



TUGAS AKHIR - EC184801

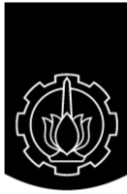
**DETEKSI KEDIPAN DENGAN METODE CNN DAN
PERCENTAGE OF EYELID CLOSURE (PERCLOS)**

Lutfi Ananditya Septiandi
NRP 0721 16 4000 0057

Dosen Pembimbing
Dr. Eko Mulyanto Yuniarno, S.T., M.T.
Ahmad Zaini, S.T., M.Sc.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan.



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**DETEKSI KEDIPAN DENGAN METODE CNN DAN
PERCENTAGE OF EYELID CLOSURE (PERCLOS)**

Lutfi Ananditya Septiandi
NRP 0721 16 4000 0057

Dosen Pembimbing
Dr. Eko Mulyanto Yuniarno, S.T., M.T.
Ahmad Zaini, S.T., M.Sc.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan.



FINAL PROJECT - EC184801

BLINKING DETECTION WITH CNN AND PERCENTAGE OF EYELID CLOSURE (PERCLOS) METHODS

Lutfi Ananditya Septiandi
NRP 0721 16 4000 0057

Advisors
Dr. Eko Mulyanto Yuniarno, S.T., M.T.
Ahmad Zaini, S.T., M.Sc.

Department of Computer Engineering
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

Halaman ini sengaja dikosongkan.

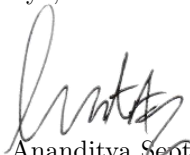
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Deteksi Kedipan dengan Metode CNN dan *Percentage of Eyelid Closure (PER-CLOS)***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 31 Januari 2021



Lutfi Ananditya Septiandi
NRP. 0721 16 4000 0057

Halaman ini sengaja dikosongkan.

LEMBAR PENGESAHAN

Deteksi Kedipan dengan Metode CNN dan *Percentage of Eyelid Closure* (PERCLOS)

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Lutfi Ananditya Septiandi (NRP: 07211640000057)

Tanggal Ujian : 5 Februari 2021

Periode Wisuda : April 2021

Disetujui oleh:

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP. 196806011995121009

(Pembimbing I)

Ahmad Zaini, S.T., M.Sc.
NIP. 197504192002121003

(Pembimbing II)

Arief Kurniawan, S.T., M.T.
NIP. 197409072002121001

(Penguji I)

Susi Juniastuti, S.T., M.Eng.
NIP. 196506181999032001

(Penguji II)



Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardhi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

Halaman ini sengaja dikosongkan.

ABSTRAK

- Nama Mahasiswa : Lutfi Ananditya Septiandi
Judul Tugas Akhir : Deteksi Kedipan dengan Metode CNN dan *Percentage of Eyelid Closure (PERCLOS)*
Dosen Pembimbing : 1. Dr. Eko Mulyanto Yuniarno, S.T., M.T.
2. Ahmad Zaini, S.T., M.Sc.

Pengembangan teknologi mengenai *face detection* dan *eyes detection* melaju sangat pesat, sehingga peneliti berlomba-lomba meneliti metode dan algoritma yang optimal untuk pengaplikasian di kehidupan sehari-hari, mulai dari pengamanan *biometrics* sampai identifikasi wajah secara automasi. Di tugas akhir ini diusulkan penggunaan metode *Convolutional Neural Network (CNN)* dan *Percentage of Eyelid Closure (PERCLOS)* pada pendeteksi kedipan mata. Sistem dibangun menggunakan *webcam personal computer* sebagai kamera dan mendeteksi secara *real-time*. Sistem dapat mengenali kondisi ketika mata tertutup atau mata terbuka dan menentukan lebar bukaan mata dengan menggunakan *Eye Aspect Ratio (EAR)* serta dapat mengestimasi skor tatapan dengan menggunakan *Percentage of Eyelid Closure (PERCLOS)*. Sistem dapat mengenali wajah dari objek bukan wajah dengan jarak pendeteksian optimal antara 40-70 cm. Model hasil *training* dapat mengklasifikasikan kondisi mata terbuka dan mata tertutup dengan menggunakan *Convolutional Neural Network* dengan arsitektur yang memiliki 3 *layer* mendapatkan hasil *accuracy* 98%.

Kata Kunci : *Face detection*, *Eyes detection*, *Convolutional Neural Network (CNN)*, *Eye Aspect Ratio (EAR)*, *Percentage of Eyelid Closure (PERCLOS)*.

Halaman ini sengaja dikosongkan.

ABSTRACT

Student Name : Lutfi Ananditya Septiandi
Final Project Title : *Blinking Detection with CNN and Percentage of Eyelid Closure (PERCLOS) Methods*
Advisors : 1. Dr. Eko Mulyanto Yuniarno, S.T., M.T.
2. Ahmad Zaini, S.T., M.Sc.

The development of technology on face detection and eyes detection is progressing very rapidly, so researchers are vying to research optimal methods and algorithms for application in everyday life, ranging from biometrics security to automatic facial identification. In this final projects is proposed the use of Convolutional Neural Network (CNN) and Percentage of Eyelid Closure (PERCLOS) methods on eye-blink detection. The systems is built using personal computer webcam as a camera and detecting in real-time. The systems can recognize conditions when eyes are closed or eyes open and determine the width of the eyes openings by using Eye Aspect Ratio (EAR) and can estimate stare scores by using Percentage of Eyelid Closure (PERCLOS). The system can recognize faces from non-face objects with an optimal detection distance between 40 until 70 cm. The training result model can classify the conditions of open and closed eyes by using a Convolutional Neural Network with an architecture that has 3 layers to get an accuracy of 98%.

Keywords : Face detection, Eyes detection, Convolutional Neural Network (CNN), Eye Aspect Ratio (EAR), Percentage of Eyelid Closure (PERCLOS).

Halaman ini sengaja dikosongkan.

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala rahmat-Nya, penulis dapat menyelesaikan tugas akhir ini dengan judul **Deteksi Kedipan dengan Metode CNN dan *Percentage of Eyelid Closure (PERCLOS)***. Penelitian ini disusun dalam rangka memenuhi salah satu persyaratan untuk memperoleh gelar sarjana di Departemen Teknik Komputer FTEIC ITS. Keberhasilan penulis dalam pelaksanaan penelitian ini tidak lepas dari bantuan banyak pihak yang terlibat. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Kedua orang tua, beserta keluarga, dan saudara-saudari tercinta yang selalu memberikan dukungan baik secara moral maupun material selama pelaksanaan penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. selaku Kepala Departemen Teknik Komputer FTEIC-ITS.
3. Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. dan bapak Ahmad Zaini, S.T., M.Sc. selaku dosen pembimbing, atas dukungan dan bimbingan selama pengerjaan penelitian ini.
4. Bapak-ibu dosen pengajar serta staff Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman dari Teknik Komputer dan juga angkatan e56.

Penulis menyadari bahwa dalam penyusunan buku hasil penelitian ini jauh dari kata sempurna, untuk itu penulis memohon kritik dan saran yang membangun sehingga dapat menjadi lebih baik lagi. Harapannya penelitian ini dapat berguna sebagai acuan penelitian-penelitian selanjutnya dan bermanfaat bagi kita semua. Aamiin.

Surabaya, Desember 2020

Penulis

Halaman ini sengaja dikosongkan.

DAFTAR ISI

ABSTRAK	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
NOMENKLATUR	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	2
2 TINJAUAN PUSTAKA	5
2.1 <i>CNN</i>	5
2.1.1 Kinerja Klasifikasi	6
2.1.1.1 <i>Accuracy</i>	7
2.1.1.2 <i>Error</i>	7
2.1.1.3 <i>Precision</i>	7
2.1.1.4 <i>Recall</i>	8
2.1.1.5 <i>F1 Score</i>	8
2.2 <i>Face Detection</i>	8
2.2.1 Pendeteksian Mata	9
2.2.2 <i>Eye Aspect Ratio (EAR)</i>	10
2.2.3 <i>Percentage of Eyelid Closure (PERCLOS)</i>	10
3 DESAIN DAN IMPLEMENTASI SISTEM	13
3.1 Gambaran Umum	13
3.2 Desain Sistem	13
3.3 <i>Preprocessing Dataset</i>	14

3.3.1	Pengumpulan <i>Dataset</i>	16
3.3.2	Pengelompokan <i>Dataset</i>	16
3.3.3	<i>Convert to Grayscale</i>	16
3.3.4	<i>Resize</i> Ukuran Citra	16
3.3.5	<i>Labeling</i> Citra	16
3.3.6	<i>Convert to Numpy Array</i>	16
3.4	<i>Training Dataset</i>	16
3.5	Pengaplikasian Model	17
3.5.1	Pembuatan <i>Region of Interest (RoI)</i>	18
3.5.2	<i>Real-Time Detection</i>	18
3.5.3	Klasifikasi Kategori Keadaan Mata	18
3.6	Estimasi Skor Tatapan Pengemudi dan <i>Eye Aspect Ratio (EAR)</i>	18
3.6.1	Menentukan Titik-titik Mata	19
3.6.2	Menghitung Lebar Bukaannya Mata	19
3.6.3	Estimasi Skor Tatapan	20
4	PENGUJIAN DAN ANALISIS	21
4.1	Pengujian Model Terhadap Klasifikasi Kedipan Mata	21
4.2	Pengujian Terhadap Video <i>Testing</i>	23
5	PENUTUP	43
5.1	Kesimpulan	43
5.2	Saran	43
	DAFTAR PUSTAKA	45
	BIOGRAFI PENULIS	47

DAFTAR GAMBAR

2.1	Ilustrasi arsitektur <i>Convolutional Neural Network (CNN)</i> [1].	5
2.2	Ilustrasi arsitektur <i>Convolutional Neural Network (CNN)</i> 3 layer.	6
2.3	Ilustrasi algoritma <i>Haar Cascade Classifier</i> dalam sistem pendeteksian mata[2].	9
2.4	Titik-titik <i>facial landmark</i> mata.	10
3.1	Alur kerja penggunaan sistem.	14
3.2	<i>Sample dataset</i> untuk <i>preprocessing</i>	15
3.3	Alur <i>preprocessing dataset</i>	15
3.4	<i>Output training</i> dengan menggunakan <i>Convolutional Neural Network (CNN)</i>	17
3.5	Alur pengaplikasian model secara <i>real-time</i>	18
3.6	Alur kerja menggunakan <i>Eye Aspect Ratio (EAR)</i> dan <i>PERcentage of eyelid CLOSure (PERCLOS)</i> . . .	19
4.1	Grafik <i>loss</i> , <i>confusion matrix</i> , dan <i>classification re- port</i> dari hasil evaluasi pengujian kedipan mata. . .	22
4.2	Hasil pendeteksian dengan menggunakan CNN dan PERCLOS.	23
4.3	<i>Data testing</i> 1-1.	23
4.4	<i>Data testing</i> 1-2.	24
4.5	<i>Data testing</i> 1-3.	24
4.6	<i>Data testing</i> 2-1.	25
4.7	<i>Data testing</i> 2-2.	25
4.8	<i>Data testing</i> 2-3.	26
4.9	<i>Data testing</i> 3-1.	26
4.10	<i>Data testing</i> 3-2.	27
4.11	<i>Data testing</i> 3-3.	27
4.12	<i>Data testing</i> 4-1.	28
4.13	<i>Data testing</i> 4-2.	28
4.14	<i>Data testing</i> 4-3.	29
4.15	<i>Data testing</i> 5-1.	29
4.16	<i>Data testing</i> 5-2.	30
4.17	<i>Data testing</i> 5-3.	30

4.18	<i>Data testing</i> 6-1.	31
4.19	<i>Data testing</i> 6-2.	31
4.20	<i>Data testing</i> 6-3.	32
4.21	<i>Data testing</i> 7-2.	32
4.22	<i>Data testing</i> 7-3.	33
4.23	<i>Data testing</i> 8-1.	33
4.24	<i>Data testing</i> 8-2.	34
4.25	<i>Data testing</i> 8-3.	34
4.26	<i>Data testing</i> 9-2.	35
4.27	<i>Data testing</i> 9-3.	35
4.28	<i>Data testing</i> 10-1.	36
4.29	<i>Data testing</i> 10-3.	36
4.30	<i>Data testing</i> 11-1.	37
4.31	<i>Data testing</i> 11-2.	37
4.32	<i>Data testing</i> 11-3.	38
4.33	<i>Data testing</i> 12-1.	38
4.34	<i>Data testing</i> 13-1.	39
4.35	<i>Data testing</i> 13-2.	39
4.36	<i>Data testing</i> 14-1.	40
4.37	<i>Data testing</i> 14-2.	40
4.38	<i>Data testing</i> 14-3.	41

NOMENKLATUR

- N_m : Jumlah *frame* mata yang terekam dalam 1 menit
 N_a : Jumlah *frame* mata berkedip yang terekam dalam 1 menit
 TP : *True Positive*
 FP : *False Positive*
 TN : *True Negative*
 FN : *False Negative*

Halaman ini sengaja dikosongkan.

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar Belakang

Perkembangan teknologi pada era masa kini sangatlah pesat apalagi dibidang komputer, hal ini ditandai dengan hampir semua pengelola data dan informasi telah dilakukan dengan komputer. Banyak pekerjaan manusia yang tergantikan dengan komputer dikarenakan sistem yang ada di komputer lebih optimal dan lebih efisien serta menghemat waktu oleh karena itu peran komputer didunia ini sangatlah penting sekarang.

Deteksi wajah merupakan salah satu teknologi yang sekarang ini banyak dikembangkan seiring berkembangnya teknologi komputer. Dengan berkembangnya teknologi deteksi wajah, penelitian mengenai *eyes detection* juga ikut berkembang pesat. Beberapa teknologi yang menggunakan deteksi mata antara lain pendeteksian kedipan mata. Ada banyak metode yang digunakan untuk mendeteksi kedipan mata, salah satunya adalah menggunakan *Percentage of Eyelid Closure (PERCLOS)*. Metode *Percentage of Eyelid Closure (PERCLOS)* digunakan diberbagai penelitian untuk menangkap kondisi mata pengemudi menggunakan sistem visi komputer. Selain itu, metode ini juga digunakan sebagai pengukuran durasi penutupan mata dan tingkat kedip[3].

Dengan adanya penilitian ini, diharapkan sistem dapat mendeteksi kondisi mata dan juga dapat mengestimasi skor tatapan pengemudi.

1.2 Permasalahan

Berdasarkan latar belakang diatas, perumusan masalah yang diangkat pada tugas akhir ini adalah:

1. Menentukan *Region of Interest (RoI)* dari wajah dan mata menggunakan *Haar Cascade Classifier*.

2. Menentukan titik-titik pada mata menggunakan *Facial Landmark*.
3. Pengukuran jarak pada lebar bukaan mata dengan menggunakan *Eye Aspect Ratio (EAR)* atau *Euclidean Distance*.
4. Mengestimasi skor tatapan pengemudi dengan menggunakan *Percentage of Eyelid Closure (PERCLOS)*.

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah membuat suatu sistem untuk mendeteksi kondisi mata dan lebar bukaan mata serta mengestimasi skor tatapan pengemudi secara *real-time* dengan metode CNN dan *Percentage of Eyelid Closure (PERCLOS)*.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut diantaranya adalah :

1. Jarak pendeteksian terbatas, sekitar 40-100cm dari *webcam personal computer*.
2. Intensitas cahaya *indoor*.
3. Arsitektur *Convolutional Neural Network* yang digunakan sebanyak 3 layer.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB 1 Pendahuluan
Bab I berisi uraian tentang latar belakang, permasalahan, tujuan, batasan masalah, metodologi, dan sistematika penulisan dari penelitian tugas akhir ini.
2. BAB 2 Tinjauan Pustaka
Bab II berisi tentang uraian secara sistematika teoriteori yang

berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu teori *Convolutional Neural Network (CNN)*, *Face detection*, dan teori penunjang lain termasuk *datasheet* spesifikasi setiap komponen lain yang membangun sistem.

3. BAB 3 Perancangan Sistem dan Implementasi

Bab III berisi tentang rancangan pemecahan masalah beserta implementasinya. Berisikan mengenai perencanaan rancangan, uraian rinci mengenai metodologi yang digunakan, dan pemaparan hasil implementasi sistem yang dilakukan.

4. BAB 4 Pengujian dan Analisis

Bab IV berisi tentang pengujian eksperimen yang telah dilakukan terhadap sistem dan parameter yang terlibat. Selain itu, bab ini juga memuat hasil dan analisis dari uji coba yang dilakukan pada Tugas Akhir ini.

5. BAB 5 Penutup

BAB V berisi tentang kesimpulan yang diambil berdasarkan penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

Halaman ini sengaja dikosongkan.

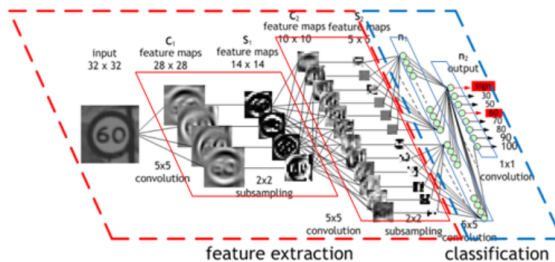
BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Pada bagian ini, teori penunjang tersebut dijabarkan secara ringkas untuk menjadi dasar dalam menyelesaikan penelitian yang lebih terarah.

2.1 CNN

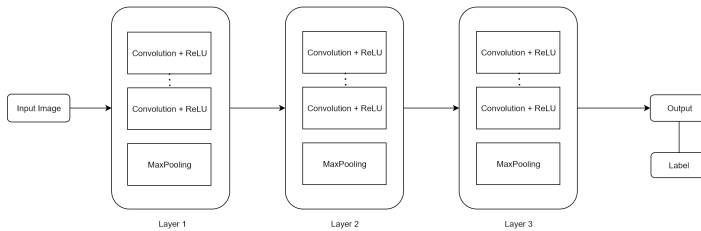
Convolutional Neural Network (CNN) merupakan *trainable multistage architectures* yang terdiri dari beberapa tahapan *input* dan *output*[4]. *Convolutional Neural Network (CNN)* termasuk dalam jenis *Deep Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. Ada dua metode yang digunakan dalam *Convolutional Neural Network (CNN)* yaitu: klasifikasi menggunakan *feedforward* dan tahap pembelajaran menggunakan *backpropagation*. Cara kerja *Convolutional Neural Network (CNN)* memiliki kesamaan pada *Multi Layer Preceptron (MLP)*, namun dalam *Convolutional Neural Network (CNN)* setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti *Multi Layer Preceptron (MLP)* yang setiap neuron hanya berukuran satu dimensi[1]. Gambar 2.1 adalah ilustrasi dari arsitektur *Convolutional Neural Network (CNN)*.



Gambar 2.1: Ilustrasi arsitektur *Convolutional Neural Network (CNN)*[1].

Pada tugas akhir ini menggunakan *Convolutional Neural Ne-*

work (CNN) yang memiliki 3 *layer* dan pada gambar 2.2 merupakan arsitektur yang digunakan. Ada 3 lapisan didalam arsitektur CNN, yaitu *convolutional layer*, *fully connected layer*, dan *subsampling layer*. *Convolutional layer* dilengkapi dengan *activation rectified linear unit* atau aktivasi ReLU. *Subsampling layer* menggunakan fungsi *max pooling* dimana fungsi ini digunakan untuk mengurangi jumlah parameter dan perhitungan jaringan, dan untuk mengendalikn *overfitting*. *Fully connected layer* menggunakan fungsi aktivasi *softmax*.



Gambar 2.2: Ilustrasi arsitektur *Convolutional Neural Network (CNN)* 3 *layer*.

2.1.1 Kinerja Klasifikasi

Dalam pengukuran kinerja suatu sistem klasifikasi menggambarkan kinerja sistem tersebut untuk mengklasifikasikan data. Salah satu metode yang dapat digunakan untuk mengukur kinerja suatu sistem klasifikasi adalah *confusion matrix*. *Confusion matrix* berisi perbandingan hasil klasifikasi dari sistem dengan hasil yang seharusnya diklasifikasikan.

Tabel 2.1: *Confusion Matrix*

<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
<i>Negative</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>
	<i>Positive</i>	<i>Negative</i>

Confusion matrix merupakan tabel dua dimensi yang terdiri dari data aktual dan data prediksi serta memiliki kelas masing-masing, dimana data aktual terletak pada bagian kolom sedangkan

data prediksi terletak pada bagian baris. Tabel 2.1 merupakan tabel *confusion matrix*. Hasil dari tabel *confusion matrix* dapat digunakan untuk berbagai pengukuran performa yang lebih detail.

2.1.1.1 Accuracy

Accuracy digunakan untuk mencari keakuratan nilai dari sebuah model yang dapat mengklasifikasikan data yang benar. Oleh karena itu, *accuracy* adalah jumlah klasifikasi data yang benar dari jumlah data keseluruhan. Nilai *accuracy* dapat diperoleh dari persamaan 2.1.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (2.1)$$

2.1.1.2 Error

Error merupakan jumlah klasifikasi data yang salah dari jumlah data keseluruhan. Ada dua macam *errors* dalam *confusion matrix* yaitu: Tipe I dan Tipe II. *False Positive* berisi satu kata *negative (False)* sehingga *False Positive* merupakan *error* tipe I. *False Negative* memiliki dua kata *negative (False + Negative)* sehingga *False Negative* merupakan *error* tipe II. Nilai *error* dapat diperoleh dari persamaan 2.2.

$$Error = \frac{FP + FN}{TP + FN + FP + TN} \times 100\% \quad (2.2)$$

2.1.1.3 Precision

Precision digunakan untuk mencari tingkat keakuratan antara data yang diminta dan hasil prediksi. Oleh karena itu, *precision* merupakan rasio prediksi *True Positive* terhadap prediksi *positive* secara keseluruhan. Nilai *precision* dapat diperoleh dari persamaan 2.3.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2.3)$$

2.1.1.4 *Recall*

Recall digunakan untuk mencari keberhasilan model dalam memulihkan kembali informasi. Oleh karena itu, *recall* merupakan rasio antara prediksi *true positive* dengan data *true positive* secara keseluruhan. Nilai *recall* dapat diperoleh dari persamaan 2.4.

$$Accuracy = \frac{TP}{TP + FN} \times 100\% \quad (2.4)$$

2.1.1.5 *F1 Score*

F1 Score merupakan nilai antara 0 dan 1 yang dimana diperoleh dari rata-rata harmonik antara *precision* dan *recall*. Nilai *F1 Score* dapat diperoleh dari persamaan 2.5.

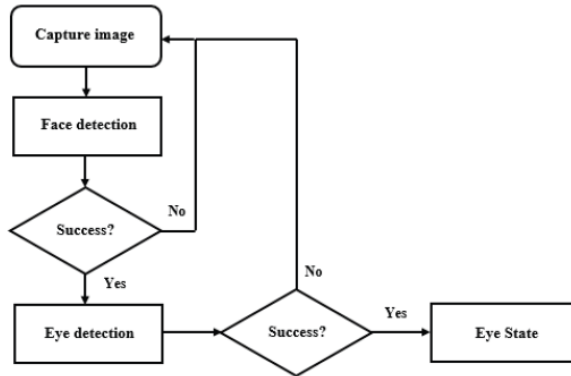
$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.5)$$

2.2 *Face Detection*

Wajah merupakan salah satu bagian dari manusia yang memiliki ciri berbeda dari setiap manusia[5]. Bagian dari wajah manusia meliputi mata, hidung, pipi, mulut, dahi, alis, dagu dan juga rambut. Dengan adanya perkembangan teknologi, peneliti menemukan teknologi untuk mendeteksi wajah yang lebih umum dikenal sebagai *face detection*. *Face detection* merupakan sebuah teknologi untuk mendeteksi wajah yang dibangun menggunakan algoritma tertentu dan berfokus untuk mendeteksi wajah manusia dalam bentuk *digital images*. *Face detection* biasanya digunakan untuk mendeteksi orang lalu lalang, mengkalkulasi ras, *gender*, rentang usia wajah, fotografi, raut muka, dan lain-lain. Pengaplikasian dari *face detection* menggunakan algoritma dan *machine learning* untuk menemukan wajah manusia dalam citra yang memiliki objek bukan wajah. Algoritma pendeteksian wajah biasanya dimulai dari mata manusia, dikarenakan mata merupakan salah satu fitur termudah untuk didektesi. Kemudian pencarian lainnya bisa saja mendeteksi alis, mulut, hidung, lubang hidung, dan iris. Setelah algoritma dapat menyim-

pulkan pendeteksian pada area wajah, langkah selanjutnya adalah sistem akan melakukan pendeteksian pada wajah tersebut dan menentukan terdeteksinya wajah atau tidak[6].

2.2.1 Pendeteksian Mata

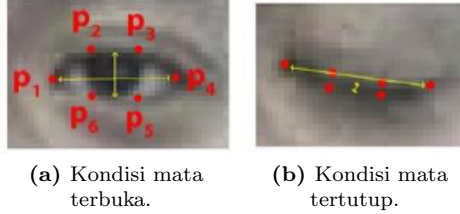


Gambar 2.3: Ilustrasi algoritma *Haar Cascade Classifier* dalam sistem pendeteksian mata[2].

Salah satu pengembangan dari *face detection* adalah *eyes detection*. *Eyes detection* merupakan pengembangan lanjut dari deteksi wajah, dimana citra wajah manusia yang berhasil dideteksi akan diproses kembali dengan mendeteksi kedua mata pada wajah[7]. Salah satu metode pendeteksian mata adalah dengan menggunakan *Haar Cascade Classifier*. *Haar Cascade Classifier* merupakan salah satu *library* dari *OpenCV* yang dapat mengenali posisi dari objek yang ingin dideteksi dengan cara memberikan *Region of Interest (RoI)* kepada objek tersebut. *Cascade Classifier* merupakan langkah untuk mendapatkan hasil yang lebih akurat dengan menghitung nilai *Haar Feature* secara banyak dan berulang. Gambar 2.3 merupakan ilustrasi dari algoritma *Haar Cascade Classifier* dalam mendeteksi mata. Pada tahap pertama, penangkapan dan pendeteksian citra wajah. Pada tahap kedua, ketika wajah dapat di deteksi proses selanjutnya adalah pendeteksian citra mata, bila tidak terdeteksi maka proses selanjutnya menangkap kembali citra wajah. Pada tahap ketiga, ketika citra mata dapat dideteksi proses

selanjutnya menentukan kondisi mata, bila tidak terdeteksi maka proses selanjutnya kembali ke tahap pertama[2].

2.2.2 Eye Aspect Ratio (EAR)



Gambar 2.4: Titik-titik *facial landmark* mata.

Eye Aspect Ratio (EAR) adalah proporsi antara lebar dan ketinggian bukaan kelopak mata atas dan kelopak mata bawah. Persamaan 2.6 digunakan untuk menghitung jarak antara kelopak mata atas dan kelopak mata bawah[8]. Dimana P_1, \dots, P_6 adalah lokasi 2D dari *facial landmark* mata. Gambar 2.4 merupakan titik-titik *facial landmark* mata. Penempatan titik-titik *facial landmark* pada mata memiliki 6 titik. Titik P_1 dan titik P_4 merupakan dua titik yang menghubungkan garis horizontal di mata, titik P_2 dan titik P_3 merupakan titik kelopak mata atas, dan titik P_5 dan titik P_6 merupakan titik kelopak mata bawah. Nilai dari *Eye Aspect Ratio (EAR)* akan dengan cepat turun hampir ke nilai nol ketika titik kelopak mata atas bertemu dengan garis horizontal atau keadaan ketika mata berkedip.

$$EAR = \frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2\|P_1 - P_4\|} \quad (2.6)$$

2.2.3 Percentage of Eyelid Closure (PERCLOS)

PERcentage of eyelid CLOSure (PERCLOS) merupakan metode yang biasa digunakan peneliti untuk menunjukkan durasi penutupan mata dan tingkat kedipan mata. Selain itu *PERcentage*

of eyelid CLOSure (PERCLOS) bisa juga didefinisikan sebagai proporsi waktu di mana kelopak mata tertutup lebih dari 70%-80% dalam jangka waktu yang ditentukan. Persamaan 2.7 yang digunakan untuk menghitung *PERcentage of eyelid CLOSure (PERCLOS)* dalam satu menit[9].

$$PERCLOS = \frac{Nm - Na}{Nm} \times 100\% \quad (2.7)$$

Halaman ini sengaja dikosongkan.

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem serta implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Gambaran Umum

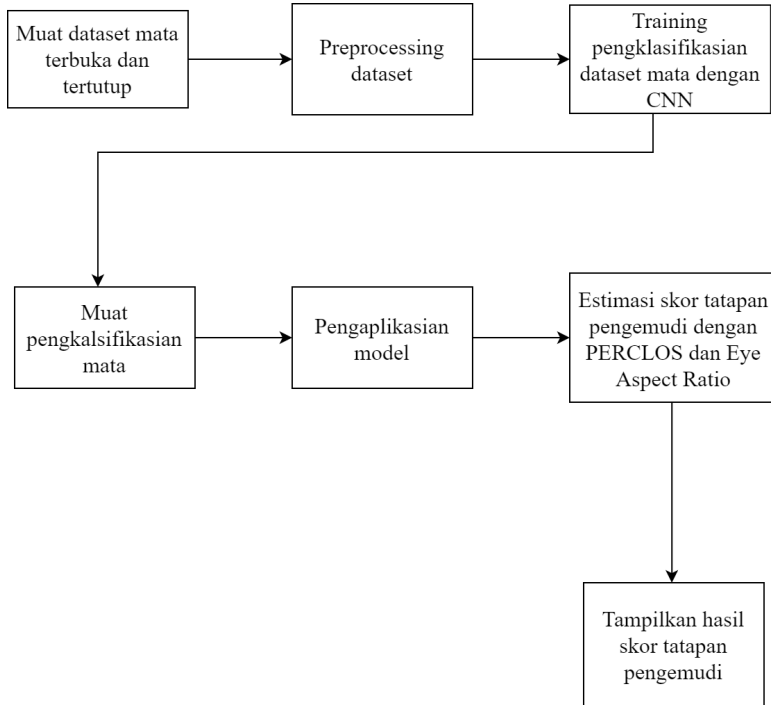
Penelitian ini bertujuan untuk mendeteksi kedipan dari pengemudi dan dapat mengestimasi skor tatapan dari pengemudi beserta kondisi mata pengemudi. Sistem ini dimulai dari input berupa citra mata dalam keadaan tertutup dan terbuka. Kemudian dari kumpulan citra tersebut di-*training* menggunakan CNN dan mendapatkan pengklasifikasian mata tertutup dan terbuka. Setelah mendapatkan hasil model *training*, model yang didapat digunakan untuk mendeteksi kondisi mata dan mengestimasi skor tatapan pengemudi beserta lebar bukaan kelopak mata secara *real-time*.

3.2 Desain Sistem

Pada sistem ini pengemudi dapat mengestimasi skor tatapan dengan menggunakan metode *PERcentage of eyelid CLOSure (PERCLOS)*. Kondisi mata yang telah di-*training* menggunakan CNN, akan di deteksi menggunakan kamera dan mengekstrasi ROI dari mata kiri dan kanan. Kemudian hasil yang didapat berupa persentase dari lebar bukaan kelopak mata atas dan bawah dengan menggunakan *ecludian distance*. Alur kerja dari sistem ini dijelaskan melalui gambar 3.1.

Terdapat 4 bab yang membangun tujuan dari penelitian ini, yaitu pengumpulan *dataset* mengenai pengelompokan kondisi mata, kemudian melakukan *training dataset* dengan menggunakan CNN, selanjutnya mengaplikasikan model *training* secara *real-time* menggunakan kamera agar mendapatkan kondisi mata, yang terakhir adalah mengestimasi skor tatapan pengemudi dengan mengguna-

kan metode *PERcentage of eyelid CLOSure (PERCLOS)* dan menentukan lebar bukaan mata dengan *Euclidean Distance*.



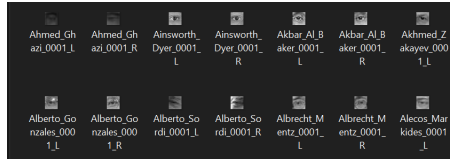
Gambar 3.1: Alur kerja penggunaan sistem.

3.3 *Preprocessing Dataset*

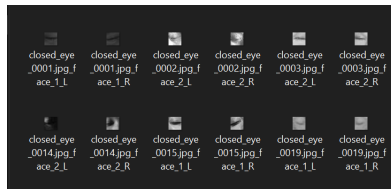
Pada tahapan ini, dilakukan pengumpulan dan pemrosesan awal agar *dataset* siap digunakan untuk proses selanjutnya. *Dataset* yang digunakan dalam tugas akhir ini berupa potongan bagian mata kanan dan kiri secara terpisah yang berasal dari dokumentasi penelitian sebelumnya [10] dan dokumentasi pribadi. Kemudian *dataset* diubah dalam format *grayscale* dan dikelompokan berdasarkan keadaan mata, yaitu: keadaan mata terbuka (*open*) dan keadaan mata tertutup (*close*) sesuai pada gambar 3.2.



(a) Pengelompokan *dataset* berdasarkan kategori kondisi mata.



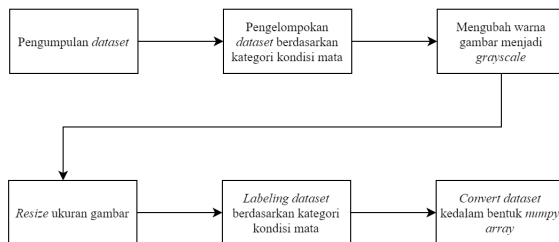
(b) *Sample* dari *dataset* dengan kondisi mata *open*.



(c) *Sample* dari *dataset* dengan kondisi mata *close*.

Gambar 3.2: *Sample dataset* untuk *preprocessing*

Setelah gambar mata di ubah dalam bentuk *grayscale*, selanjutnya gambar di ubah atau di-*convert* dalam bentuk *numpy array*. Gambar 3.3 merupakan alur dari *preprocessing*.



Gambar 3.3: Alur *preprocessing dataset*.

3.3.1 Pengumpulan *Dataset*

Dataset yang digunakan dalam pengerjaan tugas akhir ini berupa potongan mata kiri dan kanan yang berasal dari github[10] dan dokumentasi pribadi. Total *dataset* sebanyak 4,868 citra dengan rincian 2,476 citra mata terbuka (*open*) dan 2,392 citra mata tertutup (*close*).

3.3.2 Pengelompokan *Dataset*

Setelah mengumpulkan *dataset*, proses selanjutnya mengelompokkan *dataset* berdasarkan kondisi mata. *Dataset* dikelompokkan menjadi dua kategori, yaitu *open* dan *close*.

3.3.3 *Convert to Grayscale*

Proses selanjutnya adalah merubah warna potongan citra dari *RGB* ke *grayscale*. Proses ini menggunakan *cv2.cvtColor()* dimana fungsi ini digunakan untuk merubah citra dari satu *color space* ke *color space* lainnya.

3.3.4 *Resize Ukuran Citra*

Setelah warna citra menjadi *grayscale*, ukuran citra diubah menjadi 50×50 *pixel* menggunakan fungsi *cv2.resize()*. Fungsi ini digunakan untuk merubah tinggi dan lebar dari citra.

3.3.5 *Labeling Citra*

Kemudian citra yang sudah di-*resize* akan di beri *label* sesuai dengan kategori kondisi mata dan dijadikan dalam bentuk *list*. Fungsi *append* merupakan fungsi yang digunakan untuk mejadikan dalam bentuk *list*.

3.3.6 *Convert to Numpy Array*

Proses terakhir mengubah *list* ke dalam *numpy array*. Dalam pengubah bentuk *array* menggunakan metode *reshape*, dimana metode ini paling *flexible* untuk mengubah bentuk suatu *array* sesuai keinginan.

3.4 *Training Dataset*

Setelah melakukan proses *preprocessing*, proses selanjutnya merupakan *training dataset*. Proses ini bertujuan untuk melatih kom-

puter agar mencapai sebuah keputusan atau prediksi dengan cara mengolah gambar sehingga terbentuk pola atau karakteristik. Total *dataset* yang di-*training* sebanyak 4,868 citra dengan rincian 2,476 citra mata terbuka (*open*) dan 2,392 citra mata tertutup (*close*).

model-001.model	2/17/2021 9:51 PM	MODEL File	9,466 KB
model-002.model	2/17/2021 9:54 PM	MODEL File	9,466 KB
model-003.model	2/17/2021 9:57 PM	MODEL File	9,466 KB
model-004.model	2/17/2021 10:00 PM	MODEL File	9,466 KB
model-005.model	2/17/2021 10:03 PM	MODEL File	9,466 KB
model-006.model	2/17/2021 10:06 PM	MODEL File	9,466 KB
model-007.model	2/17/2021 10:09 PM	MODEL File	9,466 KB
model-008.model	2/17/2021 8:57 PM	MODEL File	9,466 KB
model-009.model	2/17/2021 10:15 PM	MODEL File	9,466 KB
model-011.model	2/17/2021 10:21 PM	MODEL File	9,466 KB
model-012.model	2/17/2021 9:03 PM	MODEL File	9,466 KB
model-013.model	2/17/2021 10:25 PM	MODEL File	9,466 KB
model-015.model	2/17/2021 10:28 PM	MODEL File	9,466 KB
model-016.model	2/17/2021 9:09 PM	MODEL File	9,466 KB
model-017.model	2/17/2021 9:10 PM	MODEL File	9,466 KB
model-020.model	2/17/2021 10:36 PM	MODEL File	9,466 KB
model-026.model	2/17/2021 10:45 PM	MODEL File	9,466 KB
model-028.model	2/17/2021 10:48 PM	MODEL File	9,466 KB
model-029.model	2/17/2021 10:50 PM	MODEL File	9,466 KB
model-030.model	2/17/2021 10:51 PM	MODEL File	9,466 KB

Gambar 3.4: *Output training* dengan menggunakan *Convolutional Neural Network (CNN)*.

Dataset yang dikumpulkan akan diproses dengan melakukan *training* yang menggunakan *Convolutional Neural Network (CNN)* dengan 3 *layer* dan ada sebanyak 30 *epochs*. Ketika *epochs* sudah mencapai 30, maka secara otomatis proses *training* akan berhenti. *Output* dari *training* berupa model yang nantinya akan digunakan untuk proses pendeteksian. Pada gambar 3.4 merupakan hasil dari *training* yang dilakukan.

3.5 Pengaplikasian Model

Proses yang ketiga adalah mengaplikasikan model yang sudah di-*training*. Pada proses ini dilakukan pendeteksian pada wajah dan mata dengan menggunakan *Haar Cascade Classifier* agar mendapatkan *Region of Interest (RoI)* pada bagian wajah dan mata. Ketika wajah dan mata sudah dapat dideteksi, maka langkah selanjutnya adalah model yang sudah di-*training* akan di muat untuk proses pendeteksian. Selanjutnya melakukan *labeling* pada pendeteksi mata dan menentukan kondisi mata terbuka (*open*) atau mata tertutup (*close*). Gambar 3.5 merupakan alur dari pendeteksian

secara *real-time*.

3.5.1 Pembuatan *Region of Interest (RoI)*

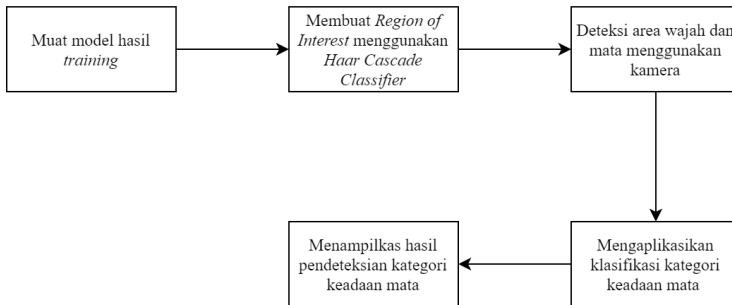
Pembuatan *region of interest (RoI)* untuk area wajah serta mata kanan dan kiri menggunakan *Haar Cascade Classifier*. Setelah itu membuat bentuk persegi untuk area wajah dan mata.

3.5.2 *Real-Time Detection*

Setelah mendapatkan *region of interest (RoI)* dari area wajah dan mata, selanjutnya pendeteksian dilakukan secara *real-time* dengan menggunakan *webcam PC*.

3.5.3 Klasifikasi Kategori Keadaan Mata

Setelah wajah dan mata dapat dideteksi, proses yang terakhir adalah mengaplikasikan klasifikasi kategori keadaan mata. Sistem akan menentukan kondisi mata seseorang ketika mata terbuka (*open*) atau mata tertutup (*close*).



Gambar 3.5: Alur pengaplikasian model secara *real-time*.

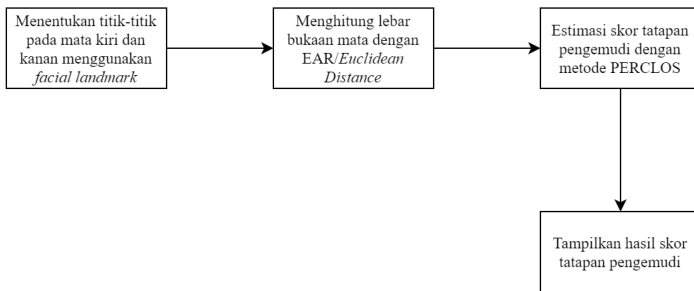
3.6 Estimasi Skor Tatapan Pengemudi dan *Eye Aspect Ratio (EAR)*

Ketika kamera sudah dapat mengenali kondisi mata terbuka atau tertutup, selanjutnya sistem akan menganalisa lebar bukaan mata. Cara menentukan lebar bukaan mata dengan menggunakan *Eye Aspect Ratio*. Untuk dapat menggunakan *Eye Aspect Ratio*

(*EAR*), yang dibutuhkan adalah menentukan titik-titik pada mata. Untuk menentukan titik-titik pada mata, bisa menggunakan *facial recognition landmark*.

Facial recognition landmark merupakan *pre-trained detector* yang berada didalam *dlib library* di *python* untuk mendeteksi landmark pada wajah. Ada 86 titik-titik atau koordinat dimana memetakan struktur wajah dimulai dari hidung, rahang bawah, mata kiri, mata kanan, alis kiri, alis kanan, dan mulut.

Setelah mendapatkan titik-titik dari mata dan nilai *Eye Aspect Ratio (EAR)*, proses selanjutnya adalah mengestimasi skor pengemudi dengan menggunakan metode *Percentage of Eyelid Closure*. Gambar 3.6 merupakan alur kerja penggunaan *Eye Aspect Ratio (EAR)* dan *PERcentage of eyelid CLOSure (PERCLOS)*.



Gambar 3.6: Alur kerja menggunakan *Eye Aspect Ratio (EAR)* dan *PERcentage of eyelid CLOSure (PERCLOS)*.

3.6.1 Menentukan Titik-titik Mata

Untuk menentukan titik-titik mata kiri dan kanan menggunakan *facial recognition landmark*. Pada tugas akhir ini hanya menggunakan enam titik yang digunakan untuk tiap mata kiri dan kanan, dimana titik-titik ini berkaitan dengan proses perhitungan lebar bukaan mata.

3.6.2 Menghitung Lebar Bukaan Mata

Proses selanjutnya menghitung lebar bukaan mata. Untuk menghitung lebar bukaan mata menggunakan *Eye Aspect Ratio (EAR)*,

dimana *Eye Aspect Ratio (EAR)* merupakan proporsi antara lebar dan ketinggian bukaan kelopak mata atas dan kelopak mata bawah.

3.6.3 Estimasi Skor Tatapan

Proses yang terakhir adalah mengestimasi skor tatapan. Untuk mengestimasi skor tatapan menggunakan *PERcentage of eyelid CLOSure (PERCLOS)*. *PERcentage of eyelid CLOSure (PERCLOS)* digunakan untuk menunjukkan durasi penutupan mata dan tingkat kedipan mata.

BAB 4

PENGUJIAN DAN ANALISIS

Pada bab ini menguraikan hasil pengujian serta analisa dari pengujian terhadap sistem yang telah dirancang pada Tugas Akhir ini. Pengujian ini dilakukan untuk menguji kemampuan sistem yang telah dibuat dalam menjawab permasalahan yang diangkat. Pengujian yang dilakukan adalah mengaplikasikan pendeteksian dengan menggunakan CNN dan PERCLOS. Spesifikasi dari *Personal Computer (PC)* yang dipakai untuk pengujian ada di tabel 4.1.

Tabel 4.1: Spesifikasi *Personal Computer (PC)*

GPU	NVIDIA GeForce MX330
CPU	Intel(R) Core(TM) i3-1005G1 CPU @ 1.2GHz (4 CPUs), ~1.2GHz
Storage	M.2 512 GB SSD
Camera	0.3MP
Display	15.6 inches, IPS FHD, 1920 x 1080
Connectivity	Wi-Fi 2x2 AC, 2 x USB 3.1 (Gen1) / USB 2.0
Memory	4 GB
OS	Windows 10 Home Single Language

4.1 Pengujian Model Terhadap Klasifikasi Kedipan Mata

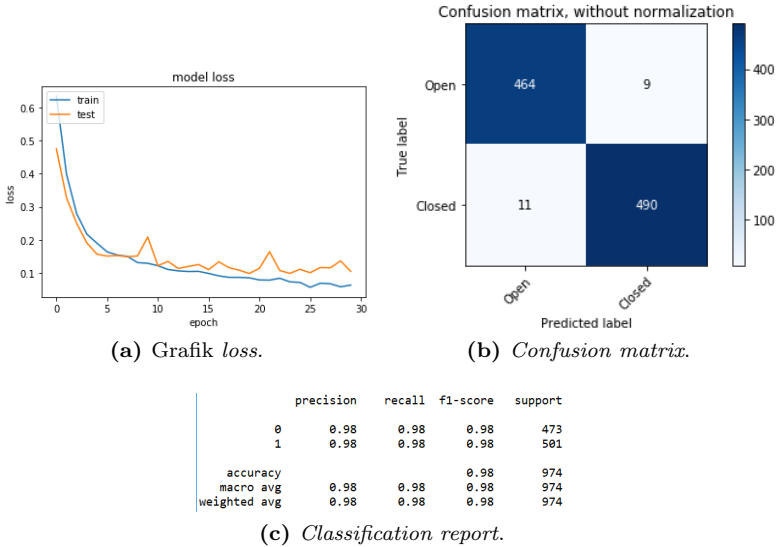
Pengujian model yang dilakukan pada ruangan *indoor* dengan pencahayaan dari lampu Hannochs PLC 4U 36 watt berwarna *cool daylight* atau warna putih. Adapun lima kriteria yang menentukan berhasil atau tidaknya pendeteksian adalah sebagai berikut:

1. Nilai dari *Eye Aspect Ratio (EAR)*.
2. Estimasi skor tatapan pengemudi dengan menggunakan *Per-*

centage of Eyelid Closure (PERCLOS).

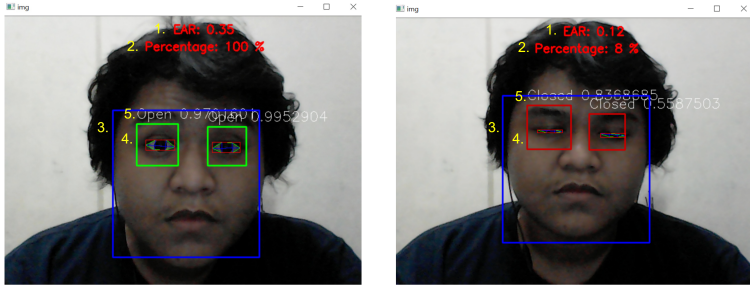
3. *Region of Interest (RoI)* dari wajah.
4. *Region of Interest (RoI)* dari mata.
5. Kondisi mata ketika terbuka (*open*) atau tertutup (*close*) serta nilai *confidence* dari prediksi *training*.

Pada klasifikasi kondisi mata menggunakan *epochs* sebesar 30 mendapatkan akurasi sebesar 98%, *Train loss* sebesar 0.6%, *Validation loss* sebesar 1.05%, dan *error* sebesar 2.05%. Gambar 4.1 merupakan grafik *loss*, *confusion matrix*, dan *classification report* dari hasil evaluasi pengujian kedepan mata.



Gambar 4.1: Grafik *loss*, *confusion matrix*, dan *classification report* dari hasil evaluasi pengujian kedepan mata.

Gambar 4.2 merupakan hasil dari pengujian model yang dilakukan secara *real-time* menggunakan CNN dan PERCLOS untuk mendapatkan lima kriteria dan menentukan berhasil atau tidaknya sistem pendeteksian.



(a) Kondisi mata terbuka.

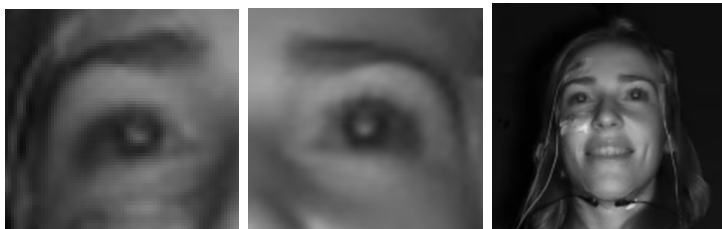
(b) Kondisi mata tertutup.

Gambar 4.2: Hasil pendeteksian dengan menggunakan CNN dan PERCLOS.

4.2 Pengujian Terhadap Video *Testing*

Pengujian menggunakan video *testing* terbagi menjadi 36 video dan setiap video memiliki durasi yang berbeda-beda untuk mengetahui kondisi mata terbuka atau tertutup, nilai lebar bukaan mata, dan persentase bukaan mata.

Dalam pengujian ini menggunakan *dataset* yang berbeda dari pengujian sebelumnya. *Dataset* yang digunakan dalam pengujian kali ini berupa potongan citra mata kanan dan kiri dari video *testing*. Total *dataset* yang digunakan berjumlah 4,096 dengan rincian 2,048 mata terbuka (*open*) dan 2,048 mata tertutup (*close*).



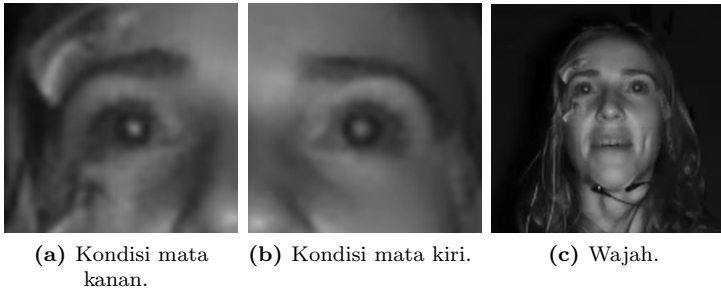
(a) Kondisi mata kanan.

(b) Kondisi mata kiri.

(c) Wajah.

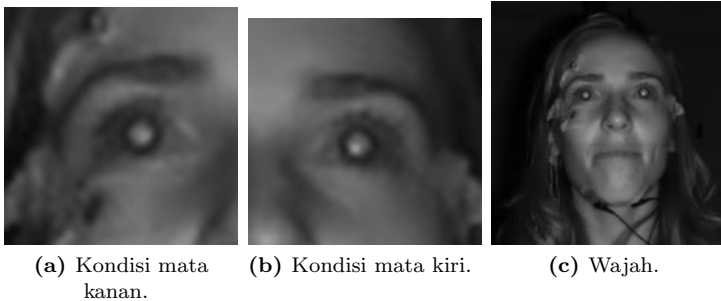
Gambar 4.3: *Data testing* 1-1.

Pengujian pertama menggunakan video *testing* 1-1. Dalam video *testing* 1-1 mempunyai 23,687 *frames*, dengan rincian 19,661 *frames* yang terdeteksi dan 4,026 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 5,822 *frames*. Gambar 4.3 merupakan *data testing* 1-1.



Gambar 4.4: *Data testing* 1-2.

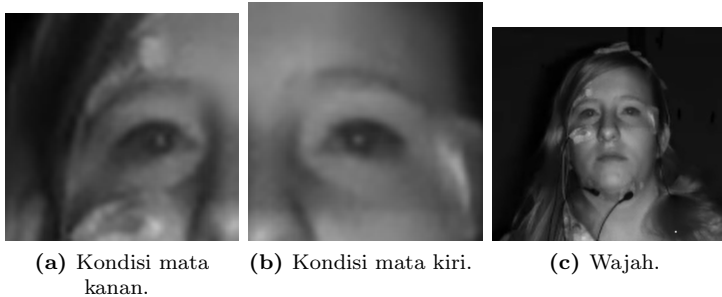
Pengujian kedua menggunakan video *testing* 1-2. Dalam video *testing* 1-2 mempunyai 9,499 *frames*, dengan rincian 3,612 *frames* yang terdeteksi dan 5,887 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 2 *frames*. Gambar 4.4 merupakan *data testing* 1-2.



Gambar 4.5: *Data testing* 1-3.

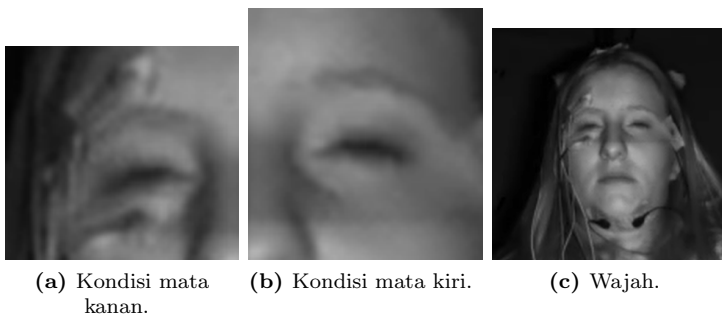
Pengujian ketiga menggunakan video *testing* 1-3. Dalam video

testing 1-3 mempunyai 8,873 *frames*, dengan rincian 1,732 *frames* yang terdeteksi dan 7,141 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 7 *frames*. Gambar 4.5 merupakan *data testing* 1-3.



Gambar 4.6: *Data testing* 2-1.

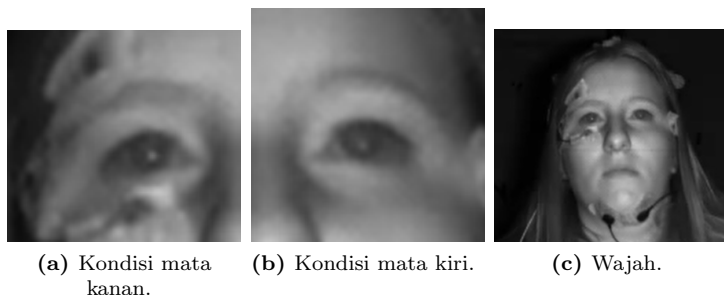
Pengujian keempat menggunakan video *testing* 2-1. Dalam video *testing* 2-1 mempunyai 17,943 *frames*, dengan rincian 2,058 *frames* yang terdeteksi dan 15,885 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 44 *frames*. Gambar 4.6 merupakan *data testing* 2-1.



Gambar 4.7: *Data testing* 2-2.

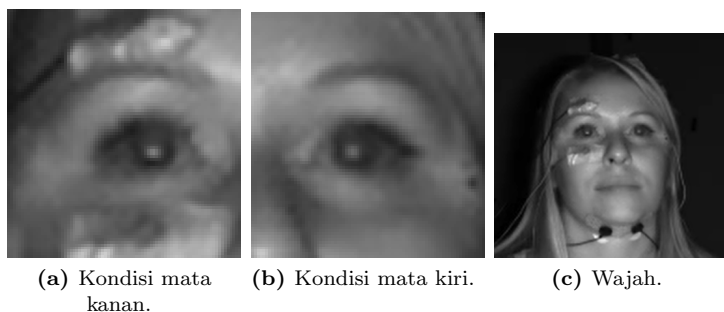
Pengujian kelima menggunakan video *testing* 2-2. Dalam vi-

deo *testing* 2-2 mempunyai 9,402 *frames*, dengan rincian 46 *frames* yang terdeteksi dan 9,356 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.7 merupakan *data testing* 2-2.



Gambar 4.8: *Data testing* 2-3.

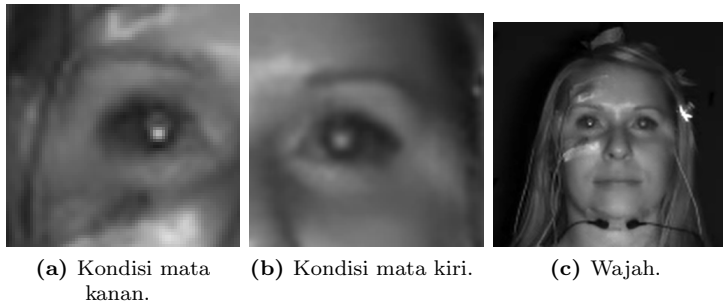
Pengujian keenam menggunakan video *testing* 2-3. Dalam video *testing* 2-3 mempunyai 8,166 *frames*, dengan rincian 942 *frames* yang terdeteksi dan 7,224 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 20 *frames*. Gambar 4.8 merupakan *data testing* 2-3.



Gambar 4.9: *Data testing* 3-1.

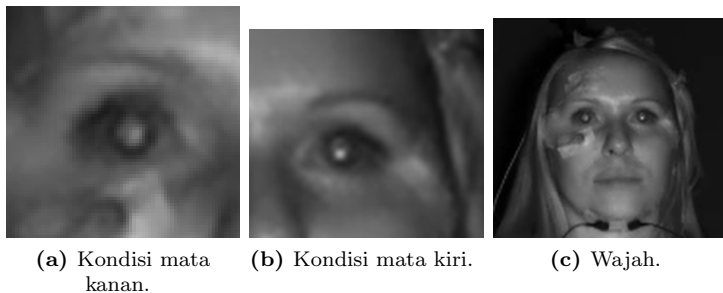
Pengujian ketujuh menggunakan video *testing* 3-1. Dalam video *testing* 3-1 mempunyai 18,218 *frames*, dengan rincian 5,685 *fra-*

mes yang terdeteksi dan 12,533 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 336 *frames*. Gambar 4.9 merupakan *data testing* 3-1.



Gambar 4.10: *Data testing* 3-2.

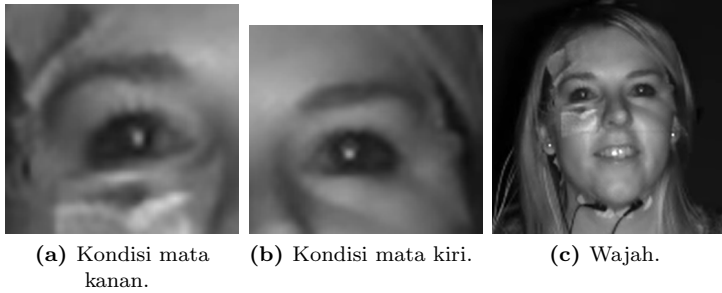
Pengujian kedelapan menggunakan video *testing* 3-2. Dalam video *testing* 3-2 mempunyai 8,859 *frames*, dengan rincian 1,011 *frames* yang terdeteksi dan 7,848 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 11 *frames*. Gambar 4.10 merupakan *data testing* 3-2.



Gambar 4.11: *Data testing* 3-3.

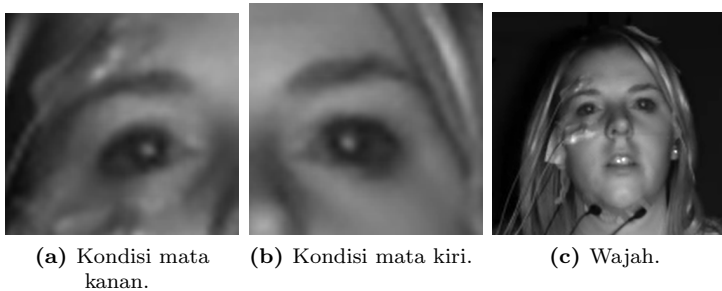
Pengujian kesembilan menggunakan video *testing* 3-3. Dalam video *testing* 3-3 mempunyai 8,878 *frames*, dengan rincian 518 *frames* yang terdeteksi dan 8,360 *frames* yang tidak dapat terdeteksi.

Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 4 *frames*. Gambar 4.11 merupakan *data testing* 3-3.



Gambar 4.12: *Data testing* 4-1.

Pengujian kesepuluh menggunakan video *testing* 4-1. Dalam video *testing* 4-1 mempunyai 17,841 *frames*, dengan rincian 12,538 *frames* yang terdeteksi dan 5,303 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 52 *frames*. Gambar 4.12 merupakan *data testing* 4-1.

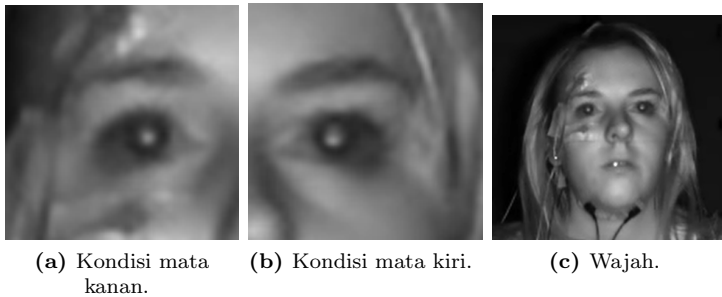


Gambar 4.13: *Data testing* 4-2.

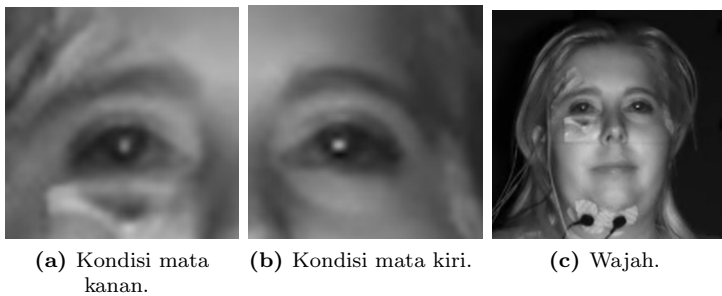
Pengujian kesebelas menggunakan video *testing* 4-2. Dalam video *testing* 4-2 mempunyai 8,793 *frames*, dengan rincian 4,509 *frames* yang terdeteksi dan 4,284 *frames* yang tidak dapat terde-

teksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 330 *frames*. Gambar 4.13 merupakan *data testing* 4-2.

Pengujian kedua belas menggunakan video *testing* 4-3. Dalam video *testing* 4-3 mempunyai 11,276 *frames*, dengan rincian 8,058 *frames* yang terdeteksi dan 3,218 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 2,387 *frames*. Gambar 4.14 merupakan *data testing* 4-3.



Gambar 4.14: *Data testing* 4-3.

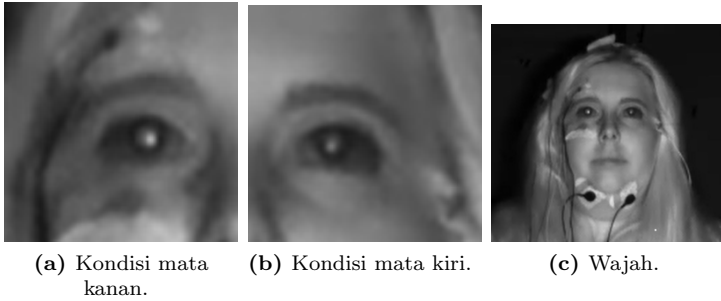


Gambar 4.15: *Data testing* 5-1.

Pengujian ketiga belas menggunakan video *testing* 5-1. Dalam video *testing* 5-1 mempunyai 18,267 *frames*, dengan rincian 12,112 *frames* yang terdeteksi dan 6,155 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu

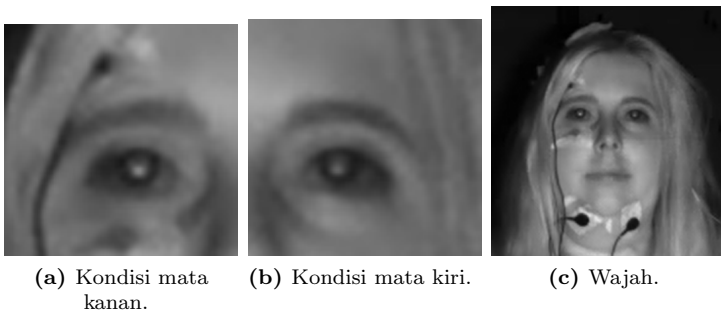
berjumlah 369 *frames*. Gambar 4.15 merupakan *data testing* 5-1.

Pengujian keempat belas menggunakan video *testing* 5-2. Dalam video *testing* 5-2 mempunyai 9,802 *frames*, dengan rincian 8,318 *frames* yang terdeteksi dan 1,484 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1,627 *frames*. Gambar 4.16 merupakan *data testing* 5-2.



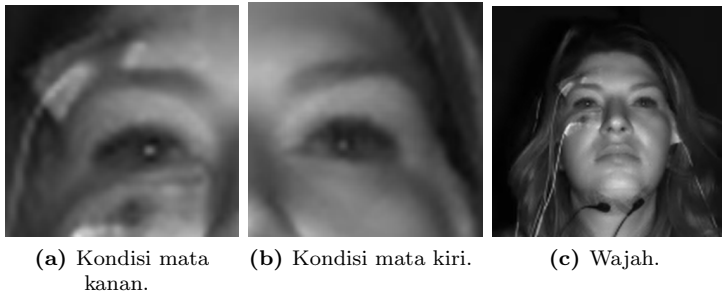
Gambar 4.16: *Data testing* 5-2.

Pengujian kelima belas menggunakan video *testing* 5-3. Dalam video *testing* 5-3 mempunyai 13,346 *frames*, dengan rincian 12,835 *frames* yang terdeteksi dan 511 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 5,116 *frames*. Gambar 4.17 merupakan *data testing* 5-3.



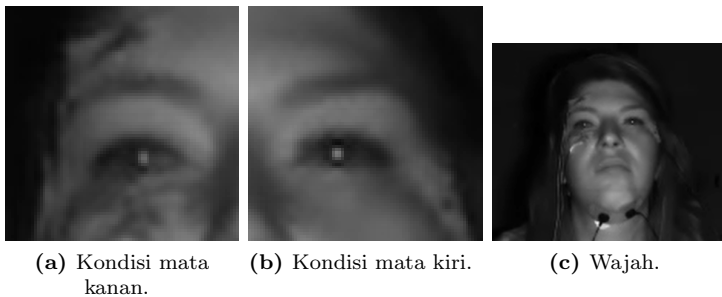
Gambar 4.17: *Data testing* 5-3.

Pengujian keenam belas menggunakan video *testing* 6-1. Dalam video *testing* 6-1 mempunyai 17,913 *frames*, dengan rincian 4,406 *frames* yang terdeteksi dan 13,507 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 35 *frames*. Gambar 4.18 merupakan *data testing* 6-1.



Gambar 4.18: *Data testing* 6-1.

Pengujian ketujuh belas menggunakan video *testing* 6-2. Dalam video *testing* 6-2 mempunyai 8,854 *frames*, dengan rincian 298 *frames* yang terdeteksi dan 8,556 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 2 *frames*. Gambar 4.19 merupakan *data testing* 6-2.



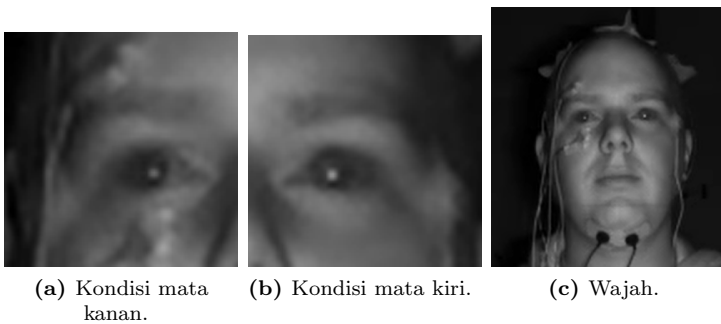
Gambar 4.19: *Data testing* 6-2.

Pengujian kedelapan belas menggunakan video *testing* 6-3. Dalam video *testing* 6-3 mempunyai 8,958 *frames*, dengan rincian 41 *frames* yang terdeteksi dan 8,917 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.20 merupakan *data testing* 6-3.



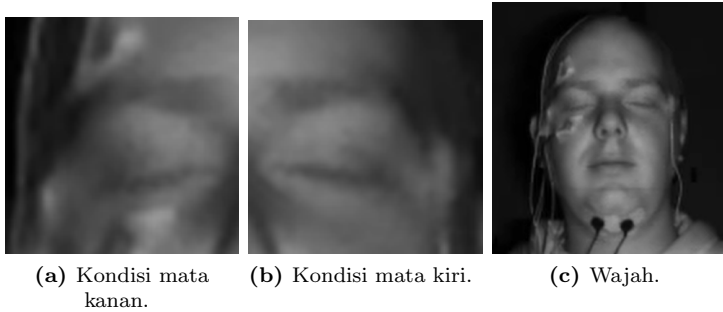
Gambar 4.20: *Data testing* 6-3.

Pengujian kesembilan belas menggunakan video *testing* 7-2. Dalam video *testing* 7-2 mempunyai 7,948 *frames*, dengan rincian 1,260 *frames* yang terdeteksi dan 6,688 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 36 *frames*. Gambar 4.21 merupakan *data testing* 7-2.



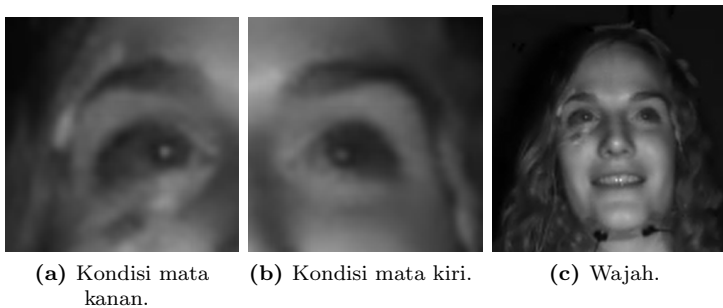
Gambar 4.21: *Data testing* 7-2.

Pengujian kedua puluh menggunakan video *testing* 7-3. Dalam video *testing* 7-3 mempunyai 8,879 *frames*, dengan rincian 155 *frames* yang terdeteksi dan 8,724 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.22 merupakan *data testing* 7-3.



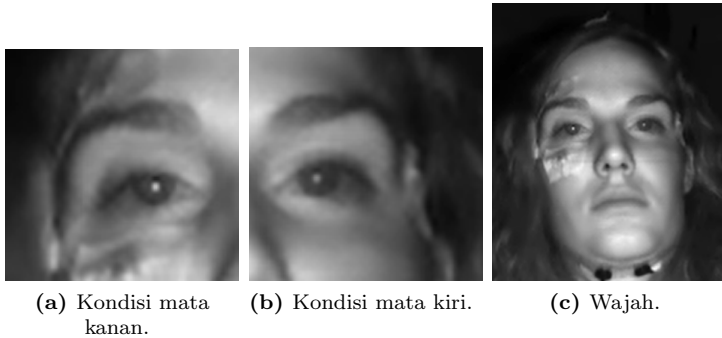
Gambar 4.22: *Data testing* 7-3.

Pengujian kedua puluh satu menggunakan video *testing* 8-1. Dalam video *testing* 8-1 mempunyai 18,120 *frames*, dengan rincian 9,326 *frames* yang terdeteksi dan 8,794 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 257 *frames*. Gambar 4.23 merupakan *data testing* 8-1.



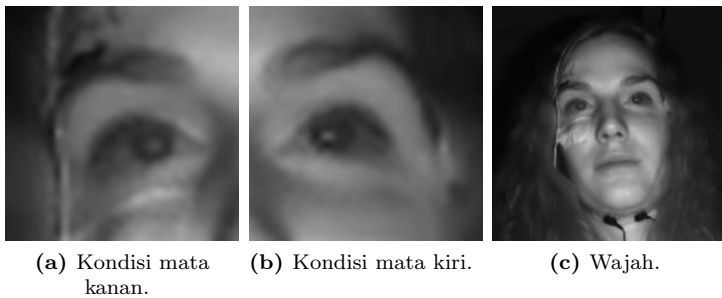
Gambar 4.23: *Data testing* 8-1.

Pengujian kedua puluh dua menggunakan video *testing* 8-2. Dalam video *testing* 8-2 mempunyai 8,920 *frames*, dengan rincian 2,249 *frames* yang terdeteksi dan 6,671 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 46 *frames*. Gambar 4.24 merupakan *data testing* 8-2.



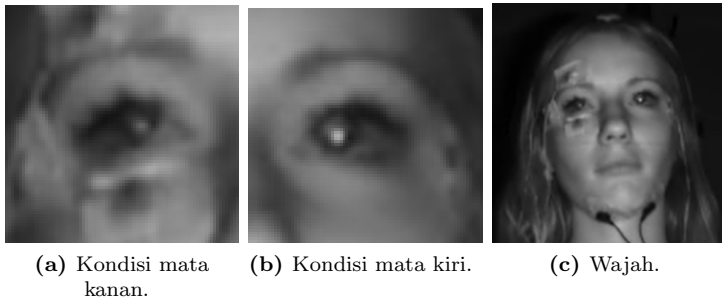
Gambar 4.24: *Data testing* 8-2.

Pengujian kedua puluh tiga menggunakan video *testing* 8-3. Dalam video *testing* 8-3 mempunyai 9,124 *frames*, dengan rincian 1,803 *frames* yang terdeteksi dan 7,321 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 99 *frames*. Gambar 4.25 merupakan *data testing* 8-3.



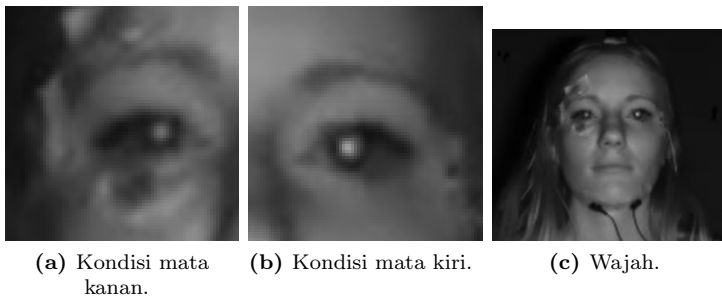
Gambar 4.25: *Data testing* 8-3.

Pengujian kedua puluh empat menggunakan video *testing* 9-2. Dalam video *testing* 9-2 mempunyai 18,227 *frames*, dengan rincian 7,275 *frames* yang terdeteksi dan 10,952 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 319 *frames*. Gambar 4.26 merupakan *data testing* 9-2.



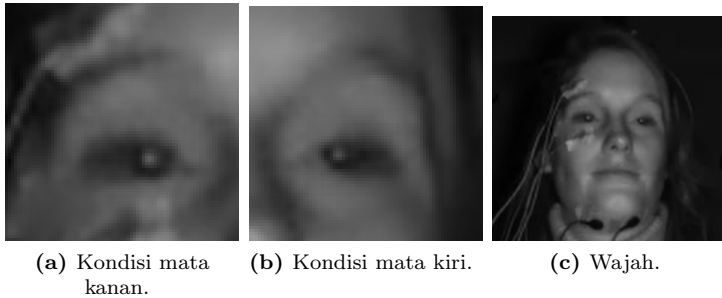
Gambar 4.26: *Data testing* 9-2.

Pengujian kedua puluh lima menggunakan video *testing* 9-3. Dalam video *testing* 9-3 mempunyai 18,159 *frames*, dengan rincian 5,247 *frames* yang terdeteksi dan 12,912 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 237 *frames*. Gambar 4.27 merupakan *data testing* 9-3.



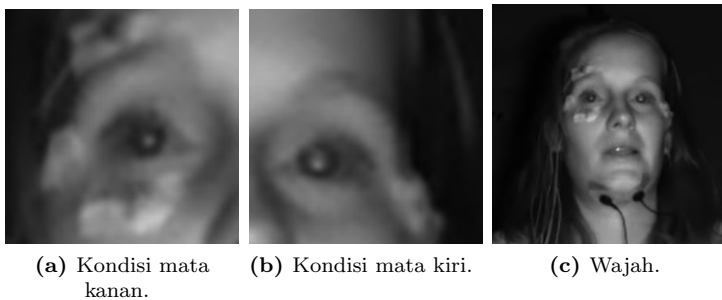
Gambar 4.27: *Data testing* 9-3.

Pengujian kedua puluh enam menggunakan video *testing* 10-1. Dalam video *testing* 10-1 mempunyai 17,867 *frames*, dengan rincian 70 *frames* yang terdeteksi dan 17,797 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.28 merupakan *data testing* 10-1.



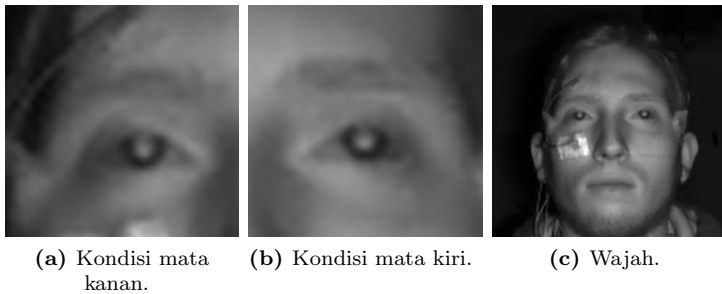
Gambar 4.28: *Data testing* 10-1.

Pengujian kedua puluh tujuh menggunakan video *testing* 10-3. Dalam video *testing* 10-3 mempunyai 17,899 *frames*, dengan rincian 1,072 *frames* yang terdeteksi dan 16,827 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 10 *frames*. Gambar 4.29 merupakan *data testing* 10-3.



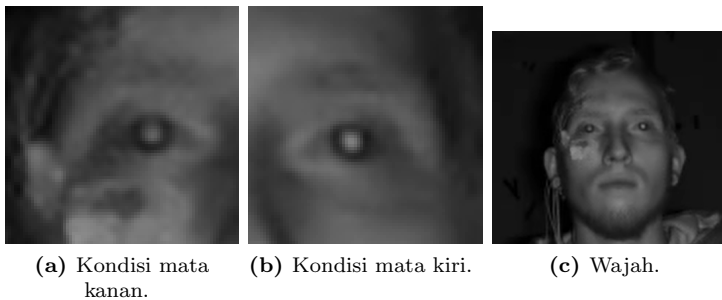
Gambar 4.29: *Data testing* 10-3.

Pengujian kedua puluh delapan menggunakan video *testing* 11-1. Dalam video *testing* 11-1 mempunyai 18,158 *frames*, dengan rincian 6,506 *frames* yang terdeteksi dan 11,652 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 244 *frames*. Gambar 4.30 merupakan *data testing* 11-1.



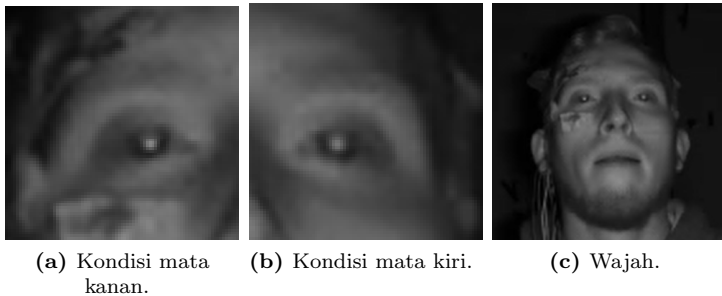
Gambar 4.30: *Data testing* 11-1.

Pengujian kedua puluh sembilan menggunakan video *testing* 11-2. Dalam video *testing* 11-2 mempunyai 17,886 *frames*, dengan rincian 1,009 *frames* yang terdeteksi dan 16,877 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 3 *frames*. Gambar 4.31 merupakan *data testing* 11-2.



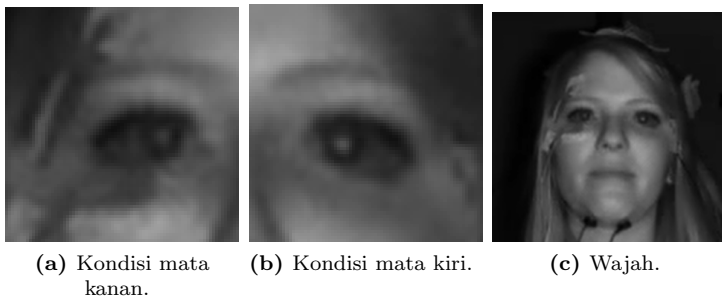
Gambar 4.31: *Data testing* 11-2.

Pengujian ketiga puluh menggunakan video *testing* 11-3. Dalam video *testing* 11-3 mempunyai 17,877 *frames*, dengan rincian 349 *frames* yang terdeteksi dan 17,528 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 2 *frames*. Gambar 4.32 merupakan *data testing* 11-3.



Gambar 4.32: *Data testing* 11-3.

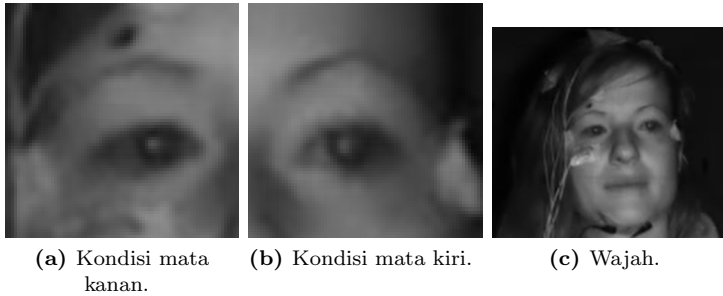
Pengujian tiga puluh satu menggunakan video *testing* 12-1. Dalam video *testing* 12-1 mempunyai 17,900 *frames*, dengan rincian 9,987 *frames* yang terdeteksi dan 7,913 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.33 merupakan *data testing* 12-1.



Gambar 4.33: *Data testing* 12-1.

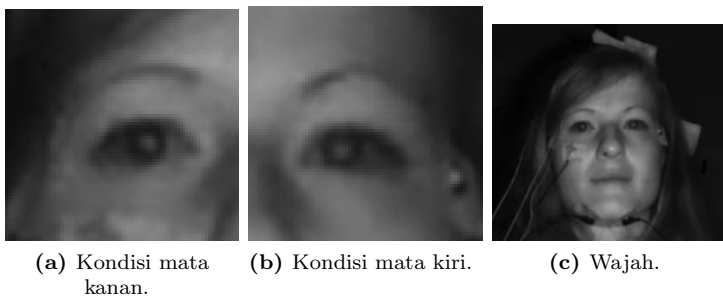
Pengujian tiga puluh dua menggunakan video *testing* 13-1. Da-

lam video *testing* 13-1 mempunyai 17,934 *frames*, dengan rincian 2,078 *frames* yang terdeteksi dan 15,856 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 32 *frames*. Gambar 4.34 merupakan *data testing* 13-1.



Gambar 4.34: *Data testing* 13-1.

Pengujian tiga puluh tiga menggunakan video *testing* 13-2. Dalam video *testing* 13-2 mempunyai 18,632 *frames*, dengan rincian 6,664 *frames* yang terdeteksi dan 11,968 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 732 *frames*. Gambar 4.35 merupakan *data testing* 13-2.



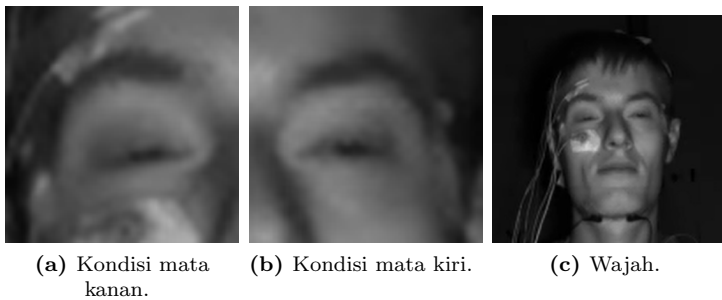
Gambar 4.35: *Data testing* 13-2.

Pengujian tiga puluh empat menggunakan video *testing* 14-1. Dalam video *testing* 14-1 mempunyai 17,885 *frames*, dengan rincian 2,614 *frames* yang terdeteksi dan 15,271 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 2 *frames*. Gambar 4.36 merupakan *data testing* 14-1.



Gambar 4.36: *Data testing* 14-1.

Pengujian tiga puluh lima menggunakan video *testing* 14-2. Dalam video *testing* 14-2 mempunyai 17,862 *frames*, dengan rincian 1,149 *frames* yang terdeteksi dan 16,713 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 1 *frames*. Gambar 4.37 merupakan *data testing* 14-2.



Gambar 4.37: *Data testing* 14-2.

Pengujian tiga puluh enam menggunakan video *testing* 14-3. Dalam video *testing* 14-3 mempunyai 17,970 *frames*, dengan rincian 1,102 *frames* yang terdeteksi dan 16,868 *frames* yang tidak dapat terdeteksi. Adapun beberapa *frames* tambahan pada *frames* tertentu berjumlah 52 *frames*. Gambar 4.38 merupakan *data testing* 14-3.



(a) Kondisi mata kanan.

(b) Kondisi mata kiri.

(c) Wajah.

Gambar 4.38: *Data testing* 14-3.

Halaman ini sengaja dikosongkan.

BAB 5

PENUTUP

5.1 Kesimpulan

Dari pelaksanaan dan pengujian sistem yang sudah dilakukan, penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Model dapat mengklasifikasikan kondisi mata terbuka dan mata tertutup dengan menggunakan *Convolutional Neural Network* dengan arsitektur yang memiliki 3 layer mendapatkan hasil *accuracy* 98% dan *error* 2.05%.
2. Model dari hasil *training* dapat berjalan secara *real-time*.
3. Pendeteksian secara *real-time* memiliki kendala bila wajah terhalangi oleh objek lain.

5.2 Saran

Demi pengembangan penelitian lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut :

1. Menambah jumlah *dataset* kondisi mata kiri dan mata kanan dari internet maupun dokumentasi sendiri.
2. Pengembangan dari tugas akhir ini dapat dikembangkan untuk penelitian pendeteksian kantuk atau *drowsiness* untuk pengemudi.
3. Untuk melakukan pengujian setidaknya memiliki pencahayaan yang cukup terang agar dapat mendeteksi wajah dan mata secara baik.

Halaman ini sengaja dikosongkan.

DAFTAR PUSTAKA

- [1] N. Shofia, “Convolutional neural network,” June 2018. (Dikutip pada halaman xix, 5).
- [2] N. L. Fitriyani, C.-K. Yang, and M. Syafrudin, “Real-time eye state detection system using haar cascade classifier and circular hough transform,” in 2016 IEEE 5th Global Conference on Consumer Electronics, pp. 1–3, IEEE, 2016. (Dikutip pada halaman xix, 9, 10).
- [3] W. Zhang, B. Cheng, and Y. Lin, “Driver drowsiness recognition based on computer vision technology,” Tsinghua Science and Technology, vol. 17, no. 3, pp. 354–362, 2012. (Dikutip pada halaman 1).
- [4] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in Proceedings of 2010 IEEE international symposium on circuits and systems, pp. 253–256, IEEE, 2010. (Dikutip pada halaman 5).
- [5] S. Abidin, “Deteksi wajah menggunakan metode haar cascade classifier berbasis webcam pada matlab,” Jurnal Teknologi Elektroika, vol. 15, no. 1, pp. 21–27, 2018. (Dikutip pada halaman 8).
- [6] M. Rouse, “Definition face detection,” February 2020. (Dikutip pada halaman 9).
- [7] I. S. Nugraha, S. S. Muljono, and M. Kom, “Aplikasi android deteksi mata menggunakan metode viola-jones,” Tugas Akhir Mahasiswa Teknik Informatika Universitas Dian Nuswantoro Semarang, 2015. (Dikutip pada halaman 9).
- [8] C. B. S. Maior, M. J. das Chagas Moura, J. M. M. Santana, and I. D. Lins, “Real-time classification for autonomous drowsiness detection using eye aspect ratio,” Expert Systems with Applications, p. 113505, 2020. (Dikutip pada halaman 10).
- [9] T. Pradhan, A. N. Bagaria, and A. Routray, “Measurement of perclos using eigen-eyes,” in 2012 4th International Conference

on Intelligent Human Computer Interaction (IHCI), pp. 1–4, IEEE, 2012. (Dikutip pada halaman 11).

- [10] yoshidan, “pytorch eyblink detection,” November 2019. (Dikutip pada halaman 14, 16).

BIOGRAFI PENULIS



Lutfi Ananditya Septiandi, atau yang lebih akrab dipanggil Lutfi, lahir di Surabaya pada 30 September 1998. Anak pertama dari tiga bersaudara ini telah menyelesaikan pendidikan di SMA Muhammadiyah 2 Kota Surabaya pada 2016 dan melanjutkannya di Departemen Teknik Komputer FTEIC-ITS. Selama menjadi mahasiswa ITS, penulis tertarik dengan bidang *Internet of Things (IoT)* dan *Image Processing*. Selain itu penulis juga aktif di bidang pengembangan diri melalui berbagai kegiatan dan organisasi seperti organisasi UKM CLICK ITS, *team event* HIMATEKTRO, kegiatan kepanitiaan rutin seperti *Multimedia & Game Event 4*, GERIGI 2017, dan lainnya.

Halaman ini sengaja dikosongkan.