberry Pi Camera Module v2* yang selanjutnya kita sebut dengan *PiCam V2* dirilis pada bulan April tahun 2016. Modul ini menggunakan *Sony IMX219 8-megapixel sensor*, lebih baik dari *PiCam* versi sebelumnya yang menggunakan *5-megapixel OmniVision OV5647 sensor*. *PiCam V2* mendukung pengambilan video mode 1080p30, 720p60 dan VGA90. Modul ini dihubungkan dengan port CSI pada *Raspberry Pi* melalui kabel *ribbon* sepanjang 15 cm.

### C. Haar Feature-based Cascade Classifiers [6]

Deteksi objek menggunakan *Haar Feature-based Cascade Classifiers* merupakan metode deteksi objek yang efektif. Metode ini diusulkan oleh Paul Viola dan Michael Jones. Ini adalah pendekatan berbasis *machine learning* di mana fungsi kaskade dilatih dari banyak gambar positif dan negatif. Kemudian digunakan untuk mendeteksi objek dalam gambar lain. Pada deteksi wajah, algoritma ini membutuhkan banyak gambar positif (gambar wajah) dan gambar negatif (gambar tanpa wajah) untuk melatih *classifier*. Selanjutnya, *feature* perlu di ekstrak darinya. Digunakan *haar features* seperti yang ditunjukkan pada gambar 2.3.



Gambar 1: Ilustrasi *Haar feature*.

Lalu terdapat konsep Cascade of Classifiers. Alih-alih menerapkan semua 6000 fitur pada sebuah jendela, fitur-fitur tersebut dikelompokkan ke dalam beberapa tahapan pengklasifikasi dan diterapkan satu per satu. (Biasanya beberapa tahap pertama akan mengandung fitur yang jauh lebih sedikit). Jika jendela gagal pada tahap pertama, buanglah. Kami tidak mempertimbangkan fitur yang tersisa di dalamnya. Jika lewat, terapkan fitur tahap kedua dan lanjutkan prosesnya. Jendela yang melewati semua tahapan adalah daerah wajah.

4

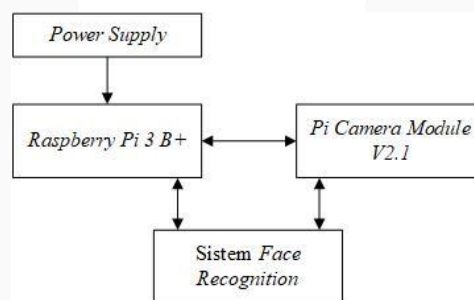
#### D. Deep Metric Learning dengan triplet training step

Pada Tugas Akhir ini dilakukan penginstalan *dlib* [8] dan modul *face\_recognition*. Davis King merupakan *author* dari *dlib* dan Adam Geitgey merupakan *author* dari modul *face\_recognition*. Ide untuk mengurangi *raw data* yang kompleks dari gambar menjadi *computer generated numbers* mulai menjadi *trend* di *machine learning*. Tugas Akhir ini tidak melakukan *training network* mulai dari awal. Akan tetapi, menggunakan *network* yang telah di *train* menggunakan *dataset* sebanyak-banyaknya 3 juta *images*. Pada *dlib facial recognition network* inilah dikeluarkan output *128-d vector* untuk kuantifikasi wajah. *Training* ini dilakukan dengan menggunakan *triplet training step*, dimana terdapat tiga buah *images* wajah. Dua *images* (*image #2* dan *image #3*) merupakan orang yang sama. Sedangkan *Image* yang lainnya merupakan wajah orang asing. *Neural Network* akan mengkuantifikasi wajah dan melakukan konstruksi *128-d vectors* atau *128 measurements* untuk masing-masing *image* wajah. Langkah selanjutnya adalah dilakukan *tweak* terhadap *weights* dari *Neural network* sehingga *128-d vectors* dari *image #2* dan *image #3* nilainya akan mendekati satu sama lain, tetapi nilainya menjauh dari *128-d vectors* dari *image #1*. Jadi, Tugas Akhir ini menggunakan metode *deep metric learning* dengan *triplet training step*. Biasanya *training* sebuah *network* bertujuan untuk menerima *input* dalam bentuk sebuah *image* lalu *output* nya berupa klasifikasi atau *label* dari *image* tersebut. Namun *deep metric learning* berbeda sama sekali. *Deep metric learning* mencoba mengeluarkan output berupa *real-value feature vector* sebanyak 128 bilangan *real* (mencakup bilangan positif, negative, dan pecahan). *Output* nya bukan berupa *label* atau bahkan koordinat/*bounding box* dari objek pada *image*.

### 3. Perancangan Sistem

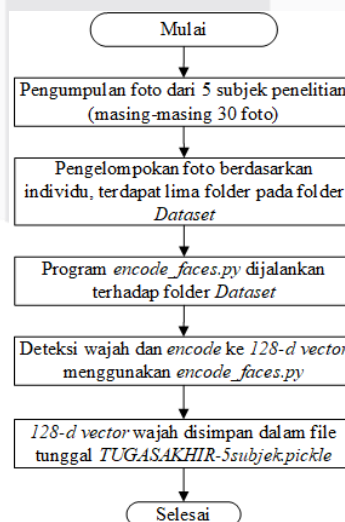
#### A. Gambaran Umum Sistem

Tugas Akhir ini mengajukan desain sistem yang terdiri dari sistem *face recognition* yang diimplementasikan pada perangkat *Raspberry Pi 3 B+* yang dilengkapi dengan *Pi Camera Module V2.1*. Adapun diagram blok sistem dari penelitian ini dapat dilihat dari gambar 2.



Gambar 2: Blok diagram sistem kerja.

#### B. Proses Training Dataset

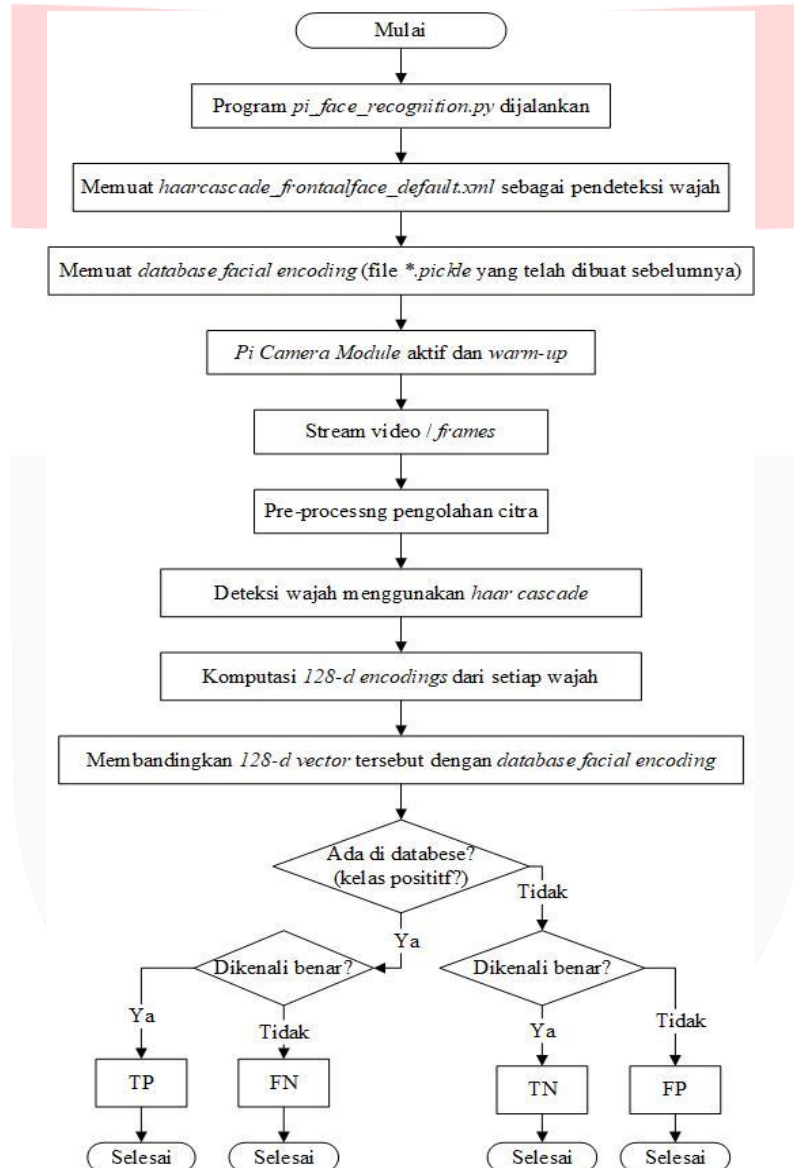


Gambar 3: Flowchart proses konstruksi 128-d vector wajah

Pada Tugas Akhir ini, sistem *face recognition* terbagi menjadi dua bagian. Bagian pertama sebagai proses persiapan merupakan proses konstruksi *128-d vector* wajah-wajah pada *Dataset*. Pada gambar 3, proses dimulai dari pengumpulan foto-foto atau *images* dari lima orang subjek penelitian. Setiap subjek penelitian memiliki 30 foto. Lalu dilakukan pengelompokan foto berdasarkan individu, sehingga terdapat lima buah folder. Setiap folder diberi nama sesuai dengan nama subjek penelitian. Lima folder tersebut ditempatkan kedalam sebuah folder yang sama yaitu folder *Dataset*. Selanjutnya program *encode\_faces.py* dieksekusi. Proses pertama dari eksekusi program *encode\_faces.py* tersebut adalah proses deteksi wajah pada semua foto dalam folder *Dataset*. Proses deteksi wajah dapat menggunakan metode *Histogram of Gradients* (HOG). Wajah-wajah yang telah dideteksi kemudian diolah sedemikian rupa dan di *encode* ke *128-d vector*. *128-d vectors* merupakan list dari 128 bilangan *real* yang digunakan untuk mengukur wajah. Semua *128-d vector* dari tiap wajah dalam folder *Dataset* akan disimpan dalam suatu file tunggal yaitu *TUGASAKHIR-5subjek.pickle*.

### C. Sistem Face Recognition (realtime)

Bagian kedua yaitu proses *face recognition* secara *realtime* dapat digambarkan ke dalam *flowchart* berikut



**Gambar 4:** Flowchart sistem *face recognition*

Pada gambar 4, proses diawali menjalankan program *pi\_face\_recognition.py* yang selanjutnya memuat file *haarcascade\_frontalface\_default.xml* pada direktori yang sama. File *\*.xml* tersebut nantinya akan digunakan dalam proses pendeteksian wajah pada *frame per frame* dari *videostream*. Selain itu, program *pi\_face\_recognition.py* juga memuat file *database facial encoding* yang berupa file *TUGASAKHIR-5subjek.pickle*. File ini dihasilkan dari proses pertama seperti yang ditunjukkan oleh gambar 3. Setelah kedua file tersebut dimuat, program *pi\_face\_recognition.py* akan mengakses *pi camera module* yang diawali dengan proses *warm-up* pada sensor *camera* selama 2 detik. Selanjutnya *pi camera modul* akan melakukan *stream video* secara

*realtime*. Subsistem *face recognition* akan melakukan pre-processing terhadap *frame* yang terdiri dari proses *re-size*, konversi dari *Blue Green Red* (BGR) ke *grayscale* untuk keperluan *face detection*, serta konversi dari BGR ke *Red Green Blue* (RGB) untuk keperluan *face recognition*. Hasil dari proses *face detection* adalah *rects*. *Rects* merupakan list dari lokasi wajah pada *frame* yang berbentuk kotak yang membatasi *area* wajah (*bounding box*). Selanjutnya, sistem akan memulai proses *face recognition* yaitu melakukan komputasi *128-d vector* terhadap wajah-wajah yang telah terdeteksi pada setiap *frame*. Hasil nya akan dibandingkan dengan *128-d vectors* dari file *TUGASAKHIR-5subjek.pickle* yang telah dimuat sebelumnya. Apabila yang membuka pintu merupakan anggota keluarga, maka akan terjadi *match* karena informasi wajah semua anggota keluarga pemilik rumah telah ada di *database facial encodings* (*TUGASAKHIR-5subjek.pickle*). Pada bagian akhir *flowchart*, akan ditentukan apakah hasil pengujian *face recognition* berupa *True Positive* (TP), *False Negative* (FN), *True Negative* (TN), atau *False Positive* (FP).

#### 4. Pengujian Sistem

Pada Tugas Akhir ini, pengujian sistem dilakukan dengan terhadap tiga jenis parameter yang sudah disebutkan di dalam program *pi\_face\_recognition.py*. Adapun ketiga parameter tersebut yaitu parameter *size*, parameter *scale factor*, dan parameter *neighbourhood*. Untuk parameter *size*, nilai *default* nya adalah , parameter *scale factor* nilai *default* nya adalah 1,1, dan parameter *neighbourhood* nilai *default* nya adalah 5. Namun pada proses pengujian, nilai parameter *size* bervariasi dari 20×20, 25×25, 30×30, dan 35×35. Nilai parameter *scale factor* bervariasi dari 1,1; 1,2; 1,3; dan 1,4. Nilai parameter *neighbourhood* bervariasi dari 3, 4, 5, dan 6. Saat pengujian dengan variasi nilai parameter *size*, maka parameter *scale factor* dan *neighbourhood* bernilai *default*. Saat pengujian dengan variasi nilai parameter *scale factor*, maka parameter *size* dan *neighbourhood* bernilai *default*. Saat pengujian dengan variasi nilai parameter *neighbourhood*, maka parameter *size* dan *scale factor* bernilai *default*. Ketiga parameter tersebut pada dasarnya merupakan parameter untuk *face detection* dan bukan merupakan parameter untuk *face recognition*. Namun, penulis memilih menggunakan ketiga parameter tersebut karena didalam program *pi\_face\_recognition.py* hanya ketiga parameter tersebut yang ada dan nilainya dapat diubah-ubah. Pada proses pengujian, akan dilakukan perubahan terhadap nilai-nilai dari ketiga parameter tersebut. Tugas Akhir ini akan berusaha memperoleh hasil dari dua metrik performansi *face recognition* yaitu *Accuracy* dan *True Positive Rate* (TPR). Pengujian dilakukan dengan cara menempatkan kamera pada ketinggian 2 meter dari atas permukaan lantai dan memberi empat penanda (*marker*) jarak di lantai terhadap kamera (1,5 meter, 2 meter, 2,5 meter, dan 3 meter). Pada setiap pengujian, subjek penelitian akan berdiri tepat pada masing-masing *marker* jarak tersebut. Subjek penelitian merupakan empat orang yang wajahnya telah dikenali oleh sistem *face recognition* pada tugas akhir ini. Selain itu , dilakukan pula pengujian terhadap orang yang wajahnya tidak ada pada *database*. Adapun n/a menandakan bahwa wajah gagal terdeteksi dan ttd menandakan tidak valid.

##### A. Data hasil Pelatihan

Tabel hasil pengujian terhadap variasi nilai parameter *size* yang terdiri dari empat tabel dapat diwakili oleh tabel 1 berikut.

Percobaan terhadap	Jarak antara wajah dengan kamera (meter)			
	1,5	2	2,5	3
Muh Ihsan S	TP	TP	TP	TP
Eko Nugroho	TP	TP	TP	n/a
Renandy Rifki	TP	TP	TP	TP
Utarya	TP	TP	TP	n/a
Orang asing	FP	FP	FP	FP
<i>Accuracy</i>	80%	80%	80%	66,67%
<i>TPR</i>	100%****	100%****	100%****	100%**

**Tabel 1:** Tabel hasil pengujian dengan parameter *size* bernilai 20×20.

Tabel hasil pengujian terhadap variasi nilai parameter *scale factor* yang terdiri dari empat tabel dapat diwakili oleh tabel 2 berikut.

Percobaan terhadap	Jarak antara wajah dengan kamera (meter)			
	1,5	2	2,5	3
Muh Ihsan S	TP	TP	TP	n/a
Eko Nugroho	TP	TP	n/a	n/a
Renandy Rifki	TP	TP	TP	n/a
Utarya	TP	TP	n/a	n/a
Orang asing	FP	FP	FP	n/a
<i>Accuracy</i>	80%	80%	66,67%	ttd
<i>TPR</i>	100%****	100%****	100%**	ttd

**Tabel 2:** Tabel hasil pengujian dengan parameter *scale factor* bernilai 1,1.

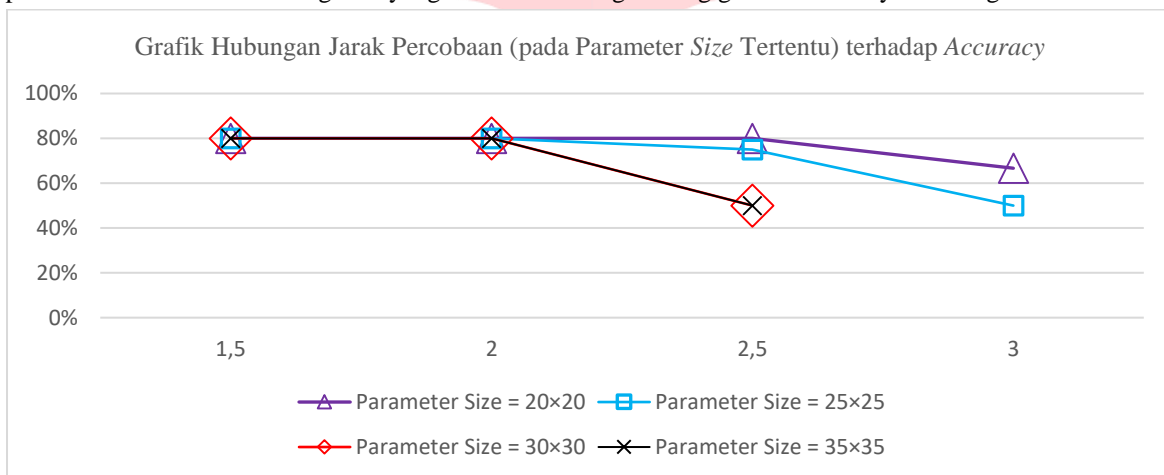
Tabel hasil pengujian terhadap variasi nilai parameter *neighbourhood* yang terdiri dari empat tabel dapat diwakili oleh tabel 3 berikut.

Percobaan terhadap	Jarak antara wajah dengan kamera (meter)			
	1,5	2	2,5	3
Muh Ihsan S	TP	TP	TP	TP
Eko Nugroho	TP	TP	TP	n/a
Renandy Rifki	TP	TP	TP	TP
Utarya	TP	TP	TP	n/a
Orang asing	FP	FP	FP	FP
Accuracy	80%	80%	80%	66,67%
TPR	100%****	100%****	100%****	100%**

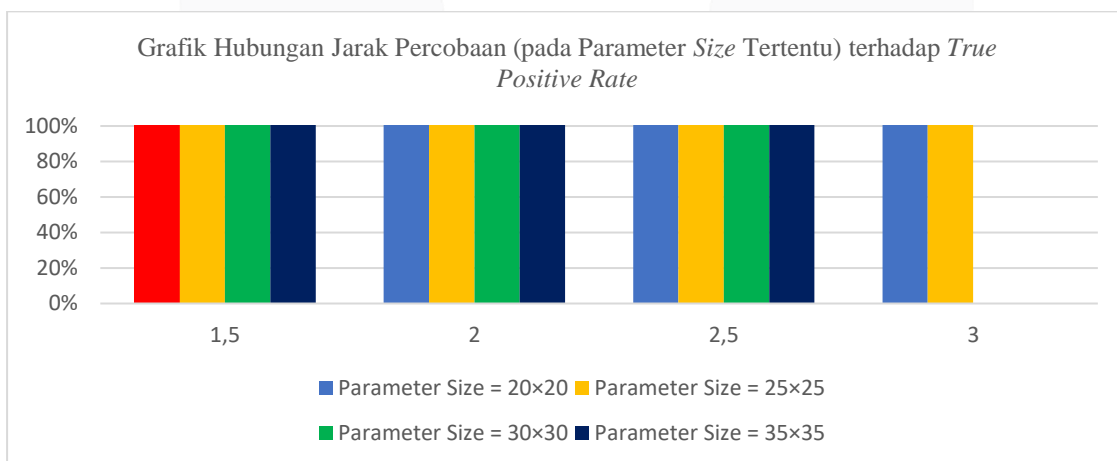
Tabel 3: Tabel hasil pengujian dengan parameter *neighbourhood* bernilai 3..

### B. Hasil Dan Analisis

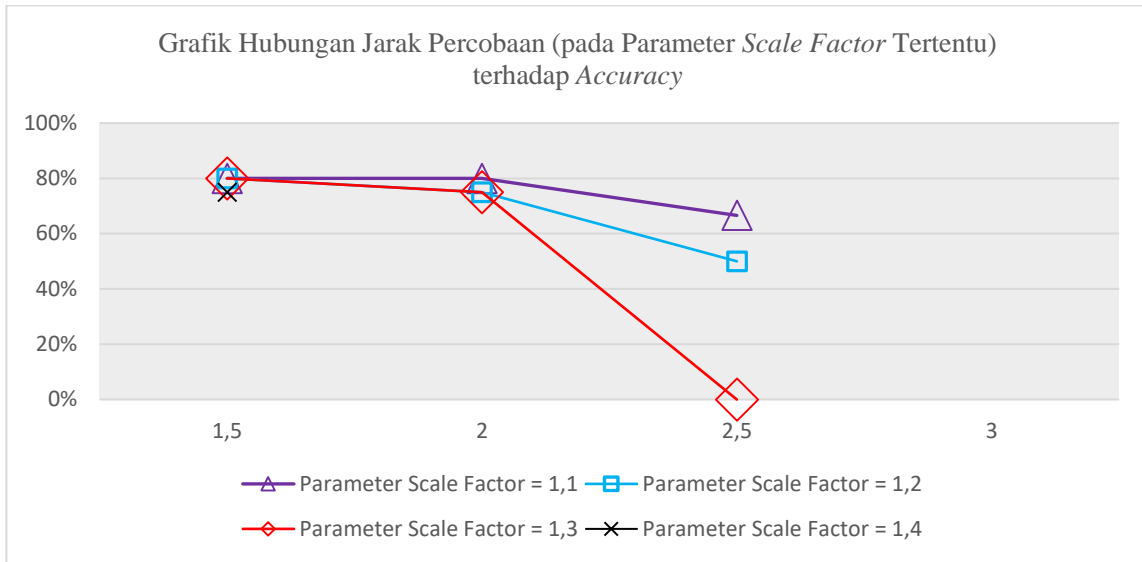
Setelah nilai *Accuracy* dan *True Positive Rate* dari masing-masing percobaan terhadap parameter *size*, parameter *scale factor*, dan parameter *neighbourhood* diperoleh, maka semua hasil pengujian dapat di representasikan kedalam enam grafik yang berbeda. Masing-masing grafik tersebut yaitu sebagai berikut.



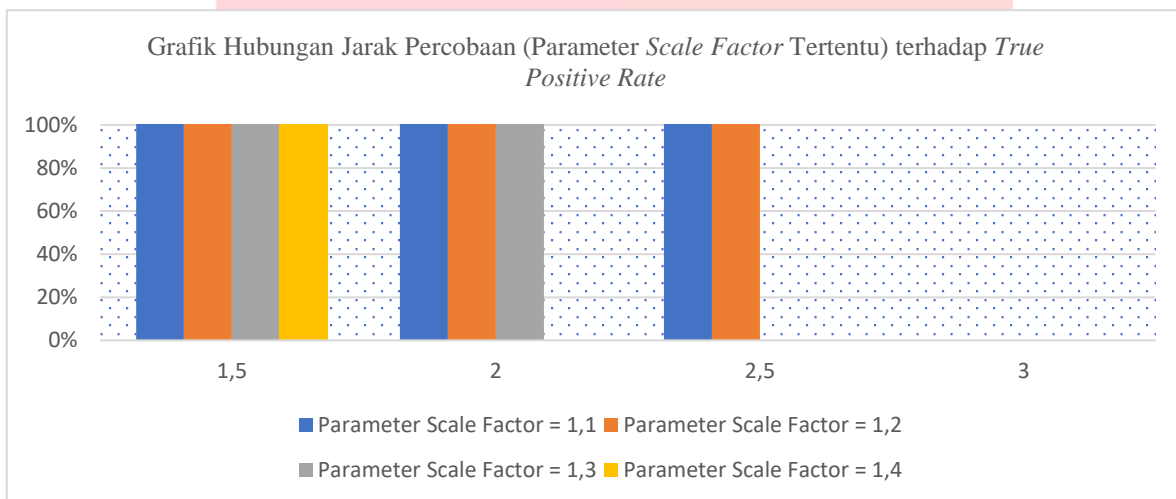
Gambar 5: Grafik hubungan jarak (*size* tertentu) terhadap *accuracy*.



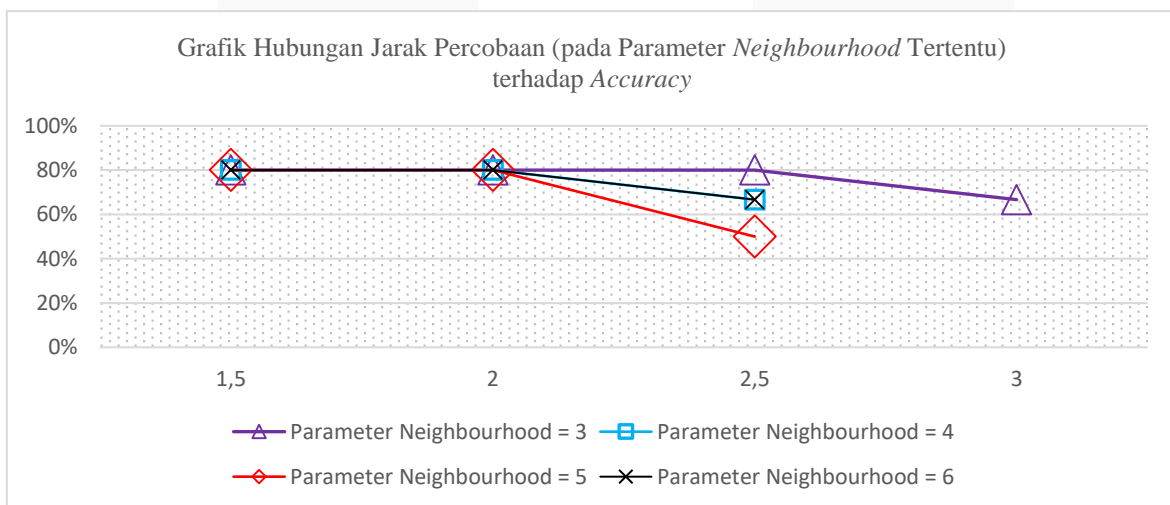
Gambar 6: Grafik hubungan jarak (*size* tertentu) terhadap *TPR*.



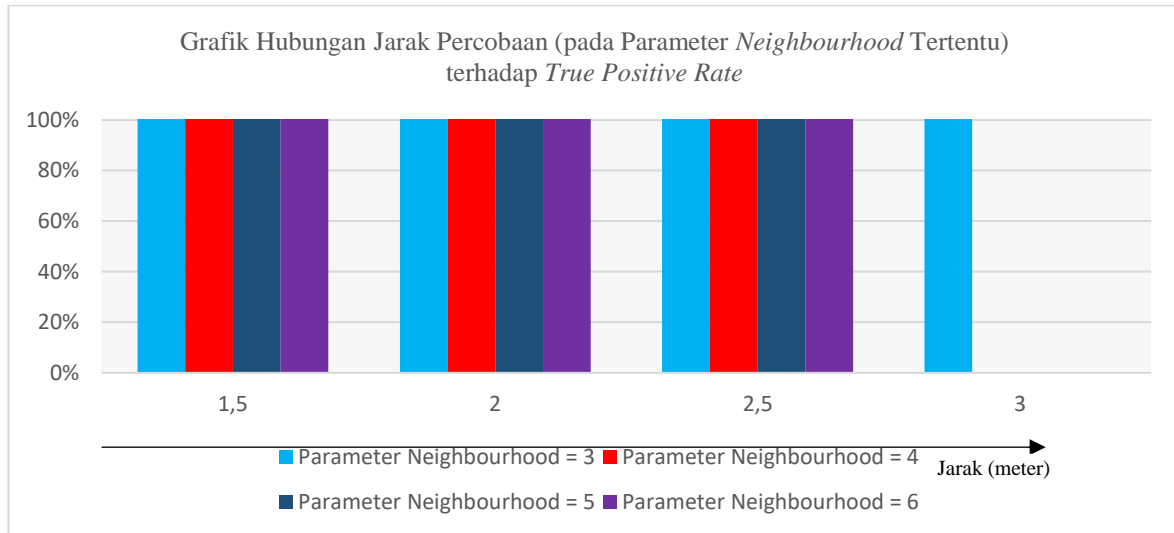
**Gambar 7:** Grafik hubungan jarak (*scalefactor* tertentu) dengan *accuracy*.



**Gambar 8:** Grafik hubungan jarak(*scalefactor* tertentu) dengan *TPR*.



**Gambar 9:** Grafik hubungan jarak(*neighbourhood* tertentu) dengan *accuracy*.



**Gambar 9:** Grafik hubungan jarak (*neighbourhood* tertentu) dengan *TPR*.

## 5. Kesimpulan

Dari hasil pengujian dan analisis yang penulis telah lakukan pada sistem *face recognition*, kesimpulan yang dapat diambil adalah sebagai berikut:

- Nilai accuracy akurasi terbaik diperoleh ketika parameter size bernilai  $20 \times 20$ , dimana nilainya stabil pada 80% dari jarak 1,5 meter hingga 2,5 meter yang berakhir pada nilai accuracy 66,67% pada jarak 3 meter. Sedangkan nilai accuracy terburuk diperoleh ketika parameter size bernilai  $30 \times 30$  dan  $35 \times 35$  yaitu sebesar 50% pada jarak 2,5 meter
- Nilai accuracy terbaik diperoleh pada saat parameter scale factor bernilai 1,1 dan parameter *neighbourhood* bernilai 3.
- Nilai True Positive Rate semuanya mencapai nilai 100%. Hal ini wajar karena faktor yang paling menentukan yaitu True Positive, yang dimana pada sistem *face recognition* ini, sangat akurat dalam membedakan wajah diantara para subjek penelitian yang wajahnya telah ada dalam database.
- Untuk True Positive Rate, parameter scale factor 1,4 menunjukkan hasil terburuk karena hanya pada jarak 1,5 meter. Hasil terbaik ditunjukkan oleh keadaan dengan parameter scale factor bernilai 1,1
- Berdasarkan hasil pengujian secara keseluruhan, sistem *face recognition* yang akurat tidak tercapai pada Tugas Akhir ini.

## 6. Daftar Referensi

- [1] Badan Pusat Statistik, "STATISTIK KRIMINAL 2017", Katalog 4401002, Jakarta : BPS, 2017.
- [2] Irfan Kurniawan, Muhamad & Sunarya, Unang & Tulloh, Rohmat. (2018). "Internet of Things : Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger". ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika. 6. 1. 10.26760/elkomika.v6i1.1.
- [3] A. Kadir, "Dasar Raspberry Pi: Panduan Praktis Mempelajari Pemrograman Perangkat Keras Menggunakan Raspberry Pi Model B". Yogyakarta: Pererbit ANDI, 2016.
- [4] The Raspberry Pi Foundation., "RASPBERRY PI 3 MODEL B", Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Diakses 16 Sept 2018, 10:15:59 WIB].
- [5] The Raspberry Pi Foundation., "CAMERA MODULE v2", Available: <https://www.raspberrypi.org/products/camera-module-v2/>, [Diakses 20 Sept 2018, 18:30:59 WIB].
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. IEEE Conf.Computer Vision and Pattern Recognition, pp. 511 –518, 2001.
- [7] PyImageSearch, "Raspberry Pi Face Recognition", Available: <https://www.pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/>, [Diakses 28 Sept 2018, 08:30:00 WIB].
- [8] Davis E. King, (2009). "Dlib-ml: A Machine Learning Toolkit". Journal of Machine Learning Research, vol 10, pp.1755-1758