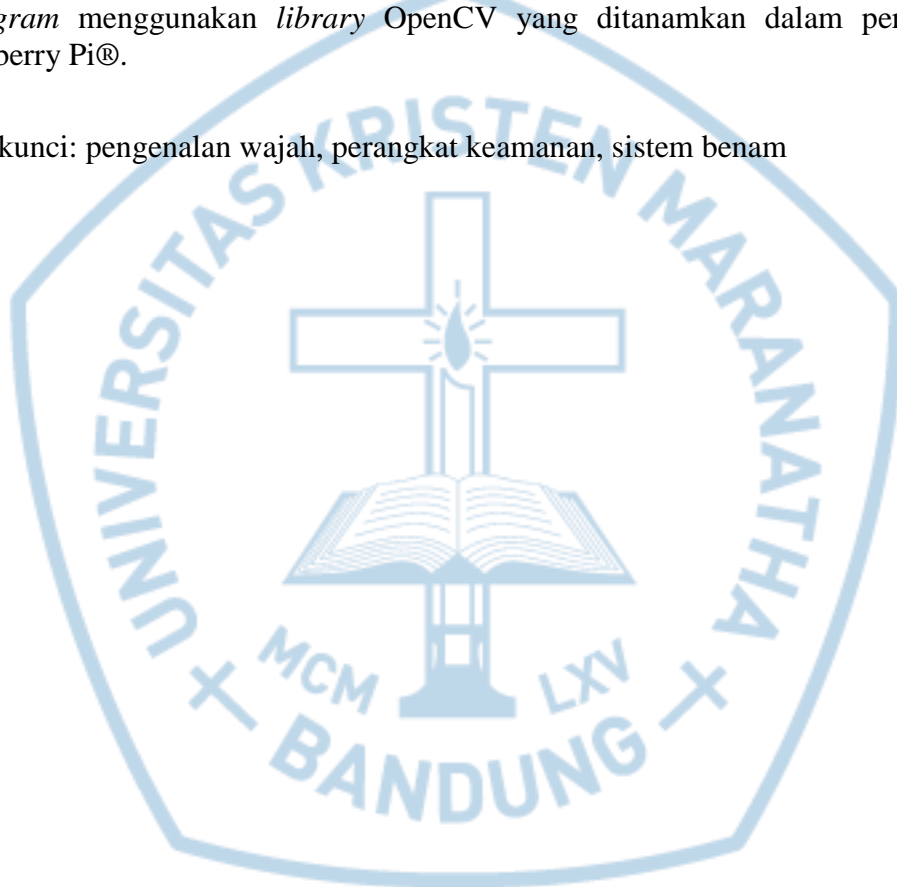


ABSTRAK

Laporan penelitian ini berisi mengenai pengembangan sebuah perangkat keamanan mobil yang berfungsi mencegah akses orang yang tidak berwenang dalam menghidupkan mesin mobil. Perangkat ini merupakan sebuah *embedded system* yang menggunakan algoritma pengenalan wajah untuk otentikasi pengemudi. Dengan menggunakan kombinasi dua buah algoritma pengenalan wajah (*Fisherface* dan *local binary pattern histogram*) dengan sistem voting, akurasi yang didapat adalah $90\% \pm 13\%$, dengan *false acceptance rate* sebesar 5% dan *false recognition rate* sebesar 7%.

Dalam penelitian ini, implementasi algoritma *Fisherface* dan *local binary pattern histogram* menggunakan *library* OpenCV yang ditanamkan dalam perangkat Raspberry Pi®.

Kata kunci: pengenalan wajah, perangkat keamanan, sistem benam



ABSTRACT

This project report is about a development of car security system device to prevents unauthorized access to ignite the car's engine. The device is an embedded system which uses face recognition algorithm for access authorization. By adopting a voting scheme of a combination of two well-known face recognition algorithms, namely Fisherface and local binary pattern histogram, the accuracy is significantly improved, which is $90\% \pm 13\%$, with 5% false acceptance rate and 7% false recognition rate.

In this project, implementation of the algorithms employs OpenCV library, which is embedded in a Raspberry Pi® device.

Keywords : embedded system, face recognition, security device

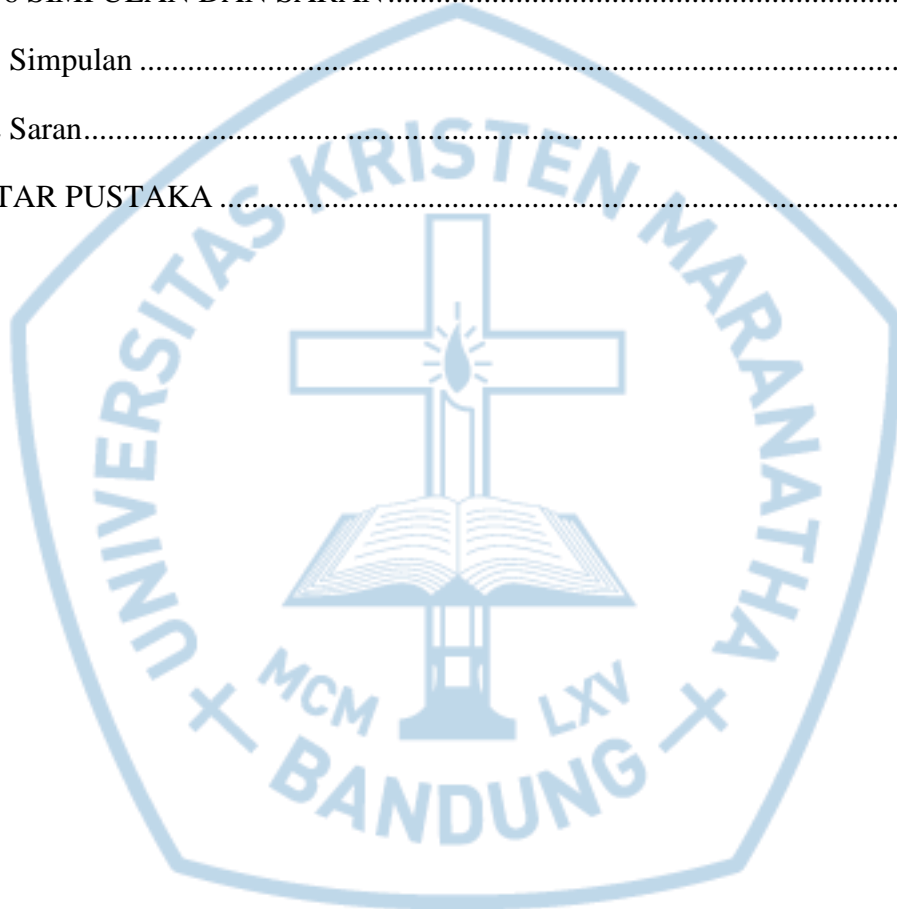


DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN.....	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR NOTASI/ LAMBANG.....	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Pembahasan	3
1.4 Ruang Lingkup.....	3
1.5 Sumber Data.....	4
1.6 Sistematika Penyajian	4
BAB 2 KAJIAN TEORI	6
2.1 Landasan Teori.....	6
2.1.1 Sistem Pengenalan Wajah	6
2.1.2 Face Detection Menggunakan Haar Cascade.....	6
2.1.3 Local Binary Patterns Histogram	8
2.1.4 Eigenfaces dan Fisherfaces	11
2.1.5 Raspberry Pi.....	13

2.1.6 OpenCV	14
2.2 Keaslian Penelitian.....	14
BAB 3 ANALISIS DAN RANCANGAN SISTEM.....	15
3.1 Analisis Sistem.....	15
3.2 Desain Sistem.....	16
3.2.1 Data Flow Diagram (DFD)	16
3.2.1.1 Data Flow Diagram Level 0.....	16
3.2.1.2 Data Flow Diagram Level 1	17
3.2.1.3 Data Flow Diagram Level 2.....	18
3.2.1.4 Data Flow Diagram Level 3	20
3.2.2 Circuit Diagram.....	24
3.3 Contoh Kasus Pemanfaatan Voting dalam Memutuskan Hasil Pengenalan Wajah	25
BAB 4 IMPLEMENTASI.....	29
4.1 Implementasi Metode.....	29
4.1.1 Inisialisasi dan Deklarasi Fungsi.....	29
4.1.2 Pengenalan Wajah.....	33
4.1.2.1 Array untuk Menampung Hasil Voting.....	34
4.1.2.2 Kamera Mendeteksi Wajah Secara Berulang.....	34
4.1.3 Menambah Pengemudi Baru	38
4.2 Implementasi Antar Muka.....	43
4.2.1 Tampilan Menu Awal	43
4.2.2 Menjalankan Sistem Pengenalan Wajah Pengemudi	43
4.2.3 Menambahkan Pengemudi Baru	44
4.2.4 Melihat Daftar Pengemudi	46
4.2.5 Menghapus Pengemudi dari Daftar.....	46

4.3 Implementasi Rangkaian dan Perangkat Keras.....	47
BAB 5 PENGUJIAN	49
5.1 Black-Box Testing	49
5.2 False Acceptance Rate dan False Recognition Rate Testing	55
5.2.1 False Acceptance Rate (FAR).....	59
5.2.2 False Recognition Rate (FRR)	64
BAB 6 SIMPULAN DAN SARAN.....	69
6.1 Simpulan	69
6.2 Saran.....	70
DAFTAR PUSTAKA	71



DAFTAR GAMBAR

Gambar 2.1 Contoh fitur untuk deteksi wajah	7
Gambar 3.1 Data Flow Diagram Level 0	17
Gambar 3.2 Data Flow Diagram Level 1	18
Gambar 3.3 Data Flow Diagram Level 2 untuk proses “1. Run Recognizer”	19
Gambar 3.4 Data Flow Diagram Level 2 untuk proses “2. Add New Driver”	20
Gambar 3.5 Data Flow Diagram Level 3 untuk proses “1.1 Preprocessing Images”	21
Gambar 3.6 Data Flow Diagram Level 3 untuk proses “1.5. Result Judgement”.	21
Gambar 3.7 Data Flow Diagram Level 3 untuk proses “2.2 Add 6 Driver Frontal Faces”	22
Gambar 3.8 Data Flow Diagram Level 3 untuk proses “2.3 Add 2 Driver Left Faces”	23
Gambar 3.9 Data Flow Diagram Level 3 untuk proses “2.4 Add 2 Driver Right Faces”	24
Gambar 3.10 Circuit diagram sistem	25
Gambar 4.1 Kode program untuk inisialisasi awal parameter	30
Gambar 4.2 Kode program untuk membuat CascadeClassifier menggunakan OpenCV	30
Gambar 4.3 Kode program untuk membuat OpenCV Recognizer	30
Gambar 4.4 Kode program untuk mengambil dan preprocessing face database ..	31
Gambar 4.5 Database wajah.....	31
Gambar 4.6 Kode program untuk training LBPH Recognizer.....	32
Gambar 4.7 Kode program untuk training Fisherface Recognizer	32
Gambar 4.8 Kode program untuk face detection	32
Gambar 4.9 Kode program inisialisasi kamera.....	33
Gambar 4.10 Kode program pemanggilan fungsi training recognizer.....	33
Gambar 4.11 Kode program untuk inisialisasi awal	34
Gambar 4.12 Kode program array hasil voting.....	34
Gambar 4.13 Kode program untuk deteksi wajah dan preprocessing gambar dari kamera.....	35


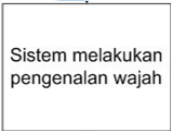



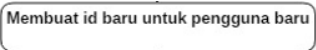
Gambar 4.14 Kode program recognizer melakukan voting terhadap gambar wajah	35
Gambar 4.15 Kode program perhitungan hasil voting	36
Gambar 4.16 Kode program untuk menghasilkan keputusan terhadap suatu pengemudi	36
Gambar 4.17 Kode program untuk mencari data pengemudi dari hasil voting	37
Gambar 4.18 Kode program untuk memberikan sinyal listrik ke rangkaian perangkat keras.....	37
Gambar 4.19 Kode program untuk menangani ketika tidak ada wajah terdeteksi	38
Gambar 4.20 Kode program untuk memasukkan nama driver baru	38
Gambar 4.21 Kode program untuk menyimpan data pengemudi baru	39
Gambar 4.22 Kode program untuk pengambilan 6 gambar secara frontal	40
Gambar 4.23 Kode program untuk pengambilan 2 gambar wajah menghadap kiri	41
Gambar 4.24 Kode program untuk pengambilan 2 gambar wajah menghadap kanan	42
Gambar 4.25 Tampilan menu awal	43
Gambar 4.26 Pilihan algoritma untuk pencarian wajah paling mirip	44
Gambar 4.27 Driver yang telah terdaftar kedalam sistem	44
Gambar 4.28 Menambahkan pengemudi dengan menghadap kamera.....	45
Gambar 4.29 Menambahkan pengemudi dengan wajah miring kiri terhadap kamera	45
Gambar 4.30 Menambahkan pengemudi dengan wajah miring kanan terhadap kamera	46
Gambar 4.31 Tampilan daftar pengemudi	46
Gambar 4.32 Tampilan saat menghapus pengemudi dari daftar	47
Gambar 4.33 Rangkaian perangkat keras.....	48
Gambar 5.1 Foto Malvin Mahesa sebagai pemilik	49
Gambar 5.2 Foto Michael Mahendra sebagai Mr. X	50
Gambar 5.3 Hasil pengujian Black Box.....	55
Gambar 5.4 Grafik hasil pengujian sesuai harapan untuk FAR.....	63
Gambar 5.5 Grafik hasil pengujian sesuai harapan untuk FRR.....	68



DAFTAR TABEL

Tabel 3.1 Contoh Kasus Voting Menghasilkan “ <i>Unknown</i> ”	27
Tabel 3.2 Contoh Kasus Voting Menghasilkan Nama.....	28
Tabel 5.1 Tabel pengujian secara umum.....	50
Tabel 5.2 Daftar Wajah Penguji.....	57
Tabel 5.3 Tabel pengujian untuk mencari <i>false acceptance rate</i>	60
Tabel 5.4 Tabel hasil pengujian sesuai harapan untuk <i>FAR</i>	62
Tabel 5.5 Tabel pengujian <i>false recognition rate</i>	64
Tabel 5.6 Tabel hasil pengujian sesuai harapan untuk <i>FRR</i>	67



DAFTAR NOTASI/ LAMBANG

Jenis	Notasi/ Lambang	Nama	Arti
Flowchart		Terminal	Disimbolkan menggunakan bentuk elips, sebagai penanda awal mula atau akhir sistem.
		Process	Disimbolkan menggunakan bentuk persegi panjang, sebagai penanda proses sistem.
		Decision	Disimbolkan menggunakan bentuk belah ketupat, sebagai penanda pemilihan keputusan.
		Arrow	Melambangkan hubungan antar simbol.
Activity Diagram		Initial Node	Disimbolkan menggunakan bentuk bulat penuh, sebagai penanda awal mula sistem.
		Action	Disimbolkan menggunakan bentuk persegi panjang, sebagai sebuah

			aktivitas dalam sistem.
		Control Flow	Disimbolkan menggunakan bentuk tanda panah kanan, sebagai alur aktivitas.
		Final Node	Disimbolkan menggunakan bentuk bulat penuh dilapisi bulat kosong, sebagai penanda akhir sistem.

Referensi:

Notasi/ Lambang *Flowchart* dari Visual Paradigm.

Notasi/ Lambang *Use Case Diagram* dari Visual Paradigm.

