

**DETEKSI PERGERAKAN MATA DAN KEDIPAN UNTUK
MEMILIH EMPAT MENU DISPLAY MENGGUNAKAN
PROBABILITAS BERDASARKAN**

FACIAL LANDMARK

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

Akbar Dicky Purwanto

NIM: 165150307111042



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

Deteksi Pergerakan Mata dan Kedipan Untuk Memilih Menu Display
Menggunakan Probabilitas Berdasarkan *Facial landmark*

Skripsi

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

Akbar Dicky Purwanto

NIM : 165150307111042

Skripsi ini telah diuji dan dinyatakan lulus pada
9 Januari 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing Tunggal



Fitri Utaminigrum, Dr. Eng., S.T., M.T.

NIP: 19820710 200812 2 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoro Kurniawan, S.T., M.T., Ph.D.

NIP: 197105182003121001

A

PERTANYAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Desember 2019



Akbar Dicky Purwanto

NIM: 165150307111042

ABSTRAK

Akbar Dicky Purwanto, Deteksi Pergerakan Mata dan Kedipan Untuk Memilih Menu Display Menggunakan Probabilitas Berdasarkan *Facial landmark*

Pembimbing: Fitri Utaminingrum, Dr. Eng., S.T., M.T

Penderita disabilitas seperti pada orang stroke kesulitan dalam menjalani aktivitas sehari – hari karena terdapat beberapa orang tubuhnya yang tidak mampu digerakkan. Namun pada penderita disabilitas, pergerakan mata masih dapat dikontrol oleh tubuh dengan baik seperti melihat ke kiri, ke kanan, ke depan, ke atas dan berkedip. Suatu metode digunakan untuk mengatasi masalah tersebut dengan menggabungkan beberapa metode lainnya. Metode *facial landmark* digunakan untuk mendapatkan data wajah dari seseorang seperti letak matanya. Untuk mendeteksi pergerakan mata dapat diprediksi menggunakan metode binary maupun metode garis vertikal dan horizontal yang terdapat pada *facial landmark*. Pendeteksian dibedakan menjadi lima yaitu deteksi mata melihat ke kanan, kiri, atas, bawah dan depan. Pengujian dilakukan pada latar belakang yang sepi dan ramai menggunakan pencahayaan terang dan redup serta jarak yang ditentukan yaitu 30cm, 40cm dan 50cm. Hasil dari metode di atas adalah pada pencahayaan terang didapatkan akurasi 91% dan pada pencahayaan redup didapatkan akurasi 87%, pada jarak 30cm didapatkan akurasi 90%, pada jarak 40cm didapatkan akurasi 91% dan pada jarak 50cm didapatkan akurasi 87%. Metode ini diterapkan pada pemilihan menu yang ditampilkan pada layar LCD. Menu tersebut berisi berbagai kebutuhan penderita disabilitas seperti memanggil perawat, ingin buang air dan memilih menu makanan. Dalam pengaplikasiannya akurasi integrasi metode dengan menu display mendapatkan akurasi sebesar 87,5% dan dapat bekerja dengan cukup baik.

Kata kunci: Deteksi Pergerakan Mata, Deteksi Kedipan Mata, *Facial landmark*, Probabilitas, Menu Display

ABSTRACT

Akbar Dicky Purwanto, Eye Movement and Blink Detection For Selecting Menu Display Using Probability Based on Facial Landmark

Supervisors: Fitri Utaminingrum, Dr. Eng., S.T., M.T

People with disabilities such as stroke have difficulty in carrying out daily activities because there are some people are unable to be move their limbs. But in people with disabilities, the eye movement can still eye movements can still move well as looking left, right, forward, up and blinking. A method is used to overcome this problem by combining several other methods. The *facial landmark* method is used to get facial data from a person such as the location of his eyes. To detect eye movements can be predicted using the binary method and the vertical and horizontal line methods on *facial landmarks*. Detection detection can be categorized into five, namely the detection of eyes when looking right, left, up, down and front. The experiment was carried out on a flat and crowded background using bright and dim lighting and the specified distances between camera and object are 30cm, 40cm and 50cm. The results of the method are 91% accuracy in bright lightning and 87% accuracy in dim lighting, 90% accuracy is obtained at 30cm distance, 91% accuracy is obtained at 40cm distance and 87% accuracy is obtained at 50cm distance. This method is applied to the menu selection displayed on the LCD screen. The menu contains various needs of people with disabilities such as calling a nurse, want to go to the toilet and choosing a food menu. In its application the accuracy of the integration of the method with the display menu gets an accuracy of 87.5% and can work well.

Keyword: Eye Movement Detection, Eye Blinking Detection, *Facial landmark*, Probabilitas, Menu Display

DAFTAR ISI

DETEKSI PERGERAKAN MATA DAN KEDIPAN UNTUK MEMILIH EMPAT MENU DISPLAY MENGGUNAKAN PROBABILITAS BERDASARKAN FACIAL LANDMARK	i
PENGESAHAN	ii
PERTANYAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Landasan Teori	6
2.2.1 Pengolahan Citra Digital	6
2.2.2 Stroke	7
2.2.3 OpenCV	8
2.2.4 Resize	8
2.2.5 Crop Image	8
2.2.6 Dlib	9
2.2.7 Facial landmark	9
2.2.8 Konversi RGB ke Biner	10
2.2.9 Probabilitas	11

2.2.10 Perhitungan Probabilitas Berdasarkan biner mata.....	11
2.2.11 Perhitungan Probabilitas Berdasarkan panjang garis vertikal dan horizontal	11
2.2.12 User Interface Menu Display	12
BAB 3 METODOLOGI	13
3.1 Tipe Penelitian	13
3.2 Strategi dan Rancangan Penelitian	13
3.2.1 Metode Secara Umum	13
3.2.2 Subyek Penelitian	14
3.2.3 Lokasi Penelitian.....	14
3.2.4 Teknik Pengumpulan Data	15
3.2.5 Teknik Analisis Data dan Pembahasan.....	15
3.2.6 Peralatan Pendukung Yang Digunakan	15
BAB 4 REKAYASA KEBUTUHAN.....	16
4.1 Gambaran Umum Sistem.....	16
4.2 Analisis Kebutuhan Sistem.....	16
4.2.1 Kebutuhan Sistem	16
4.3 Kebutuhan Fungsional	23
BAB 5 PERANCANGAN DAN IMPLEMENTASI	25
5.1 Perancangan Sistem.....	25
5.1.1 Perancangan Perangkat Keras	25
5.1.2 Perancangan Perangkat Lunak.....	27
5.2 Implementasi	40
5.2.1 Implementasi Perangkat Keras	40
5.2.2 Implementasi Perangkat Lunak.....	41
BAB 6 Pengujian dan analisis	55
6.1 Pengujian Keakurasian Dalam Pendeteksian <i>Facial landmark</i> Pada Kondisi Latar Belakang Ramai Dan Sepi.....	55
6.1.1 Tujuan Pengujian.....	55
6.1.2 Prosedur Pengujian	55
6.1.3 Pelaksanaan Pengujian.....	55
6.1.4 Hasil dan Analisis Pengujian.....	55

6.2 Pengujian Keakurasian Dalam Pendeteksian Pergerakan Mata Pada Pencahayaian Terang (lux 50-400) dan Redup (lux 1-49).....	58
6.2.1 Tujuan Pengujian.....	58
6.2.2 Prosedur Pengujian.....	58
6.2.3 Pelaksanaan Pengujian.....	58
6.2.4 Hasil dan Analisis Pengujian.....	58
6.3 Pengujian Keakurasian Dalam Pendeteksian Pergerakan Mata Pada Jarak 30cm, 40cm dan 50cm antara wajah dengan kamera	61
6.3.1 Tujuan Pengujian.....	61
6.3.2 Prosedur Pengujian	61
6.3.3 Pelaksanaan Pengujian.....	61
6.3.4 Hasil dan Analisis Pengujian.....	62
6.4 Pengujian Kinerja Integrasi Pergerakan Mata Yang Telah Diterapkan Pada Pemilihan Menu Pada Layar LCD	64
6.4.1 Tujuan Pengujian.....	64
6.4.2 Prosedur Pengujian	64
6.4.3 Pelaksanaan Pengujian.....	64
6.4.4 Hasil dan Analisis Pengujian.....	64
BAB 7 Kesimpulan dan saran.....	70
7.1 Kesimpulan.....	70
7.2 Saran	70
Daftar referensi	72
LAMPIRAN A TABEL.....	75
LAMPIRAN B GAMBAR	90

DAFTAR TABEL

Tabel 2.1 Hasil Analisis Kajian Pustaka.....	4
Tabel 4.1 Spesifikasi Raspberry Pi 3 B+.....	16
Tabel 4.2 Spesifikasi Webcam Logitech c170.....	18
Tabel 4.3 Spesifikasi 3.5Inch HDMI LCD.....	18
Tabel 4.4 Spesifikasi Sd Card Micro SDHC.....	19
Tabel 4.5 Spesifikasi Light Meter HS1010.....	20
Tabel 4.6 Spesifikasi HP Pav Notebook 14-ab134tx.....	22
Tabel 5.1 Konfigurasi Pin Raspberry Pi.....	26
Tabel 5.2 Data Latih Deteksi Pergerakan Mata.....	31
Tabel 5.3 Pengujian 1 Deteksi Pergerakan Mata.....	32
Tabel 5.4 Pengujian 2 Deteksi Pergerakan Mata.....	32
Tabel 5.5 Pengujian 3 Deteksi Pergerakan Mata.....	32
Tabel 5.6 Nilai Deteksi Mata.....	32
Tabel 5.7 Data Latih Deteksi Kedipan Mata.....	35
Tabel 5.8 Pengujian 1 Deteksi Kedipan Mata.....	36
Tabel 5.9 Pengujian 2 Deteksi Kedipan Mata.....	36
Tabel 5.10 Pengujian 3 Deteksi Kedipan Mata.....	36
Tabel 5.11 Nilai Kedipan Mata.....	36
Tabel 5.12 Keterangan <i>State Machine</i> Pemilihan Menu.....	38
Tabel 5.13 Kode Program Library Yang Digunakan.....	42
Tabel 5.14 Kode Program Sistem Utama.....	42
Tabel 5.15 Kode Program Implementasi Deteksi Pergerakan Mata.....	43
Tabel 5.16 Kode Program Implementasi Deteksi Kedipan Mata.....	46
Tabel 5.17 Kode Program Implementasi Pemilihan menu.....	48
Tabel 6.1 Pengujian Deteksi Facial Landmark Pada Latar Belakang Ramai Dan Sepi.....	56
Tabel 6.2 Akurasi Pengujian Terhadap Cahaya Terang dan Redup.....	59
Tabel 6.3 Akurasi Pengujian Terhadap Jarak 30cm, 40cm dan 50cm.....	62
Tabel 6.4 Hasil Percobaan Integrasi Menu Display.....	69

DAFTAR GAMBAR

Gambar 2.1 Pengolahan Citra Digital Segmentasi	7
Gambar 2.2 Penderita Stroke	8
Gambar 2.3 Cropping Gambar	9
Gambar 2.4 Face Landmark	10
Gambar 2.5 Konversi RGB ke GrayScale	10
Gambar 2.6 Gambar Biner	11
Gambar 2.7 Menu Display Pada Makanan	12
Gambar 3.1 Skema Metodologi Penelitian	14
Gambar 4.1 Raspberry Pi 3 B+	17
Gambar 4.2 Webcam Logitech c170	18
Gambar 4.3 3.5Inch HDMI LCD	19
Gambar 4.4 Sd Card Micro SDHC	19
Gambar 4.5 Power Supply	20
Gambar 4.6 Light Meter HS1010	21
Gambar 4.7 Speaker	21
Gambar 4.8 HP Pav Notebook 14-ab134tx	22
Gambar 5.1 Diagram Block Sistem	25
Gambar 5.2 Ilustrasi Peletakan Alat	26
Gambar 5.3 Perancangan Pemasangan Komponen	27
Gambar 5.4 Diagram Block Penggunaan Library	28
Gambar 5.5 Flowchart Sistem Utama	29
Gambar 5.6 Flowchart Deteksi Pergerakan Mata	33
Gambar 5.7 Face Landmark Mata	34
Gambar 5.8 Flowchart Deteksi Kedipan Mata	37
Gambar 5.9 State Machine Pemilihan Menu	38
Gambar 5.10 Implementasi Alat	41
Gambar 5.11 Implementasi Pemasangan Komponen	41
Gambar 6.1 Deteksi <i>Facial landmark</i> . (a) Pada Latar Belakang Sepi. (b) Pada Latar Belakang Ramai	57

Gambar 6.2 Pengujian Pada Pencahayaan Terang dan Redup. (a) Pengujian Pada Pencahayaan Terang. (b) Pengujian Pada Pencahayaan Gelap	60
Gambar 6.3 Nilai Deteksi.....	61
Gambar 6.4 Pengujian Deteksi Pada Jarak 30cm dan 50cm. (a) Pengujian Pada Jarak 30cm. (b) Pengujian Pada Jarak 40cm. (c) Pengujian Pada Jarak 50cm.	63
Gambar 6.5 Menu Dalam Keadaan Idle	65
Gambar 6.6 Tampilan Menu Utama	65
Gambar 6.7 Memilih Menu Perawat	66
Gambar 6.8 Mengganti Pilihan Menu	66
Gambar 6.9 Tampilan Menu Makanan	67
Gambar 6.10 Memilih Menu Toilet.....	67
Gambar 6.11 Memindah Menu ke Exit	68
Gambar 6.12 Memilih Menu Exit	68



DAFTAR LAMPIRAN

LAMPIRAN A TABEL	75
A.1 Tabel Data Uji Pada Subjek 1 hingga 15	75
LAMPIRAN B GAMBAR	90
B.1 Gambar Hasil Deteksi <i>Facial landmark</i>	90
B.2 Gambar Hasil Deteksi Pergerakan Mata	91



BAB 1 PENDAHULUAN

1.1 Latar belakang

Stroke menjadi penyebab kematian terbanyak didunia setelah penyakit jantung dan menjadi penyebab utama dari disabilitas. Jumlah penderita stroke didunia mencapai 33 juta, dengan 16,9 juta terkena stroke serangan pertama pada tahun 2010 (Putri, Islam dan Subadi 2018). Penderita stroke mengalami keterbatasan maupun gangguan dalam beraktivitas fisik yaitu dikarenakan melemahnya fungsi motorik yang berakibat pada kelemahan menggerakkan kaki, kelemahan menggerakkan tangan, kelemahan untuk bangun dari tempat tidur, kelemahan untuk duduk, ketidakmampuan bicara, dan tidak mampu menggerakkan fungsi motorik lainnya (Cadilhac, et al. 2019). Pada penderita stroke akut, penderita tidak dapat melakukan gerakan tubuh selain bola mata, akan tetapi perawat tidak bisa menjaga terus menerus selama 24 jam, namun pada kenyataannya penderita stroke sangat membutuhkan perhatian khusus. Oleh karena itu diperlukan suatu solusi untuk mengatasi masalah tersebut.

Pendeteksian pergerakan mata membutuhkan suatu metode agar pergerakan mata tersebut dapat dideteksi. Pada penelitian sebelumnya pendeteksian pergerakan mata pada orang disabilitas menggunakan metode haar cascade yang digabungkan dengan beberapa metode lainnya. Penelitian yang berjudul "Eye Movement as Navigator for Disabled Person" membuktikan bahwa pendeteksian pergerakan mata dapat menggunakan haar cascade sebagai metode untuk pendeteksian letak mata dan metode hough circle sebagai pendeteksian pergerakan bola mata. Namun penelitian tersebut memiliki akurasi 83% untuk mendeteksi pergerakan mata, hal tersebut dikarenakan haar cascade dapat mendeteksi benda lain selain wajah manusia dan mempunyai nilai akurasi rendah pada latar belakang yang bervariasi (Utaminigrum, et al. 2016). Untuk meningkatkan akurasi diperlukan adanya suatu penggabungan dari beberapa metode agar pendeteksian lebih sempurna.

Dari permasalahan di atas dibutuhkan suatu solusi untuk mengatasinya, *facial landmark* memiliki kelebihan untuk mendeteksi titik – titik pada wajah sehingga dapat menentukan suatu bentuk biologis dari wajah manusia (Vezzetti, et al. 2012). *Facial landmark* berisi berbagai macam gambar data latih dari wajah manusia sehingga sangat akurat dalam pendeteksiannya. Selain itu metode *facial landmark* dapat menentukan bagian – bagian dari wajah manusia seperti bagian mulut, hidung, alis dan mata.

Untuk memudahkan penyandang penderita stroke yang lumpuh total penelitian ditujukan untuk menciptakan suatu alat menggunakan metode *facial landmark* yang dapat mendeteksi area wajah. Pada penerapan metode *facial landmark* dengan kamera dipergunakan untuk mengambil data citra yang berfungsi sebagai sensor. Setelah data citra wajah didapatkan untuk mengambil area mata, dapat menggunakan metode *facial landmark* dan untuk mendeteksi pergerakan mata menggunakan segmentasi biner yang digunakan untuk memilih

menu pada suatu layar atau tampilan. Layar tersebut berisi menu yaitu ingin memanggil perawat untuk meminta bantuan, penderita stroke ingin buang air kecil atau air besar, pemilihan menu makanan yang terdapat di layar. Dengan menggunakan citra digital penyandang stroke yang terganggu dalam aktivitas kesehariannya dapat terbantu.

1.2 Rumusan masalah

Berdasarkan latar belakang dapat dirumuskan masalah penelitian sebagai berikut :

1. Berapakah akurasi deteksi *facial landmark* pada kondisi latar belakang yang sepi dan ramai ?
2. Berapakah akurasi deteksi pergerakan mata dalam pencahayaan yang terang (lux 50-150) dan redup (lux 1-49) ?
3. Berapakah akurasi deteksi pergerakan mata pada jarak 30cm, 40cm dan 50cm antara wajah dengan kamera ?
4. Bagaimana kinerja integrasi pergerakan mata yang diterapkan pada pemilihan menu pada layar LCD ?

1.3 Tujuan

1. Mengetahui akurasi deteksi *facial landmark* pada kondisi latar belakang yang sepi dan ramai.
2. Mengetahui akurasi deteksi pergerakan mata dalam pencahayaan yang terang (lux 50-150) dan redup (lux 1-49).
3. Mengetahui akurasi deteksi pergerakan mata pada jarak 30cm, 40cm dan 50cm antara wajah dengan kamera.
4. Mengetahui kinerja integrasi pergerakan mata yang diterapkan pada pemilihan menu pada layar LCD.

1.4 Manfaat

Membantu penyandang disabilitas penderita stroke dan cacat tangan dalam kesehariannya untuk memanggil perawat ketika dibutuhkan saat ingin memilih menu makanan, ingin memanggil perawat karena membutuhkan pertolongan dan ingin buang air dengan cara melakukan pergerakan mata untuk pemilihan menu pada layar LCD.

1.5 Batasan masalah

Batasan ruang lingkup penelitian mencakup :

1. Posisi kamera diletakkan tepat di depan pengguna dengan jarak 30cm, 40cm dan 50cm, kamera harus dapat menangkap seluruh bagian wajah.
2. Pengujian dilakukan pada intensitas cahaya terang terang (lux 50-400) dan redup (lux 1-49).

3. Resolusi gambar 400 x 300 pixel.

1.6 Sistematika pembahasan

Sistematika penulisan ini sebagai berikut:

BAB I : Pendahuluan

Bab ini membahas tentang latar belakang yang menjadi alasan untuk penulis menyusun topik penelitian, rumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan, dan jadwal pelaksanaan.

BAB II : Landasan Kepustakaan

Bab ini membahas mengenai teori-teori dan referensi yang digunakan dalam membantu penulisan dan menemukan jawaban atas rumusan masalah.

BAB III : Metodologi Penelitian

Bab ini menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan sistem, implementasi, pengujian serta pengambilan kesimpulan dan saran.

Bab IV : Rekayasa Kebutuhan

Bab ini berisi detail pernyataan masalah, identifikasi sistem, serta daftar terstruktur kebutuhan perangkat secara fungsional, data, dan non-fungsional.

Bab V : Perancangan dan Implementasi

Bab ini berisi perancangan dan implementasi dari sistem citra digital menggunakan PC.

Bab VI : Pengujian

Bab ini berisi strategi dan data pengujian, ringkasan hasil pengujian serta evaluasi hasil kerja sistem secara keseluruhan.

Bab VII : Penutup

Bab ini berisi ringkasan kesimpulan dari pencapaian sistem dan saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Tabel 2.1 Hasil Analisis Kajian Pustaka

No	Penelitian Sebelumnya	Persamaan	Kelebihan	Kekurangan
1	“Face Detection & Face Recognition Using Open Computer Vision Classfires” (Dinalankara 2017)	Deteksi area mata	Pendeteksian efisien karena hanya dipakai area mata saja	Menggunakan metode haar cascade yang akurasinya rendah
2	“Eye Movement as Navigator for Disabled Person” (Utaminingrum, et al. 2016)	Deteksi pergerakan mata	Pendeteksian pergerakan mata difokuskan pada bagian sclera, sclera tiap orang berwarna putih.	Menggunakan metode haar cascade yang akurasinya rendah
3	“Efficient Eye Blink Detection Method for disabledhelping domain” (Mohammed dan Anwer 2014)	Deteksi kedipan pada mata	Proses eksekusi yang efisien karena gambar diolah pada segmentasi biner	Menggunakan metode haar cascade yang akurasinya rendah
4	“Optimization of the Data Representation Integrated From in the Viola-Jones Algorithm for a Person's Face Search” (Alyushin, Alyushin dan Kolobashkina 2018)	Pendeteksian area wajah	Mengolah framework agar bisa diproses untuk dideteksi	Menggunakan metode Viola Jones yang rentan mendeteksi benda lain selain wajah

Pada penelitian yang dilakukan oleh Lahiru Dinalankara berjudul “Face Detection & Face Recognition Using Open Computer Vision Classfires” memuat bahwa sebuah sistem yang menggunakan citra digital sebagai metode utamanya dan menggunakan haar cascades untuk metode dalam pendeteksian lokasi wajah dengan menandainya menggunakan persegi empat. Metode LBPH(Local Binary Pattern Histogram) digunakan untuk memperkirakan wajah dari seseorang, namun tidak cocok digunakan pada segmentasi mata karena warna dari mata akan berubah. Setelah bagian wajah dapat dideteksi maka bagian tersebut akan di potong untuk mempercepat proses eksekusi pendeteksian wajah (Dinalankara 2017).

Pada penelitian lainnya yang dilakukan oleh Fitri Utaminingrum berjudul “Eye Movement as Navigator for Disabled Person” menjelaskan bahwa untuk mendeteksi lokasi mata dapat menggunakan metode haar cascade, lalu untuk mendeteksi pergerakan mata atau mendeteksi pergerakan pupil dapat

menggunakan metode hough circle transform. Dari kedua metode tersebut kita dapat mengetahui pergerakan mata ke kanan, ke kiri, ke atas, ke bawah, maupun ke depan. Pada penelitian ini berfokus pada bagian sclera untuk dideteksi, memiliki kelebihan yaitu tidak semua orang memiliki pupil berwarna hitam namun semua orang memiliki sclera yang berwarna putih, hal tersebut dapat meningkatkan akurasi dalam pendeteksian pergerakan mata. Pada penelitian ini memuat akan haar cascade yang dapat mendeteksi benda lain selain mata, untuk itu diperlukan adanya metode pendeteksian baru yang lebih akurat (Utaminingrum, et al. 2016).

Pada penelitian yang dilakukan oleh Mohammed yang berjudul "Efficient EyeBlink Detection Method for disabledhelping domain" memuat mengenai pendeteksian kedipan mata yang menggunakan metode haar cascade. Metode pada penelitian ini ditujukan kepada orang disabilitas yang hanya bisa menggerakkan kedipan matanya saja. Citra yang semula berwarna RGB akan diubah ke bentuk biner untuk mempercepat proses eksekusi, setelah itu citra akan dideteksi berdasarkan pixel hitam pada bagian mata.

Penelitian yang berjudul "Optimization of the Data Representation Integrated From in the Viola-Jones Algorithm for a Person's Face Search" memuat mengenai pendeteksian area wajah untuk mengenali wajah tersebut agar bisa dilakukan proses deteksi. Pendeteksian wajah dilakukan menggunakan metode viola jones untuk mendapatkan framework dari wajah. Setelah dilakukan pendeteksian area wajah maka framework tersebut diolah menggunakan beberapa metode agar proses pengenalan wajah dapat dideteksi dengan baik. Namun pada metode viola jones memiliki kelemahan yaitu pendeteksian kurang akurat ketika gambar yang diproses bukan hanya area wajah.

Dari keempat penelitian di atas seperti pada Tabel 2.1 dapat diketahui bahwa untuk pendeteksian bagian wajah dan mata dapat menggunakan haar cascade dan viola jones namun haar cascade dan viola jones juga dapat mendeteksi benda lain selain wajah. Untuk itu metode *facial landmark* digunakan karena pendeteksian wajah manusia menggunakan *facial landmark* tidak dapat mendeteksi benda lain. Pada penelitian pertama dapat kita ambil kesimpulan bahwa untuk mendeteksi bagian mata, hanya perlu mengambil area mata saja, tidak perlu memproses seluruh wajah untuk mempercepat proses eksekusi program. Pada penelitian kedua dapat kita ambil kesimpulan bahwa pendeteksian pergerakan mata dapat kita deteksi pada bagian sclera saja karena akan lebih efektif dibandingkan dengan deteksi bagian pupil. Pada penelitian ketiga dapat diambil kesimpulan bahwa pendeteksian area mata dapat menggunakan segmentasi biner, hal ini ditujukan untuk mempercepat proses eksekusi program. Penelitian keempat membuktikan bahwa framework yang didapatkan dari metode pengenalan maupun pendeteksian area wajah dapat diproses agar bisa melakukan proses pendeteksian dengan lebih akurat.

Beberapa poin di atas menjelaskan mengenai proses deteksi pergerakan mata yang dapat disimpulkan sebagai berikut: Proses deteksi bagian mata menggunakan *facial landmark*, setelah bagian mata terdeteksi maka selanjutnya

bagian mata akan di potong menjadi dua. Kedua bagian mata tersebut akan di segmentasi menjadi biner terlebih dahulu sebelum di deteksi pergerakannya. Untuk mendeteksi pergerakannya bagian yang dideteksi yaitu sclera.

2.2 Landasan Teori

2.2.1 Pengolahan Citra Digital

Pengolahan citra digital merupakan pengolahan suatu citra atau gambar yang digunakan untuk berbagai macam deteksi, seperti mengolah warna pada suatu gambar dengan mengetahui nilai RGB untuk memperoleh suatu informasi yang bertujuan untuk mengetahui obyek yang akan deteksi (Irfan, Sumbodo dan Candradewi 2017). Pada beberapa tahun terakhir pengenalan wajah memiliki pertimbangan yang signifikan dan dihargai sebagai salah satu aplikasi yang menjanjikan dibidang analisis gambar (Dwivedi 2018) dan juga banyak algoritme pengenalan wajah dikembangkan selama beberapa tahun terakhir yang difungsikan sebagai identifikasi atau rekognisi (Mohamed, et al. 2019). Salah satu contoh pengolahan citra digital adalah mengubah warna gambar atau dapat dikatakan menyegmentasi gambar menjadi warna lainnya seperti pada Gambar 2.1. Pada penelitian ini metode yang digunakan berupa algoritme *facial landmark*, yaitu metode untuk mendeteksi posisi wajah maupun posisi bola mata.



(a) Citra RGB



(b) Kanal merah



(c) Kanal hijau



(d) Kanal biru

Gambar 2.1 Pengolahan Citra Digital Segmentasi

Sumber : (pamungkas 2018)

2.2.2 Stroke

Stroke merupakan penyakit serius melemahkan sistem syaraf yang dapat menyebabkan kematian. Meskipun terdapat berbagai kemajuan teknologi medis, sebagian besar pasien yang selamat dari stroke masih mengalami efek samping. Banyak hal yang menyebabkan terjadinya penyakit stroke salah satunya adalah konsumsi alkohol yang berlebihan. Pada penderita stroke yang selamat efek samping yang ditimbulkan beragam seperti melemahnya sistem syaraf untuk dapat bergerak dan pada penderita yang terbilang parah, penderita tidak dapat menggerakkan seluruh badannya (Shiotsuki, et al. 2019). Gambar 2.2 merupakan penderita stroke yang membutuhkan kenyamanan dalam perawatannya.



Gambar 2.2 Penderita Stroke

Sumber : (Liput 2019)

2.2.3 OpenCV

Opencv merupakan library yang banyak dipergunakan pada pemrosesan gambar. Perangkat lunak ini dikembangkan oleh intel corporation. Operasi-operasi yang terdapat di dalamnya terdiri dari bermacam-macam fungsi seperti deteksi objek, pelacakan dan salah satunya berupa konversi dari gambar RGB ke biner secara otomatis (Sugano dan Miyamoto 2010). Karena itu library opencv sangat dibutuhkan pada pemrosesan gambar agar kode program menjadi lebih singkat dan mudah. Pada penelitian ini opencv banyak dipergunakan untuk mengolah data gambar agar data gambar bisa dideteksi oleh metode *facial landmark*.

2.2.4 Resize

Gambar memiliki suatu ukuran berupa pixel. Ukuran pada gambar terdapat dua yaitu resolusi dan rasio aspek. Untuk menampilkan suatu gambar pada berbagai perangkat, terdapat ukuran dan aspek rasio yang harus disesuaikan terlebih dahulu agar perangkat dapat berfungsi maksimal (Hashemzadeh, Asheghi dan Farajzadeh 2019). Resize merupakan suatu metode untuk mengubah resolusi atau rasio aspek menjadi lebih besar atau lebih kecil. Pada penelitian ini resize digunakan untuk mengubah resolusi dari gambar menjadi lebih kecil. Hal tersebut bertujuan untuk mempercepat proses eksekusi program (Hashemzadeh, Asheghi dan Farajzadeh 2019).

2.2.5 Crop Image

Crop image merupakan suatu metode untuk mendapatkan suatu area gambar pada gambar tertentu dengan cara memotong bagian tersebut dan menjadikannya gambar baru. Pada teknik cropping kita dapat mengambil suatu bagian gambar dengan cara memotong gambar yang akan disesuaikan ukurannya seperti pada Gambar 2.3 (Zhou, et al. 2019). Sistem ini menggunakan teknik crop yang digunakan untuk cropping bagian mata dari gambar wajah.



Gambar 2.3 Cropping Gambar

Sumber : Journal, Scale adaptive image cropping for UAV object detection

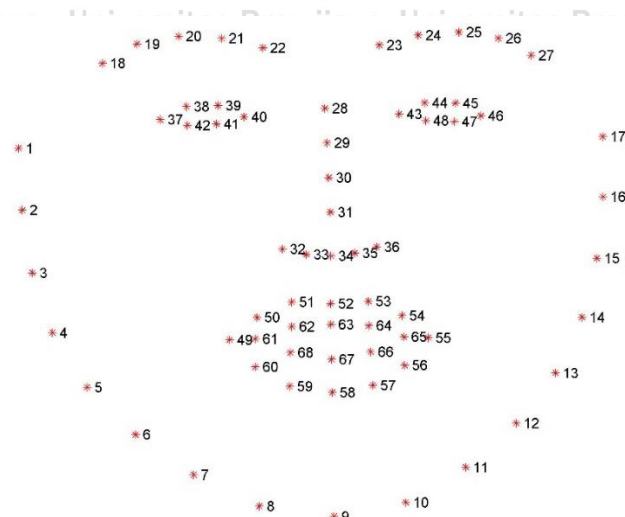
2.2.6 Dlib

Library dlib merupakan alat bantu untuk menganalisis bagian wajah/ muka seseorang dengan cara mengekstrak nilainya (Niu dan Chen 2018). Dengan menggunakan dlib wajah dari manusia dapat menghasilkan 128 dimensional feature vector (Chen dan Sang 2018). Library dlib pada penelitian ini digunakan untuk mengekstrak wajah manusia agar didapat dipergunakan pada metode *facial landmark*.

2.2.7 Facial landmark

Metode *facial landmark* adalah sebuah titik-titik pada wajah untuk menentukan bentuk suatu biologis (Vezzetti, et al. 2012). Pada penggunaannya *facial landmark* mengalokasikan beberapa titik pada bagian wajah yang akan ditampilkan secara otomatis ketika wajah seseorang terdeteksi oleh kamera (Han, et al. 2019). Dari kedua pemahaman di atas dapat disimpulkan bahwa face landmark berfungsi untuk mengalokasikan titik-titik pada wajah secara otomatis yang bertujuan untuk memprediksikan lokasi bentuk dari wajah tersebut yang berupa mata, mulut dan hidung. Pada Gambar 2.4 merupakan contoh titik-titik yang akan dialokasikan pada wajah.

Pada penelitian ini, hanya difokuskan untuk mendeteksi bagian mata. Untuk itu titik-titik yang diambil hanya yang terdapat pada bagian mata saja. Titik-titik yang terdapat pada bagian mata yaitu di antara titik 37 hingga titik 48. Titik 37 hingga titik 48 nantinya akan diproses sehingga didapatkan citra mata.



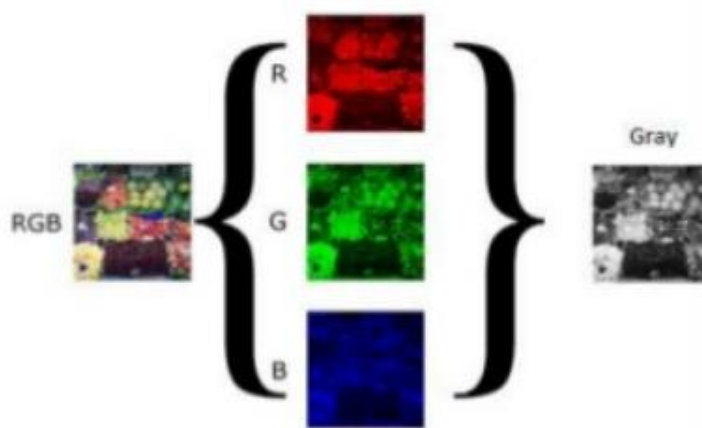
Gambar 2.4 Face Landmark

Sumber : (Rosebrock 2017)

2.2.8 Konversi RGB ke Biner

Untuk mendapatkan gambar biner dari gambar RGB, langkah awalnya dengan cara mengubah gambar RGB menjadi gambar grayscale terlebih dahulu dengan rumus persamaan 2.1 (Utaminigrum, et al. 2016). Persamaan tersebut ketika diilustrasikan akan tampak seperti pada Gambar 2.5. Setelah gambar menjadi grayscale barulah kita konversi kembali menjadi biner dengan memberi nilai treshold/ nilai batasan. Gambar 2.6 merupakan contoh gambar yang sudah di rubah menjadi gambar biner.

$$X_{(i,j)} = \frac{(Red_{(i,j)} + Green_{(i,j)} + Blue_{(i,j)})}{3} \quad (2.1)$$



Gambar 2.5 Konversi RGB ke GraySclae

Sumber : (LinkedIn 2003)



Gambar 2.6 Gambar Biner

Sumber : (gnovice 2013)

2.2.9 Probabilitas

Probabilitas merupakan suatu ukuran mengenai kemungkinan peristiwa dapat terjadi dimasa mendatang. Probabilitas dapat dinyatakan menggunakan nilai 0 sampai dengan nilai 1 dalam persentase. Probabilitas dapat digunakan untuk mengambil keputusan terhadap beberapa faktor yang mempengaruhinya. Persamaan probabilitas dapat dituliskan seperti pada persamaan 2.2 sehingga didapatkan nilai 0 sampai dengan nilai 1 (Setiawan 2010). Fungsi probabilitas digunakan pada penelitian ini untuk menghitung probabilitas pergerakan mata berdasarkan biner yang digunakan untuk menentukan pergerakan ke kiri, kanan, atas, bawah dan depan.

$$\text{Probabilitas} = \frac{\text{Jumlah data yang dianalisis}}{\text{Jumlah total data}} \quad (2.2)$$

2.2.10 Perhitungan Probabilitas Berdasarkan biner mata

Pada pergerakan mata terdapat tiga pergerakan yang dapat dideteksi yaitu melihat kiri, melihat kanan dan melihat depan. Untuk dapat mendeteksi pergerakan mata, maka dibutuhkan suatu metode yaitu dengan menggabungkan metode probabilitas berdasarkan biner. Pada persamaan 2.3, dijelaskan bahwa prediksi pergerakan mata didapatkan dengan cara membagi area putih pada bagian kiri dengan bagian kanan sehingga didapatkan nilai deteksi.

$$\text{Pergerakan mata}_{(i,j)} = \frac{\text{Area putih kiri}_{(i,j)}}{\text{Area putih kanan}_{(i,j)}} \quad (2.3)$$

2.2.11 Perhitungan Probabilitas Berdasarkan panjang garis vertikal dan horizontal

Untuk mendapatkan pergerakan mata ke atas dan ke bawah digunakan suatu metode yaitu perhitungan probabilitas berdasarkan panjang garis vertikal

dan horizontal yang terdapat pada mata. Garis vertikal dan horizontal diukur berdasarkan panjang dan tinggi mata, setelah itu dilakukan perhitungan probabilitas yang terdapat pada persamaan 2.4. Ketika mata melihat ke atas maka tinggi dari mata akan bertambah dan ketika mata melihat ke bawah maka tinggi dari mata akan berkurang. Berdasarkan hal tersebut didapatkan nilai untuk mendeteksi pergerakan mata.

$$Kedipan\ mata_{(i,j)} = \frac{Garis\ vertikal_{(i,j)}}{Garis\ horizontal_{(i,j)}} \quad (2.4)$$

2.2.12 User Interface Menu Display

User interface merupakan suatu visualisasi terhadap proses sistem untuk berinteraksi terhadap pengguna. Pada user interface terdapat banyak desain yang dapat digunakan salah satunya adalah menu display. Menu display adalah suatu desain yang berisi pilihan menu yang digunakan sistem untuk berinteraksi dengan pengguna. Pada menu display cocok dipergunakan untuk menampilkan berbagai macam pilihan menu, salah satunya adalah untuk menampilkan menu makanan seperti pada Gambar 2.7 (Filimonau, et al. 2017). Sedangkan pada penelitian ini menu display digunakan untuk menampilkan beberapa kebutuhan yang diperlukan oleh penderita stroke atau disabilitas, salah satunya digunakan untuk menampilkan menu makanan.



Gambar 2.7 Menu Display Pada Makanan

Sumber : (freehanddesignprint 2013)

BAB 3 METODOLOGI

Dalam proses penelitian terdapat beberapa hal yang perlu dilakukan dan di rencanakan, dalam hal ini proses tersebut saling keterkaitan antara satu dengan yang lainnya. Metodologi akan menjelaskan mengenai metodologi penelitian yang akan di gunakan.

3.1 Tipe Penelitian

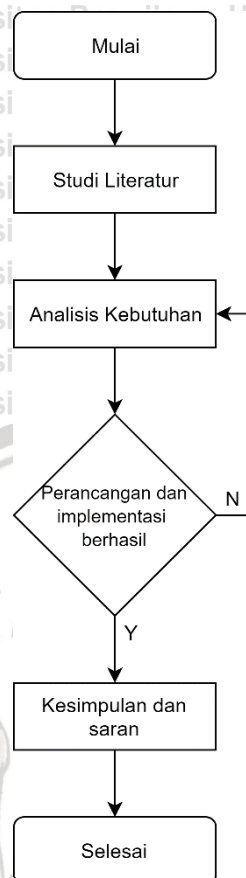
Pada penelitian ini yang berjudul “Deteksi Pergerakan Mata dan Kedipan Mata Untuk Memilih Menu Display Menggunakan Probabilitas Berdasarkan *Facial landmark*” merupakan penelitian implementatif yang bertipe pengembangan lanjut atau enhancement. Penelitian sebelumnya yang berjudul “Eye Movement as Navigator for Disabled Person” menjadi acuan utama dikarenakan fokus penelitian yang dikerjakan hampir sama, yaitu adalah deteksi pergerakan mata. Perbedaan pada penelitian sebelumnya adalah metode yang digunakan yaitu haar cascade dengan batasan latar belakang sepi sedangkan pada penelitian ini menggunakan metode *facial landmark* yang dapat diimplementasikan pada latar belakang ramai.

3.2 Strategi dan Rancangan Penelitian

Bagian ini akan menjelaskan mengenai strategi dalam perancangan penelitian yang meliputi strategi, partisipan, lokasi, metode dan peralatan.

3.2.1 Metode Secara Umum

Dalam proses penelitian terdapat beberapa hal yang perlu dilakukan dan di rencanakan, dalam hal ini proses tersebut saling keterkaitan antara satu dengan yang lainnya. Metode secara umum akan menjelaskan mengenai metodologi penelitian yang akan di gunakan. Metode penelitian mencakup beberapa proses di antaranya adalah studi literatur untuk mencari informasi mengenai masalah yang terkait, analisa kebutuhan untuk mencari spesifikasi alat dari sistem, perancangan dan implementasi untuk merakit dan mengimplementasikan sistem, pengujian dan hasil untuk mengetes tingkat akurasi dan kinerja sistem, kesimpulan diberikan untuk mengevaluasi kinerja sistem dan saran dituliskan untuk membantu penelitian ke depannya. Untuk skema metode tersebut, dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Metodologi Penelitian

3.2.2 Subyek Penelitian

Sistem pada penelitian ini ditujukan pada orang disabilitas salah satunya adalah penderita stroke. Namun untuk subyek penelitian ini diujikan kepada orang dengan berbagai jenis umur dan jenis kelamin. Selain itu diambil juga data mengenai jenis mata dengan tipe yang bervariasi seperti bulat, oval dan sipit. Akan tetapi penelitian ini tidak diperuntukkan untuk kelainan mata seperti juling. Orang akan di arahkan untuk memilih menu pada layar dengan cara menggerakkan matanya untuk melihat ke kanan, ke kiri, ke atas, ke bawah dan ke depan.

3.2.3 Lokasi Penelitian

Untuk mengujikan berbagai macam kondisi, lokasi penelitian perlu dikondisikan sedemikian mungkin agar sistem dapat melakukan pengujian dengan maksimal. Pada pengujian latar belakang akan diujikan pada ruang kosong untuk latar belakang yang sepi dan di ruang yang penuh benda atau halaman rumah untuk latar belakang yang ramai. Pada pengujian pencahayaan ruang kosong akan dikondisikan gelap dan terang. Untuk itu lokasi penelitian yang digunakan adalah ruang kosong, halaman rumah dan ruang yang penuh benda.

3.2.4 Teknik Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengambil data gambar wajah orang dengan cara memfoto dan mengambil video orang tersebut ketika menggerakkan matanya ke kanan, ke kiri, ke depan, ke atas dan ke bawah. Subjek yang akan diuji akan diposisikan wajahnya tepat di depan kamera, dengan jarak wajah terhadap kamera 30cm, 40cm dan 50cm. Untuk melakukan pengujian cahaya, maka kondisi ruangan akan dikondisikan redup dan terang.

3.2.5 Teknik Analisis Data dan Pembahasan

Untuk melakukan analisis data, data uji kuantitatif yang sebelumnya telah didapatkan akan dilakukan analisis dengan cara memasukkan gambar dan video data uji ke dalam program pendeteksian pergerakan mata. Pada data uji gambar, analisis nilai akurasi dilakukan dengan memasukkan data uji gambar ke dalam program yang dapat mengolah gambar menjadi output berupa deteksi arah mata melihat. Sedangkan pada data uji video, analisis nilai akurasi dilakukan dengan memasukkan data uji video ke dalam program yang dapat mengolah video untuk dapat dideteksi ke mana arah pergerakan mata tersebut bergerak.

3.2.6 Peralatan Pendukung Yang Digunakan

Alat yang digunakan untuk mendukung penelitian ini adalah laptop, raspberry pi, LCD, Sdcard, Power supply, web kamera, speaker dan sensor lux. Laptop digunakan sebagai alat untuk memprogram sistem. Untuk menjalankan sistem menggunakan microcontroller raspberry pi, LCD, speaker dan sensor web kamera digunakan sebagai komponen utama dari sistem sedangkan sdcard digunakan sebagai media penyimpanan dan power supply digunakan sebagai power raspberry pi. Untuk melakukan pengujian dibutuhkan sensor lux untuk mengukur tingkat pencahayaan.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem deteksi pergerakan mata dan deteksi kedipan mata ini bekerja dengan cara menggabungkan berbagai metode. Input berupa gambar diambil dengan menggunakan sensor kamera, setelah input didapatkan selanjutnya microcontroller raspberry pi akan memproses gambar input tersebut. Library opencv dan dlib dipasangkan pada raspberry pi yang berfungsi sebagai metode bantu untuk memproses gambar agar gambar bisa dideteksi bagian matanya dengan menggunakan metode *facial landmark*. Setelah bagian mata didapatkan lalu metode deteksi pergerakan mata dan kedipan mata digunakan untuk memprediksi arah pergerakan mata. Hasil dari prediksi akan di outputkan pada layar LCD.

4.2 Analisis Kebutuhan Sistem

Sistem ini dapat bekerja karena menggunakan berbagai macam kebutuhan perangkat lunak dan perangkat keras yang diperlukan agar sistem dapat bekerja dengan baik. Kebutuhan dari sistem juga dapat dikategorikan menjadi kebutuhan fungsional dan kebutuhan non fungsional.

4.2.1 Kebutuhan Sistem

Agar sistem dapat berjalan maka sistem membutuhkan beberapa komponen dan juga perangkat lunak. Kebutuhan dari sistem dapat dikategorikan menjadi kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dipergunakan pada sistem ini adalah :

1. Raspberry Pi 3 B+

Raspberry pi merupakan microcontroller yang berfungsi seperti komputer, namun ukurannya lebih kecil dan cocok dipergunakan pada sistem tertanam. Operating sistem pada raspberry pi terdapat banyak sekali namun pada umumnya biasanya menggunakan operating sistem raspbian yaitu operating sistem pengembangan dari Debian buster. Pada raspberry pi terdapat banyak jenisnya, untuk penelitian kali ini menggunakan raspberry pi jenis 3 B+ seperti pada Gambar 4.1 yang memiliki spesifikasi seperti yang terdapat pada Tabel 4.1 (Logitech 1981).

Tabel 4.1 Spesifikasi Raspberry Pi 3 B+

Broadcom	BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
SDRAM	1GB LPDDR2
Wireless LAN	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac

Bluetooth	4.2, BLE
Gigabit Ethernet	USB 2.0 (maksimal throughput 300 Mbps)
Pin GPIO	Extended 40-pin
Port	HDMI, Micro SD, CSI, DSI, 4x USB 2.0, Ethernet,
Power Input	5 V/2.5A DC



Gambar 4.1 Raspberry Pi 3 B+

Sumber : (Raspberrypi 2012)

2. Webcam Logitech c170

Dikutip dari journal "Modelling and hardware implementation of quantization levels of digital cameras in DCT based image compression" pada masa komputasi sekarang, banyak sistem embedded yang datanya diambil melalui kamera digital. Terlebih lagi ketika menggunakan image processing maka kamera digital sangat dibutuhkan (Dixit dan Vijaya 2019). Pada web kamera sendiri merupakan kamera kecil yang dapat terhubung dengan kabel usb agar data dari kamera dapat diakses dengan langsung pada suatu komputer. Web kamera memiliki banyak keunggulan salah satunya yaitu bentuknya yang kecil namun kualitas dari gambar dapat dibilang bagus, selain itu juga dapat digunakan untuk video kamera yang sangat dibutuhkan pada berbagai macam sistem pemrosesan gambar agar bisa berjalan secara real time/ langsung (Thajee, et al. 2018). Pada webcam logitech c170 memiliki

banyak keunggulan, salah satunya adalah kompatibel terhadap microcontroller raspberry pi dan ukurannya sangat kecil sehingga cocok digunakan pada raspberry pi maupun sistem tertanam lainnya. Pada webcam logitech c170 memiliki spesifikasi yang dapat dilihat pada tabel 4.2 (Logitech 1981). Gambar 4.2 merupakan bentuk fisik dari webcam logitech c170.

Tabel 4.2 Spesifikasi Webcam Logitech c170

Kualitas Foto	5Mp
Kualitas Video	640 x 480 pixel
Usb	2.0
Lain-lain	Terdapat mikrofon



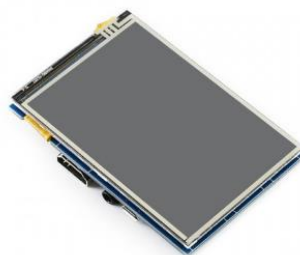
Gambar 4.2 Webcam Logitech c170

3. 3.5Inch HDMI LCD

LCD merupakan salah satu media terbaru untuk menampilkan gambar dengan cara menggunakan liquid crystal/ kristal cair. Pada kegunaannya LCD sering dipakai pada televisi, komputer monitor, layar android dsb (Luo, et al. 2019). Pada penelitian ini LCD dipakai sebagai media penampil/ output dari microcontroller raspberry pi. Pada penelitian kali ini menggunakan 3.5inch hdmi LCD sebagai media penampil/ output. Tabel 4.3 merupakan spesifikasi dari 3.5inch hdmi LCD yang akan digunakan (waveshare 2016). Gambar 4.3 merupakan bentuk dari 3.5Inch HDMI LCD.

Tabel 4.3 Spesifikasi 3.5Inch HDMI LCD

Resolusi	480 x 360 pixel
Touch screen	Resistive touch control
Compatible	Semua raspberry pi
Port	HDMI Port
Pengaturan	Pengaturan kecerahan, kontras



3.5Inch HDMI LCD

480x320, IPS

Gambar 4.3 3.5Inch HDMI LCD

Sumber : (waveshare 2016)

4. SD Card Micro SDHC

Sd card digunakan sebagai media penyimpanan pada microcontroller raspberry pi. Sd card yang dipergunakan bertipe Micro SDHC class 10 yang memiliki kapasitas sebesar 16Gb. Pada SDHC Class 10 ini memiliki kecepatan membaca sebesar 10mb/sec. Tabel Spesifikasi dapat dilihat pada Tabel 4.4 (sdcard 2014). Gambar 4.4 merupakan bentuk dari Sd card Micro SDHC.

Tabel 4.4 Spesifikasi Sd Card Micro SDHC

Tipe	Micro SDHC Class 10
Kapasitas Penyimpanan	16Gb
Kecepatan	10MB/ sec



Gambar 4.4 Sd Card Micro SDHC

Sumber : (sdcard 2014)

5. Power Supply

Power supply digunakan untuk menghidupkan raspberry pi. Power supply untuk pada raspberry pi membutuhkan daya sebesar 3 ampere. Gambar 4.5 merupakan power supply yang digunakan sebagai power pada raspberry pi



Gambar 4.5 Power Supply

6. Lux Sensor

Lux sensor merupakan sensor yang dapat mengukur kapasitas pencahayaan pada suatu tempat. Lux sensor digunakan sebagai pengukur kecerahan ketika pengambilan data uji yang digunakan untuk pengujian pencahayaan pada 1-49 pada keadaan redup dan lux 50-400 pada keadaan terang. Lux sensor yang digunakan adalah Sunche Light Meter HS1010 yang memiliki spesifikasi seperti pada Tabel 4.5. Gambar 4.6 merupakan bentuk dari Light Meter HS1010.

Tabel 4.5 Spesifikasi Light Meter HS1010

Range Pengukuran	2000/20000/200000 Lux
Frekuensi sampling	2 kali per detik
Akurasi	4% pada lux < 10000, 5% pada lux > 10000
Resolusi	1 Lux/0.1Fc



Gambar 4.6 Light Meter HS1010

7. Speaker

Speaker digunakan untuk mengoutputkan suara pada raspberry pi karena microcontroller raspberry pi tidak memiliki perangkat yang mengeluarkan suara maka speaker digunakan untuk mengatasi itu. Speaker yang digunakan harus memiliki kabel AUX audio untuk dapat dicolokkan ke raspberry pi. Gambar 4.7 merupakan speaker yang akan digunakan pada sistem ini.



Gambar 4.7 Speaker

8. Laptop

Laptop digunakan sebagai alat bantu memprogram sistem dan melakukan uji coba program agar dapat diimplementasikan pada microcontroller raspberry pi. Laptop yang digunakan pada proyek ini bertipe HP Pavilion Notebook - 14-ab134tx yang memiliki spesifikasi seperti pada Tabel 4.6 (support.hp.com 2019). Bentuk dari laptop ini dapat dilihat pada gambar 4.8.

Tabel 4.6 Spesifikasi HP Pav Notebook 14-ab134tx

Memory	4 GB DDR3L-1600 SDRAM (1 x 4 GB)
Microprocessor	Intel® Core™ i7-6500U with Intel® HD Graphics 520 (2.5 GHz, 4 MB cache, 2 cores)
Video Grafis	NVIDIA GeForce 940M (2 GB DDR3L dedicated)
Hard Drive	1 TB 5400 rpm SATA
Kamera	HP TrueVision HD Webcam

**Gambar 4.8 HP Pav Notebook 14-ab134tx**

4.2.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang diperlukan oleh sistem adalah :

1. Sistem Operasi Raspbian

Sistem operasi raspbian akan dipakai pada microcontroller raspberry pi. Pada sistem operasi raspbian sudah tersedia berbagai macam software yang dibutuhkan, di antaranya adalah python ide yang dapat digunakan untuk memprogram sistem.

2. Sistem Operasi Windows 10

Sistem operasi windows 10 akan dipergunakan pada laptop. Sistem operasi windows 10 akan di instal python ide yang berfungsi untuk uji coba program pada laptop.

3. Python 3.6

Python 3.6 merupakan compiler yang digunakan untuk memprogram sistem agar dapat bekerja. Pada raspberry pi python 3.6 terdapat pada thonny

IDE yang sudah terinstal pada OS raspbian. Sedangkan pada laptop python 3.6 dipasang pada Spyder.

4. Library OpenCV 4.1.0

Library opencv akan digunakan untuk mengolah data gambar seperti cropping, segmentasi, resize dan memunculkan windows. Dengan menggunakan library ini kita dapat memproses gambar dengan mudah dan simpel.

5. Library Dlib

Library dlib digunakan untuk mengekstrak gambar wajah yang akan digunakan pada metode face landmark. Dari gambar wajah tersebut akan dideteksi titik-titik yang berada pada wajah sehingga dapat dipetakan.

6. VNC Viewer

Vnc viewer digunakan untuk mengontrol raspberry pi agar bisa di kontrol melalui laptop. Aplikasi ini berfungsi untuk memudahkan peneliti agar dapat melakukan kegiatan pada raspberry pi tanpa membutuhkan keyboard dan LCD tambahan dengan mengontrolnya pada laptop.

7. Face Landmark

Merupakan metode deteksi titik-titik dari wajah yang akan menampilkan beberapa poin yang terdapat pada wajah yang di ambil dari beberapa data latih yang terdapat pada face landmark. Poin-poin tersebut diambil dari library dlib.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional pada sistem mencakup bagaimana sistem dapat bekerja secara keseluruhan. Mulai dari sistem dapat mengambil input, memproses data gambar hingga menghasilkan output. Untuk itu kebutuhan fungsional akan dijelaskan secara rinci pada deskripsi berikut :

1. Logitech Webcam

Logitech webcam digunakan untuk mengambil data gambar secara langsung/ dengan cara streaming. Hal ini ditujukan agar deteksi pergerakan mata dapat dideteksi dan diproses secara real time.

2. Raspberry pi 3 B+ sebagai microcontroller

Microcontroller pada sistem menggunakan raspberry pi yang bekerja pada seluruh sistem. Raspberry pi akan digunakan untuk memproses data gambar hingga menjadi deteksi.

3. Proses pengolahan data gambar

Untuk mengolah data gambar yang mentah dibutuhkan library opencv dan dlib. Langkah awalnya gambar rgb akan diubah menjadi gambar biner dengan bantuan opencv. Setelah data biner didapatkan dlib akan digunakan untuk mendapatkan ekstraksi nilai gambar dari wajah. Barulah metode *facial landmark* dapat digunakan.

4. Metode *Facial landmark*

Setelah dlib dapat mengekstrak gambar wajah, metode face landmark akan mendeteksi gambar tersebut dengan memberikan angka/ poin pada wajah.

Pada penelitian ini poin yang digunakan hanya yang terdapat pada point mata saja yaitu mulai dari poin 37 hingga poin 48. Bagian mata tersebut akan dipotong dan dijadikan gambar baru yang akan masuk pada metode deteksi pergerakan.

5. Deteksi pergerakan

Agar pergerakan mata dapat dideteksi, sistem menggunakan metode probabilitas yang digunakan pada biner untuk mendeteksi gerakan ke kiri, kanan dan depan lalu probabilitas yang digunakan pada garis vertikal dan horizontal untuk mendeteksi gerakan ke atas dan bawah.

6. Implementasi pemilihan menu

Setelah pergerakan mata dapat dideteksi maka deteksi tersebut diimplementasikan pada pemilihan menu yang akan dioutputkan pada layar LCD.

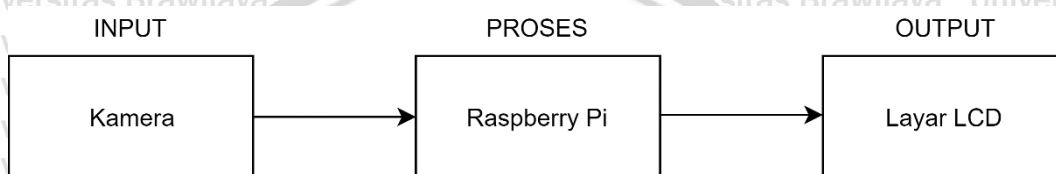


BAB 5 PERANCANGAN DAN IMPLEMENTASI

Setelah bagian rekayasa kebutuhan selesai didefinisikan, maka selanjutnya memasuki tahap perancangan. Tahap perancangan berisi bagaimana sistem dirakit. Setelah perancangan selesai dilakukan, sistem akan di implementasikan dengan cara di rakit pada bagian perangkat kerasnya dan dilakukan pemrograman pada perangkat lunaknya.

5.1 Perancangan Sistem

Pada Gambar 5.1 menjelaskan mengenai cara kerja sistem secara keseluruhan, mulai dari pengambilan input hingga output pada sistem.



Gambar 5.1 Diagram Block Sistem

Penjelasan diagram block pada gambar 5.1 adalah sistem pada awalnya akan mengambil data input berupa objek wajah, setelah itu objek wajah akan di proses pada raspberry pi untuk diambil bagian mata. Bagian gambar mata yang diambil akan di deteksi pergerakannya untuk mengontrol pemilihan menu yang terdapat di layar LCD.

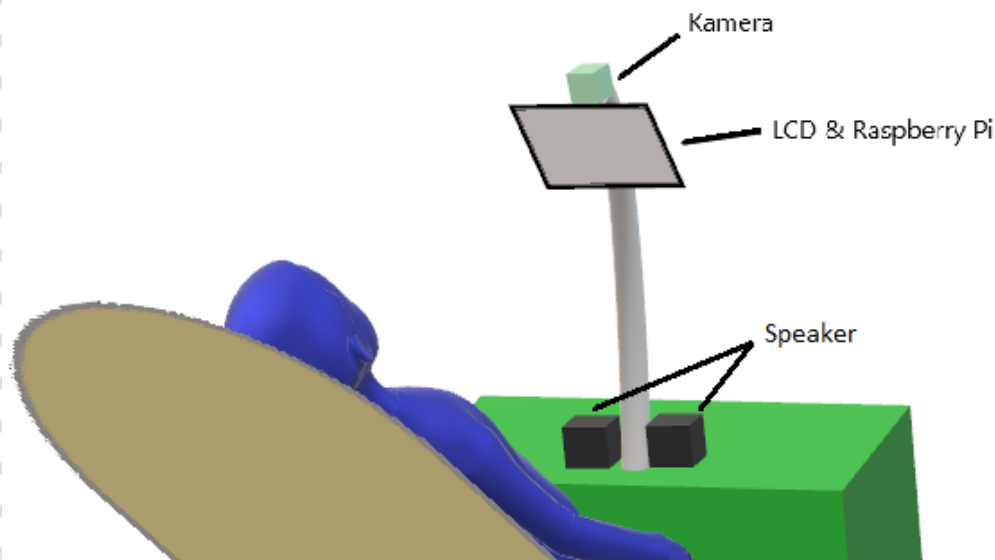
5.1.1 Perancangan Perangkat Keras

Pada bagian ini akan menjelaskan mengenai perancangan perangkat keras mulai dari prototipe alat dan pemasangan komponen.

5.1.1.1 Prototipe Alat

Prototipe alat bertujuan agar sistem ditempatkan pada tempat yang tepat agar prediksi pendeteksian mata mendapatkan akurasi yang tinggi. Untuk itu peletakan dari alat harus mencakup beberapa kebutuhan di antaranya adalah :

1. Sistem pencahayaan harus mencakup muka dari pengguna dan harus tampak terang.
2. Kamera harus dapat menangkap gambar dari wajah pengguna. Peletakan alat dapat digambarkan dengan ilustrasi seperti pada Gambar 5.2.



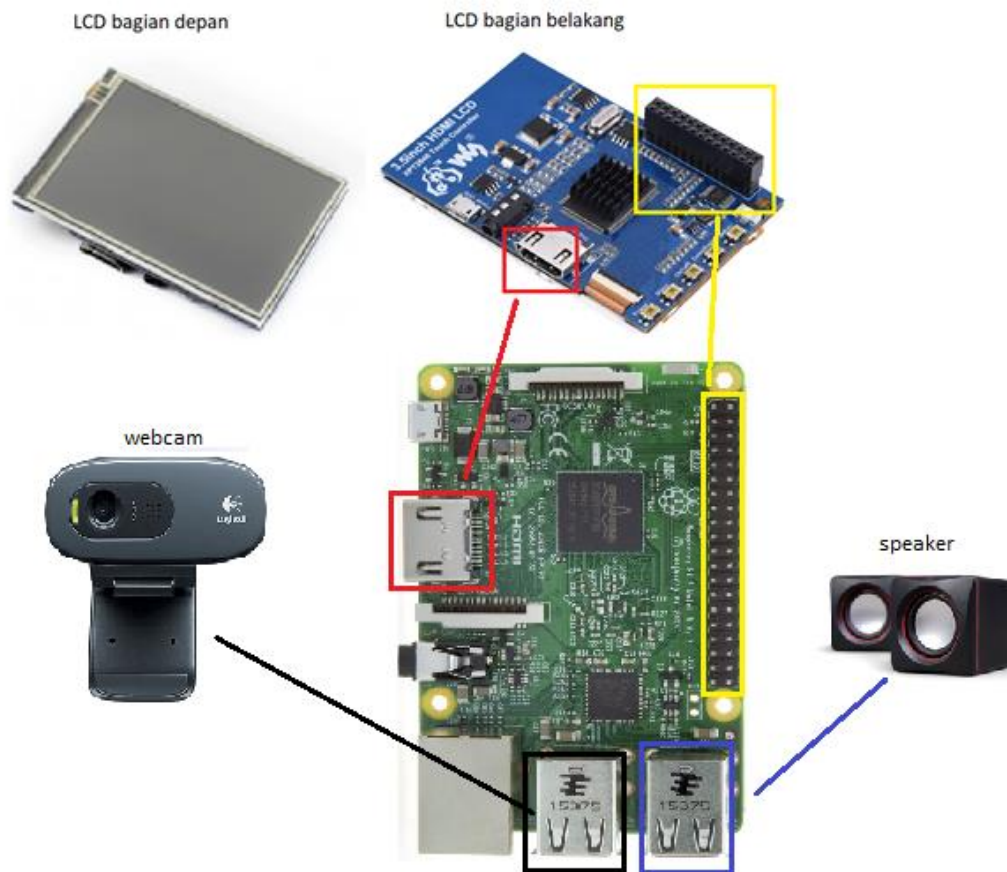
Gambar 5.2 Ilustrasi Peletakan Alat

5.1.1.2 Perancangan Pemasangan Komponen

Pada perancangan komponen mencakup pemasangan dari kamera yang dipasangkan pada raspberry pi dan pemasangan LCD pada raspberry pi. Gambar 5.3 menjelaskan pemasangan komponen tersebut.

Tabel 5.1 Konfigurasi Pin Raspberry Pi

Pin Raspberry pi	Komponen
Port USB	webcam
Port USB	speaker
HDMI Port	HDMI LCD
Pin 1-26	Pin 1-26 LCD



Gambar 5.3 Perancangan Pemasangan Komponen

Pada Tabel 5.1 menunjukkan pin - pin komponen yang dipasangkan pada raspberry pi. Komponen webcam dipasangkan pada port usb yang terdapat pada raspberry pi, komponen speaker dipasangkan pada port usb, sedangkan komponen LCD terdapat 2 komponen yang harus dipasangkan yaitu port display yang terdapat pada lcd harus dipasangkan pada port hdmi yang terdapat pada raspberry pi dan pin 1-26 pada LCD harus dipasangkan pada port 1-26 pada raspberry pi. Gambar 5.3 menunjukkan bagaimana komponen tersebut dipasangkan pada tempatnya.

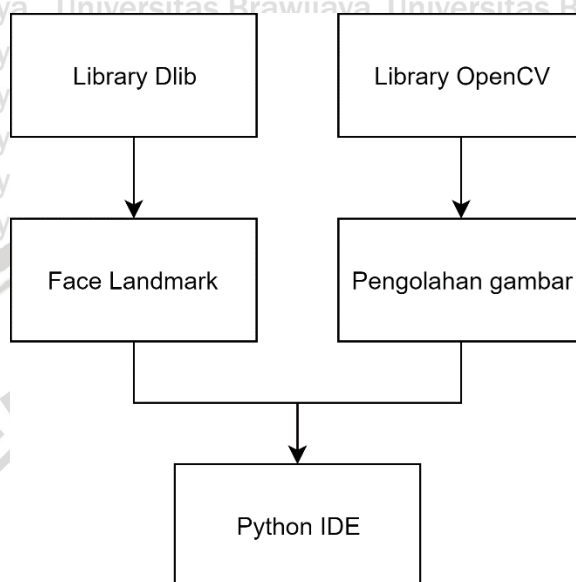
5.1.2 Perancangan Perangkat Lunak

Bagian ini menjelaskan tentang perancangan perangkat lunak mulai dari perancangan pemasangan library, deteksi pergerakan mata dan kedepan mata.

5.1.2.1 Perancangan Pemasangan Library

Langkah awal dalam perancangan perangkat lunak adalah dengan memasang library pada python ide. Library yang perlu dipasangkan adalah library dlib dan library opencv. Library dlib dibutuhkan untuk mengakses metode *facial landmark*, dlib bekerja dengan cara mengekstrak gambar pada wajah agar *facial landmark* dapat mengolah data wajah agar terdapat poin-poin nilai pada

wajah. Library opencv digunakan untuk mengolah gambar, agar gambar dapat diubah ukurannya, mengubah gambar RGB menjadi biner dan memberi gambar berbagai macam tulisan. Pada gambar 5.4 dijelaskan bahwa pemasangan library dlib digunakan untuk *facial landmark* dan library opencv digunakan untuk pengolahan gambar yang dipasangkan pada python ide.



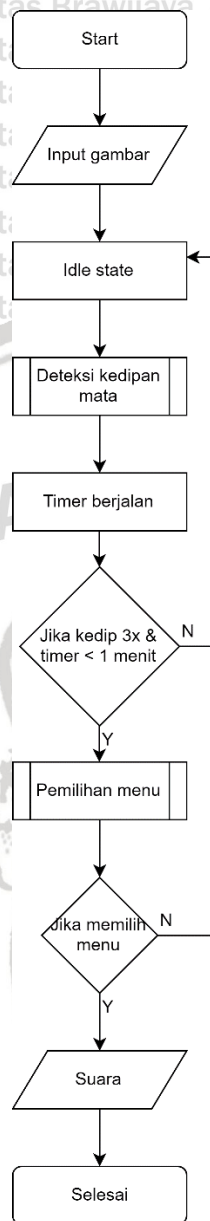
Gambar 5.4 Diagram Block Penggunaan Library

5.1.2.2 Perancangan Sistem Utama

Perancangan sistem utama ini memuat mengenai seluruh kinerja sistem dari sistem hidup, memproses data gambar dan melakukan pemilihan menu hingga sistem mengeluarkan output. Untuk menjelaskan perancangan sistem utama gambar 5.5. Untuk menjelaskan flowchart deskripsi di bawah dituliskan sebagai berikut :

1. Saat sistem dihidupkan maka sistem akan berubah menjadi keadaan idle, keadaan idle yaitu keadaan dimana sistem tidak dapat mengeluarkan output dan hanya bisa mendeteksi pergerakan kedipan mata saja.
2. Untuk mengubah keadaan idle ke pemilihan menu, pengguna harus mengedipkan matanya sebanyak tiga kali.
3. Ketika pengguna mengedipkan matanya pada hitungan pertama, maka timer akan berjalan. Ketika pengguna mengedipkan matanya sebanyak 3x dan timer menunjukkan kurang dari 1 menit, maka menu pemilihan akan muncul. Dan ketika timer melebihi 1 menit, maka hitungan kedipan pengguna akan di reset. Hal ini bertujuan untuk membedakan pengguna ketika sedang berkedip secara alami, dengan berkedip untuk memilih menu.
4. Setelah menu pemilihan muncul, pengguna dapat memilih menu tersebut. Ketika pengguna memilih menu, maka sistem akan mengeluarkan output berupa suara, namun ketika pengguna tidak memilih menu atau bisa

dikatakan pengguna memilih menu exit, maka sistem akan kembali ke keadaan idle.



Gambar 5.5 Flowchart Sistem Utama

5.1.2.3 Perancangan Deteksi Pergerakan Mata

Setelah library terpasang pada python ide, selanjutnya adalah merancang deteksi pergerakan pada mata, pada deteksi kali ini ditujukan pada perancangan saat mata melihat ke kiri, ke kanan dan tengah. Untuk menjelaskan rancangan deteksi pergerakan mata gambar 5.6 berisi mengenai flowchart deteksi pergerakan mata.

Agar lebih jelas deskripsi mengenai flowchart dituliskan sebagai berikut :

1. Langkah awal untuk mendeteksi pergerakan mata adalah dengan mengambil input gambar menggunakan kamera.

2. Setelah input gambar didapatkan selanjutnya dengan mengubah ukuran gambar menjadi 150 pixel x 100 pixel, hal ini bertujuan agar eksekusi program berjalan lancar dan tidak terjadi lagging.
3. Gambar muka akan dideteksi oleh *facial landmark* dan akan dilakukan cropping atau pemotongan gambar pada bagian mata saja.
4. Gambar mata yang awalnya berupa rgb akan di konversi menjadi gambar biner dengan mengubahnya ke grayscale terlebih dahulu.
5. Langkah selanjutnya adalah membagi gambar menjadi dua bagian untuk dibandingkan nantinya.
6. Kedua bagian gambar mata tersebut akan dikalkulasi menggunakan probabilitas dengan menggunakan persamaan 5.1.

$$Pw_{(i,j)} = \frac{\text{Area putih kiri}_{(i,j)}}{\text{Area putih kanan}_{(i,j)}} \quad (5.1)$$

7. Langkah selanjutnya dengan melakukan pengujian untuk menentukan nilai $Pw_{(i,j)}$ yang digunakan untuk memprediksi pergerakan mata ke kanan, ke kiri dan ke depan. Pengujian untuk mendapatkan akurasi dari data latih dilakukan dengan menggunakan 84 data latih yang terdapat pada Tabel 5.2. Untuk mendapatkan akurasi pengujian dilakukan perhitungan menggunakan persamaan 5.2 yang digunakan untuk mendapatkan nilai akurasi pergerakan ke kiri, ke kanan dan ke depan, lalu untuk mendapatkan akurasi keseluruhan dilakukan perhitungan menggunakan persamaan 5.3. Pada Tabel 5.3 dilakukan pengujian ke satu sehingga didapatkan akurasi rata-rata 85%. Pada Tabel 5.4 dilakukan pengujian ke dua sehingga didapatkan akurasi rata-rata 93%. Pada Tabel 5.5 dilakukan pengujian ke tiga sehingga didapatkan akurasi rata-rata 86%. Sehingga dapat disimpulkan bahwa akurasi terbaik didapatkan pada pengujian ke dua. Nilai dari pengujian ke dua ini yang akan digunakan sebagai nilai prediksi untuk mendeteksi pergerakan mata.

$$\text{Akurasi} = \frac{\sum \text{sample data pada range } Pw_{(i,j)}}{\text{Total seluruh data}} \quad (5.2)$$

$$\begin{aligned} \text{Akurasi rata – rata} \\ = \frac{\text{Nilai akurasi kiri} + \text{depan} + \text{kanan}}{3} \end{aligned} \quad (5.3)$$



Tabel 5.2 Data Latih Deteksi Pergerakan Mata

Gerakan	Nilai Pengujian Pw	Σ sample data	Gerakan	Nilai Pengujian Pw	Σ sample data	Gerakan	Nilai Pengujian Pw	Σ sample data
atas	0.7	2	depan	2.3	2	bawah	4.5	1
	0.7	1		2.3	2		4.6	1
	0.8			2.4			4.9	2
	0.9	3		2.4	5		4.9	2
	0.9			2.5			5	
	0.9	8		2.5	4		5	2
	2			2.5			5.1	
	2			2.5			5.1	3
	2			2.5			5.1	
	2			2.6	5.2		3	
	2			2.6	5.2			3
	2			2.6	5.2		3	
	2			2.6	5.3			3
	2	2.7		5.3	3			
	2.1	15		2.7			10	5.3
	2.1			2.7	5.4			2
	2.1			2.7	5.5			
	2.1			2.7	5.5			2
	2.1			2.7	5.6			
	2.1			2.7	5.6		3	
	2.1			2.7	5.7			2
	2.1			2.7	5.7		2	
	2.1			2.7	5.7			2
	2.1			2.8	5.8		2	
	2.1			2.8	5.8			2
	2.1			2.8	6		2	
	2.1			2.8	6.1			1
	2.1			2.8	6.1		1	
	2.1			2.8	6.2			1
	2.2	11		2.8	7		6.3	
	2.2			2.9			6.4	2
	2.2			2.9			6.4	
	2.2			2.9			6.5	3
	2.2			2.9			6.5	
	2.2			2.9	6.5		3	
	2.2			2.9	6.6			3
	2.2			2.9	6.6		3	
	2.2			3	6.7			3
	2.2			3	6.7		3	
	2.2			3	6.8			3
	2.3	14		3	9		6.8	
	2.3			3			6.9	3
2.3	3		6.9	3				
2.3	3		6.9			3		
2.3	3		7	2				
2.3	3		7		2			
2.3	3		7.1	1				
2.3	3.1		7.2		4			
2.3	3.1		7.2	4				
2.3	3.1		7.2		4			
2.3	3.1	7.2	4					
2.3	3.1	7.3		3				
2.3	3.1	7.3	3					
2.3	3.2	7.3		3				
2.3	3.2	7.4	1					
2.4	6	3.2		8	7.6	3		
2.4		3.2	7.6		2			
2.4		3.2	7.6			2		
2.4		3.2	7.9		2			
2.4		3.2	7.9	2				
2.4		3.2	8		1			
2.5		9	3.3	7		8.2	1	
2.5			3.3		8.4	1		
2.5	3.3		8.5		1			
2.5	3.3		8.6			2		
2.5	3.3		8.6		2			
2.5	3.3		8.7	1				
2.5	3.3		8.8		1			
2.5	3.4		8.9	1				
2.5	3.4		9		1			
2.5	3.4	9	1					
2.5	5	3.4		6	9	4		
2.5		3.4	9		4			
2.5		3.4	9			4		
2.5		3.4	9		4			
2.5	2	3.5	2	9.5		1		
2.6		3.5		9.6	1			
2.6		3.6		9.7		1		
2.7	1	3.7	1	9.9	2			
2.8	2	3.7	2	9.9		2		
2.8		3.9		10	6			
2.9	3	4.6	1	10		6		
2.9		4.8		10	6			
2.9		4.8		10		6		
3	1	4.8	1	10	6			
3.2	1	-	-	10		6		
3.2	1	-	-	11	1			
Tot al data	84	84	Tot al data	84	84	Total data	84	84

Tabel 5.3 Pengujian 1 Deteksi Pergerakan Mata

Gerakan	Nilai Deteksi	Akurasi
Kiri	$Pw_{(i,j)} \leq 1,3$	87%
Depan	$0,7 < Pw_{(i,j)} < 1,3$	78%
Kanan	$Pw_{(i,j)} \leq 0,7$	90%
Akurasi rata-rata		85%

Tabel 5.4 Pengujian 2 Deteksi Pergerakan Mata

Gerakan	Nilai Deteksi	Akurasi
Kiri	$Pw_{(i,j)} \leq 1,2$	99%
Depan	$0,6 < Pw_{(i,j)} < 1,2$	90%
Kanan	$Pw_{(i,j)} \leq 0,6$	90%
Akurasi rata-rata		93%

Tabel 5.5 Pengujian 3 Deteksi Pergerakan Mata

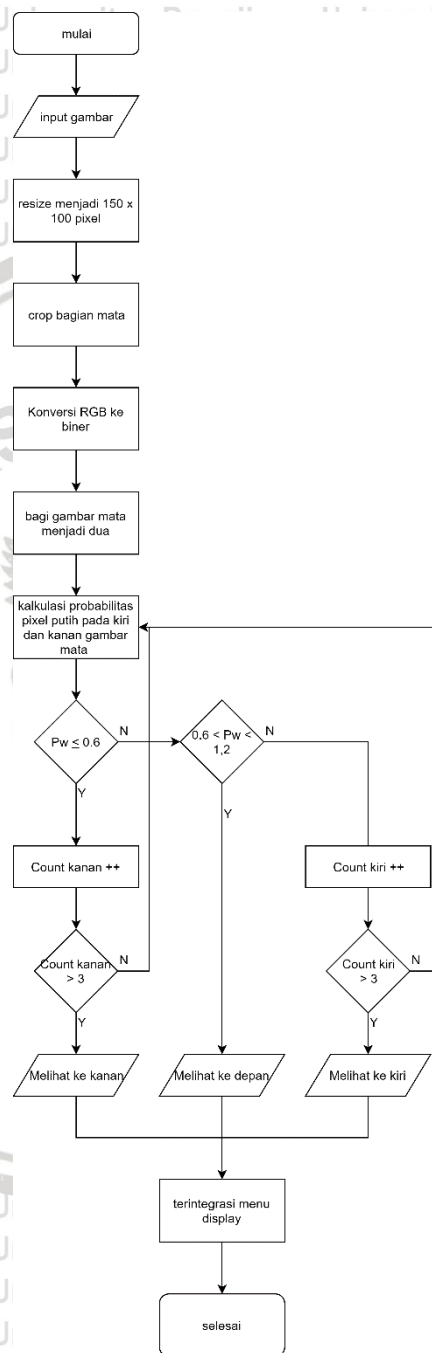
Gerakan	Nilai Deteksi	Akurasi
Kiri	$Pw_{(i,j)} \leq 1,1$	99%
Depan	$0,5 < Pw_{(i,j)} < 1,1$	87%
Kanan	$Pw_{(i,j)} \leq 0,5$	72%
Akurasi rata-rata		86%

8. Persamaan 5.1 digunakan untuk digunakan memprediksi pergerakan mata. Nilai $Pw_{(i,j)}$ digunakan untuk membandingkan deteksi pergerakan mata. Ketika mata melihat ke kanan maka nilai $Pw_{(i,j)}$ yang didapat akan kurang dari sama dengan 0,6, ketika mata melihat ke depan maka nilai $Pw_{(i,j)}$ yang didapat akan lebih dari 1 dan kurang dari 1,2 dan ketika mata melihat ke kiri maka nilai $Pw_{(i,j)}$ yang didapat akan lebih dari sama dengan 1,2 seperti pada Tabel 5.3.

Tabel 5.6 Nilai Deteksi Mata

Pergerakan mata	Nilai $Pw_{(i,j)}$
Melihat ke kiri	$Pw_{(i,j)} \geq 1,2$
Melihat ke depan	$0,6 < Pw_{(i,j)} < 1,2$
Melihat ke kanan	$Pw_{(i,j)} \leq 0,6$

9. Ketika sistem mendeteksi pergerakan mata, maka nilai count kiri atau kanan akan bertambah setiap kali pengguna melihat ke kiri dan ke kanan. Ketika nilai count melebihi 3 sebelum pengguna melihat ke arah lain, maka sistem akan mengoutputkan hasil deteksi.
10. Fungsi deteksi mata akan diintegrasikan untuk pemilihan menu.



Gambar 5.6 Flowchart Deteksi Pergerakan Mata

5.1.2.4 Perancangan Deteksi Kedipan Mata

Pada perancangan kedipan mata digunakan untuk mendeteksi mata ketika berkedip/ melihat ke bawah, melihat ke depan dan melihat ke atas. Untuk

penjelasan lebih lanjut dapat dilihat pada gambar 5.8. Penjelasan mengenai flowchart tersebut lebih lengkapnya adalah :

1. Proses ini hampir sama dengan deteksi pergerakan pada mata namun yang membedakannya dimulai pada proses setelah konversi gambar RGB ke biner
2. Pada proses deteksi kedipan mata menggunakan perhitungan pada panjang dan lebar dari mata.
3. Untuk mengukur panjang dan lebar pixel pada mata, maka perlu dibuat garis vertikal dan horizontal dengan menggunakan bantuan titik pada *facial landmark*. Titik yang digunakan untuk membentuk garis tersebut adalah titik 37,38,40,42,43,44,46 dan 48 seperti pada gambar 5.7.



Gambar 5.7 Face Landmark Mata

4. Setelah garis vertikal dan horizontal dibuat selanjutnya adalah mendapatkan nilai dari garis tersebut dengan menggunakan persamaan 5.4.

$$Pb_{(i,j)} = \frac{\text{Garis vertikal}_{(i,j)}}{\text{Garis horizontal}_{(i,j)}} \quad (5.4)$$

5. Langkah selanjutnya dengan melakukan pengujian untuk menentukan nilai $Pb_{(i,j)}$ yang digunakan untuk memprediksi pergerakan mata ke atas, ke bawah dan ke depan. Untuk mendapatkan akurasi pengujian dilakukan perhitungan menggunakan persamaan 5.5 yang digunakan untuk mendapatkan nilai akurasi pergerakan ke atas, ke bawah dan ke depan, lalu untuk mendapatkan akurasi keseluruhan dilakukan perhitungan menggunakan persamaan 5.6. Pengujian untuk mendapatkan akurasi dari data latih dilakukan dengan menggunakan 84 data latih yang terdapat seperti pada Tabel 5.7. Pada Tabel 5.8 dilakukan pengujian ke satu sehingga didapatkan akurasi rata-rata 85%. Pada Tabel 5.9 dilakukan pengujian ke dua sehingga didapatkan akurasi rata-rata 90%. Pada Tabel 5.10 dilakukan pengujian ke tiga sehingga didapatkan akurasi rata-rata 89%. Sehingga dapat disimpulkan bahwa akurasi terbaik didapatkan pada pengujian ke dua. Nilai dari pengujian ke dua ini yang akan digunakan sebagai nilai prediksi untuk mendeteksi pergerakan kedipan mata.

$$\text{Akurasi} = \frac{\sum \text{sample data pada range } Pb_{(i,j)}}{\text{Total seluruh data}} \quad (5.5)$$

$$\text{Akurasi rata-rata} = \frac{\text{Nilai akurasi atas} + \text{depan} + \text{bawah}}{3} \quad (5.6)$$



Tabel 5.7 Data Latih Deteksi Kedipan Mata

Gerakan	Nilai Pengujian Pw	Σ sample data	Gerakan	Nilai Pengujian Pw	Σ sample data	Gerakan	Nilai Pengujian Pw	Σ sample data		
atas	0.7	2	depan	2.3	2	bawah	4.5	1		
	0.7			2.3			4.6	1		
	0.8	1		2.4	2		4.9	2		
	0.9			2.4			4.9			
	0.9	3		2.5	5		5	2		
	0.9			2.5			5			
	2	8		2.5			5.1	3		
	2			2.5			5.1			
	2			2.5			5.1			
	2			2.6	4		5.2	3		
	2			2.6			5.2			
	2			2.6			5.2			
	2			2.6	10		5.3	3		
	2			2.7			5.3			
	2.1	15		2.7			5.3	1		
	2.1			2.7			5.4			
	2.1			2.7			5.5	2		
	2.1			2.7			5.5			
	2.1			2.7			5.6	2		
	2.1			2.7			5.6			
	2.1			2.7			5.7	3		
	2.1			2.7			5.7			
	2.1			2.8	7		5.7	2		
	2.1			2.8			5.8			
	2.1			2.8			5.8	1		
	2.1			2.8			6			
	2.1	11		2.8	7		6.1	2		
	2.1			2.8			6.1			
	2.2			2.8			6.2	1		
	2.2			2.8			6.3	1		
	2.2			2.9	7		6.4	2		
	2.2			2.9			6.4			
	2.2			2.9			6.5	3		
	2.2			2.9			6.5			
	2.2			2.9			6.5	3		
	2.2			2.9			6.6			
	2.2			3	9		6.6	3		
	2.2			3			6.7	1		
	2.2	3		6.8			1			
	2.3	3		6.9			3			
	2.3	3		6.9						
	2.3	3		6			6.9	2		
2.3	3	7								
2.3	3	7	1							
2.3	3	7.1								
2.3	3.1	8		7.2	4					
2.3	3.1			7.2						
2.3	6			7.2	3					
2.3				3.1		7.2				
2.3				3.1	7.3					
2.3				3.1	7.3					
2.3			3.2	7.3						
2.3			3.2	7.4	1					
2.4	3.2	8		7.6	3					
2.4	3.2			7.6						
2.4	3.2			7.6	2					
2.4	3.2			7.9						
2.4	3.2			7.9						
2.4	3.2	7		8	1					
2.5	3.3			8.2	1					
2.5	3.3			8.4	1					
2.5	3.3			8.5	1					
2.5	3.3			8.6	2					
2.5	3.3			8.6						
2.5	3.3	8.7	1							
2.5	3.3	8.8	1							
2.5	3.4	8.9	1							
2.5	3.4	6		9	4					
2.5	3.4			9						
2.5	3.4			9						
2.5	3.4			9						
2.5	3.4	1		9.5	1					
2.5	3.5			9.6	1					
2.6	2		3.5	1		9.7	1			
2.6			3.6			9.9	2			
2.7	1	3.7	2		9.9					
2.8	2				3.7	10	6			
2.8			3.9	1	10					
2.9	3		4.6	1	10					
2.9			4.8	1	10					
2.9	1		4.8	1	10					
3			-		10					
3.2	1	-		11	1					
Total data	84	84	Total data	84	84	Total data	84	84		

Tabel 5.8 Pengujian 1 Deteksi Kedipan Mata

Gerakan	Nilai Deteksi	Akurasi
Atas	$Pb_{(i,j)} \leq 2,5$	69%
Depan	$2,5 < Pb_{(i,j)} < 4,8$	90%
Bawah	$Pb_{(i,j)} \leq 4,8$	96%
Akurasi rata-rata		85%

Tabel 5.9 Pengujian 2 Deteksi Kedipan Mata

Gerakan	Nilai Deteksi	Akurasi
Atas	$Pb_{(i,j)} \leq 2,6$	87%
Depan	$2,6 < Pb_{(i,j)} < 4,9$	87%
Bawah	$Pb_{(i,j)} \leq 4,9$	96%
Akurasi rata-rata		90%

Tabel 5.10 Pengujian 3 Deteksi Kedipan Mata

Gerakan	Nilai Deteksi	Akurasi
Atas	$Pb_{(i,j)} \leq 2,7$	90%
Depan	$2,7 < Pb_{(i,j)} < 5$	81%
Bawah	$Pb_{(i,j)} \leq 5$	96%
Akurasi rata-rata		89%

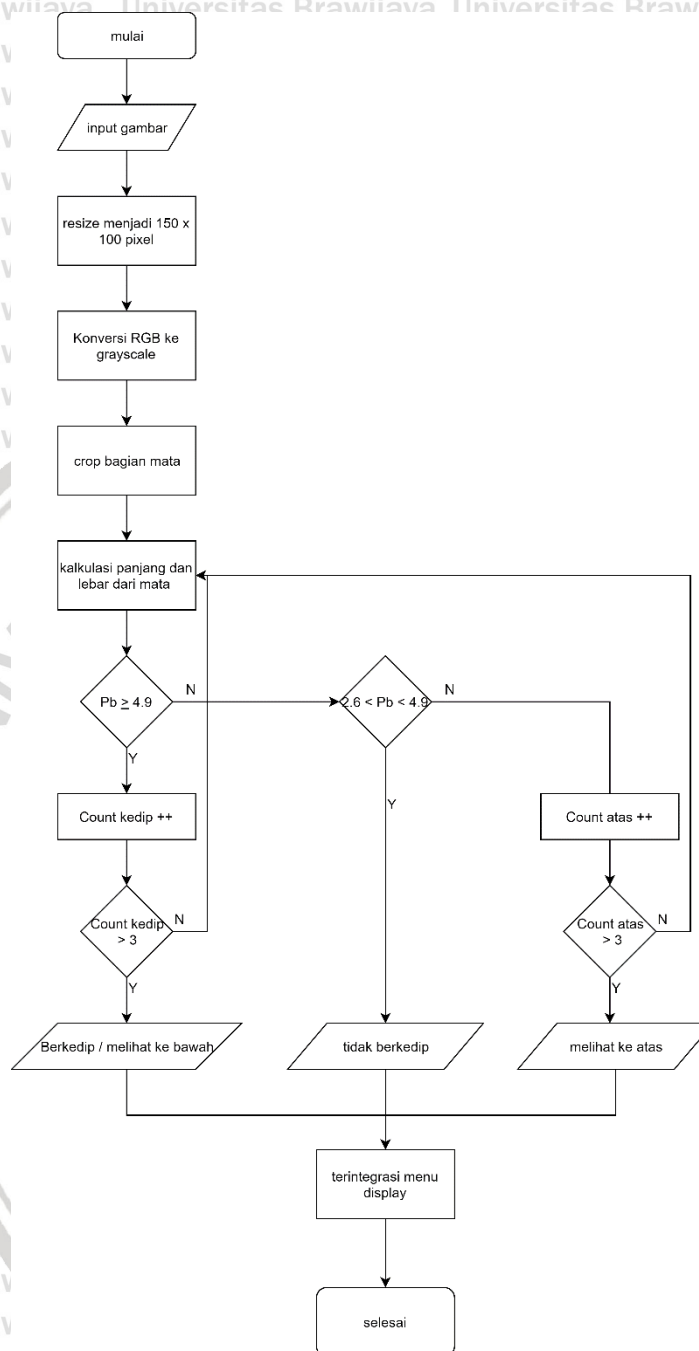
6. Setelah melakukan pengujian, nilai yang didapatkan digunakan untuk memprediksi pergerakan mata dengan cara melakukan probabilitas pada nilai $Pb_{(i,j)}$. Ketika mata melihat ke atas maka nilai $Pb_{(i,j)}$ akan kurang dari sama dengan 2,6, ketika melihat ke bawah atau berkedip maka nilai $Pb_{(i,j)}$ lebih dari sama dengan 4,9 dan ketika melihat ke depan atau tidak berkedip maka nilai $Pb_{(i,j)}$ di antara 2,6 dan 4,9 seperti pada Tabel 5.11.

Tabel 5.11 Nilai Kedipan Mata

Pergerakan mata	Nilai $Pb_{(i,j)}$
Melihat ke atas	$Pb_{(i,j)} \leq 2,6$
Melihat ke bawah	$Pb_{(i,j)} \geq 4,9$
Melihat ke depan/tidak berkedip	$2,6 < Pb_{(i,j)} < 4,9$

7. Ketika sistem mendeteksi pergerakan kedipan mata, maka nilai count kedip atau atas akan bertambah setiap kali pengguna melihat ke atas dan berkedip. Ketika nilai count melebihi 3 sebelum pengguna melihat ke arah lain, maka sistem akan mengoutputkan hasil deteksi.

8. Fungsi deteksi kedipan mata akan diintegrasikan untuk pemilihan menu dan saat sistem masuk ke dalam keadaan idle.

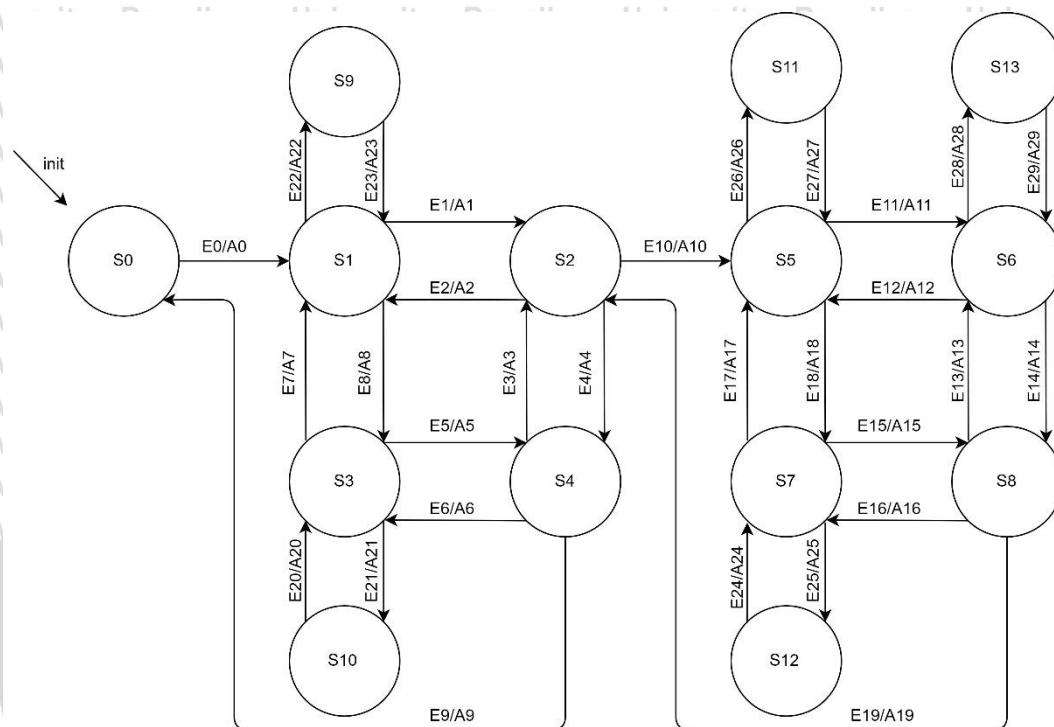


Gambar 5.8 Flowchart Deteksi Kedipan Mata

5.1.2.5 Perancangan Pemilihan Menu

Pada perancangan pemilihan menu sistem di desain menggunakan empat kotak pemilihan menu, pada kotak pertama dibuat untuk memanggil perawat ketika pasien membutuhkan bantuan perawat, pada kotak kedua digunakan untuk pemilihan menu makanan, pada kotak ketiga digunakan untuk pemilihan menu ketika pasien ingin buang air dan kotak ke empat digunakan untuk keluar dari

sistem menuju idle. Pada kotak pemilihan menu makanan, akan terdapat menu baru yaitu pilihan menu makanan. Kotak pertama berisi tulisan ayam, kotak kedua berisi tulisan *sea food*, kotak ketiga berisi tulisan mie dan kotak terakhir berisi menu untuk kembali ke menu awal. Untuk lebih jelaskan dapat dilihat pada *state machine* Gambar 5.9 dan untuk penjelasannya dapat dilihat pada Tabel 5.12.



Gambar 5.9 State Machine Pemilihan Menu

Tabel 5.12 Keterangan State Machine Pemilihan Menu

State	Event	Action
S0	Idle state	E0 Berkedip 3x A0 Masuk ke pemilihan menu 1
S1	Menu 1 (memanggil perawat)	E1 Melihat ke kanan A1 Pindah ke menu 2
S2	Menu 2 memilih makanan	E2 Melihat ke kiri A2 Pindah ke menu 1
S3	Menu 3 ingin buang air	E3 Melihat ke atas A3 Pindah ke menu 2
S4	Menu exit ke idle	E4 Melihat ke bawah A4 Pindah ke menu exit "idle"
S5	Menu makanan 1	E5 Melihat ke kanan A5 Pindah ke menu exit "idle"

S6	Menu makanan 2	E6	Melihat ke kiri	A6	Pindah ke menu 3
S7	Menu makanan 3	E7	Melihat ke kanan	A7	Pindah ke menu 1
S8	Menu exit ke menu utama	E8	Melihat ke bawah	A8	Pindah ke menu 3
S9	Suara A	E9	Berkedip	A9	Pindah ke idle state
S10	Suara B	E10	Berkedip	A10	Pindah ke menu makanan 1
S11	Menu makanan 1 ditebali	E11	Melihat ke kanan	A11	Pindah ke menu makanan 2
S12	Menu makanan 3 ditebali	E12	Melihat ke kiri	A12	Pindah ke menu makanan 1
S13	Menu makanan 2 ditebali	E13	Melihat ke atas	A13	Pindah ke menu makanan 2
		E14	Melihat ke bawah	A14	Pindah ke menu exit "menu utama"
		E15	Melihat ke kanan	A15	Pindah ke menu exit "menu utama"
		E16	Melihat ke kiri	A16	Pindah ke menu makanan 3
		E17	Melihat ke atas	A17	Pindah ke menu makanan 1
		E18	Melihat ke bawah	A18	Pindah ke menu makanan 3
		E19	Berkedip	A19	Pindah ke menu utama, menu 1
		E20	Menunggu hingga suara selesai	A20	Pindah ke menu 3
		E21	Berkedip	A21	Suara B berbunyi
		E22	Berkedip	A22	Suara A berbunyi
		E23	Menunggu hingga suara selesai	A23	Pindah ke menu 1

	E24	Menunggu 5 detik	A24	Pindah ke menu makanan 2
	E25	Berkedip	A25	Gambar menu makanan 3 terjadi penebalan selama 5 detik
	E26	Berkedip	A26	Gambar menu makanan 1 terjadi penebalan selama 5 detik
	E27	Menunggu 5 detik	A27	Pindah ke menu makanan 1
	E28	Berkedip	A28	Gambar menu makanan 3 terjadi penebalan selama 5 detik
	E29	Menunggu 5 detik	A29	Pindah ke menu makanan 3

5.2 Implementasi

Implementasi sistem adalah tahap untuk merealisasikan perancangan sistem yang sebelumnya telah dibuat. Pada bagian ini akan membahas mengenai implementasi perangkat keras dan implementasi perangkat lunak

5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras dibuat sesuai dengan perancangan pada bab sebelumnya. Implementasi perangkat keras akan terbagi menjadi implementasi alat dan implementasi komponen.

5.2.1.1 Implementasi Alat

Implementasi alat berisi mengenai bentuk jadi dari sistem tersebut. Implementasi alat meliputi pembuatan alat penyangga, peletakan sistem, peletakan sensor dan wadah dari sistem. Gambar 5.10 merupakan bentuk dari implementasi alat yang telah dibuat. Alat penyangga terbuat dari besi dan kayu sedangkan wadah sistem terbuat dari akrilik. Untuk peletakan sistem dan sensor dapat dilihat pada Gambar 5.10.



Gambar 5.10 Implementasi Alat

5.2.1.2 Implementasi Pemasangan Komponen

Implementasi komponen berisi mengenai bentuk jadi dari sistem dan komponen ketika telah dipasangkan. Gambar 5.11 merupakan bentuk dari implementasi pemasangan komponen. Pada komponen sensor kamera akan ditancapkan pada usb raspberry pi, komponen speaker akan ditancapkan pada jack raspberry pi sebagai output dan power dari speaker akan ditancapkan pada usb raspberry pi.



Gambar 5.11 Implementasi Pemasangan Komponen

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan penerapan dari flowchart dan state machine yang telah dibuat pada bagian perancangan perangkat lunak. Implementasi perangkat lunak berisi code program yang telah dibuat pada python ide.

5.2.2.1 Implementasi Library yang Digunakan

Pada implementasi library menggunakan beberapa library yang akan berfungsi untuk membantu dalam menyusun program sehingga dapat digunakan dengan lebih efektif. Tabel 5.13 merupakan isi library yang digunakan pada sistem.

Tabel 5.13 Kode Program Library Yang Digunakan

Library
1 import cv2
2 import numpy as np
3 import dlib
4 import imutils
5 import pygame

Penjelasan kode program :

1. Baris 1 : merupakan library yang digunakan untuk mengolah citra digital agar dapat di proses
2. Baris 2 : library ini digunakan untuk mengolah array
3. Baris 3 : fungsi dari library Dlib digunakan untuk mengolah metode *facial landmark*
4. Baris 4 : merupakan library yang digunakan untuk mengolah metode *facial landmark*
5. Baris 5 : library yang digunakan untuk mengeluarkan output suara

5.2.2.2 Implementasi Sistem Utama

Pada implementasi sistem utama berisi mengenai jalur program bekerja saat sistem mendapatkan input hingga mengeluarkan output secara garis besarnya. Sistem akan mendapatkan input pada gambar dan mengolah gambar tersebut hingga ditampilkan pada suatu layar. Pengguna akan memilih menu pada layar tersebut dan akan mengeluarkan output melalui suara. Tabel 5.14 berisi kode program dari cara kerja sistem utama.

Tabel 5.14 Kode Program Sistem Utama

Sistem Utama
1 #Program Utama
2 cap1 = cv2.VideoCapture(0)
3 keyboard = np.zeros((400, 400, 3), np.uint8)
4 widtha = 200
5 heighta = 200
6 th = 8
7 thmax=-1
8 kondisi1=1
9 def idle(x,y,let):
10 cv2.rectangle(keyboard,(x + th, y + th),(x + 400 - th,
11 y + 400 - th),(148,105,0),thmax)
12 font_letter = cv2.FONT_HERSHEY_PLAIN
13 font_scale = 10
14 font_th = 6
15 text_size = cv2.getTextSize(let, font_letter, font_scale,
font_th)[0]
width_text, height_text = text_size[0], text_size[1]

```

16 text_x = int((400 - width_text) / 2) + x
17 text_y = int((400 + height_text) / 2) + y
18 cv2.putText(keyboard, let, (text_x, text_y), font_letter,
font_scale, (0,0,0), font_th)
19 def get_blinking_ratio(eye_points, facial_landmarks):
20 #DETEKSI KEDIP
21 while True:
22 ret, frame = cap1.read()
23 if kondisi1==1:
24 cv2.rectangle(keyboard, (0,0), (400,400),
(255,255,255), thmax)
25 idle(0,0,"idle")
26 if counter_stotal >=1 :
27 increment+=1
28 if blinking_ratio > bawah and kondisi1==1:
29 counter_start +=1
30 else :
31 if counter_start >=3 :
32 counter_stotal +=1
33 counter_start=0
34 if counter_stotal ==3 :
35 if increment > 600 :
36 increment=0
37 counter_stotal=0;
38 elif increment <= 600 :
39 cv2.rectangle(keyboard, (0,0), (400,400),
(255,255,255), thmax)
40 #MENU UTAMA

```

Penjelasan :

1. Baris 2 : merupakan fungsi untuk mendapatkan input berupa video streaming
2. Baris 3 : merupakan fungsi untuk membuat suatu layar kosong dengan ukuran 400x400 pixel
3. Baris 4-7 : merupakan deklarasi variabel untuk membuat menu pada layar kosong
4. Baris 9-19 : suatu fungsi yang digunakan untuk membuat menu pada layar
5. Baris 21-22 : merupakan suatu fungsi untuk membaca input video streaming
6. Baris 23 – 38 : berisi suatu fungsi mendeteksi kedipan untuk mengubah dari menu idle hingga menampilkan menu utama
7. Baris 39 : bagian menu utama akan dideklarasikan

5.2.2.3 Implementasi Deteksi Pergerakan Mata

Implementasi deteksi pergerakan mata, berisi mengenai deteksi pergerakan mata ke kanan, ke kiri dan ke depan. Implementasi ini nantinya akan digunakan pada pemilihan menu pada implementasi pemilihan menu. Tabel 5.15 berisi kode program implementasi deteksi pergerakan mata.

Tabel 5.15 Kode Program Implementasi Deteksi Pergerakan Mata

Deteksi Pergerakan Mata	
1	#INISIALISASI

```

2 countkanan = 1
3 countkiri = 1
4 kanan = 0.6
5 kiri = 1.2
6
7 def nothing(x):
8     pass
9
10 cv2.namedWindow("Trackbars")
11 cv2.createTrackbar("Binary", "Trackbars", 0, 255, nothing)
12
13 def midpoint(p1,p2):
14     return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)
15
16 #VOID DETEKSI
17 def get_gaze_ratio(eye_points, facial_landmarks):
18     left_eye_region
19     np.array([(facial_landmarks.part(eye_points[0]).x,
20 facial_landmarks.part(eye_points[0]).y),
21 (facial_landmarks.part(eye_points[1]).x,
22 facial_landmarks.part(eye_points[1]).y),
23 (facial_landmarks.part(eye_points[2]).x,
24 facial_landmarks.part(eye_points[2]).y),
25 (facial_landmarks.part(eye_points[3]).x,
26 facial_landmarks.part(eye_points[3]).y),
27 (facial_landmarks.part(eye_points[4]).x,
28 facial_landmarks.part(eye_points[4]).y),
29 (facial_landmarks.part(eye_points[5]).x,
30 facial_landmarks.part(eye_points[5]).y)], np.int32)
31     height, width, _ = cap.shape
32     mask = np.zeros((height, width), np.uint8)
33     cv2.polylines(mask, [left_eye_region], True, 255, 2)
34     cv2.fillPoly(mask, [left_eye_region], 255)
35     eye = cv2.bitwise_and(gray, gray, mask=mask)
36
37     min_x = np.min(left_eye_region[:, 0])
38     max_x = np.max(left_eye_region[:, 0])
39     min_y = np.min(left_eye_region[:, 1])
40     max_y = np.max(left_eye_region[:, 1])
41
42     shape = eye[min_y: max_y, min_x: max_x]
43     h = cv2.getTrackbarPos("Binary", "Trackbars")
44     ret, threshold_eye
45     cv2.threshold(shape, h, 255, cv2.THRESH_BINARY)
46     cv2.imshow("eye", threshold_eye)
47     height, width = threshold_eye.shape
48     left_side_threshold = threshold_eye[0: height, 0:
49 int(width / 2)]
50     left_side_white = cv2.countNonZero(left_side_threshold)
51     right_side_threshold = threshold_eye[0: height, int(width
52 / 2): width]
53     right_side_white
54     cv2.countNonZero(right side threshold)

```

```

45
46 if left_side_white == 0:
47     gaze_ratio = 1
48 elif right_side_white == 0:
49     gaze_ratio = 5
50 else:
51     gaze_ratio = left_side_white / right_side_white
52 return gaze_ratio
53
54 #MAIN PROGRAM
55 while True:
56     #frame = cap
57     ret, frame = cap1.read()
58     width = 150
59     height = 100
60     cap = cv2.resize(frame, (width, height))
61     gray = cv2.cvtColor(cap, cv2.COLOR_BGR2GRAY)
62     faces = detector(gray, 0)
63     for face in faces:
64         landmarks = predictor(gray, face)
65         # Gaze detection
66         gaze_ratio_left_eye = get_gaze_ratio([36, 37, 38, 39,
40, 41], landmarks)
67         gaze_ratio_right_eye = get_gaze_ratio([42, 43, 44,
45, 46, 47], landmarks)
68         gaze_ratio = (gaze_ratio_right_eye +
69 gaze_ratio_left_eye) / 2
70
71 #OUTPUT DETEKSI PERGERAKAN
72 if gaze_ratio <= kanan:
73     countkanan +=1
74     countkiri = 0
75     if countkanan > 4 :
76         cv2.putText(frame, "right", (50, 100), font,
2, (0, 255, 0), 3)
77     elif kanan < gaze_ratio < kiri:
78         cv2.putText(frame, "center", (50, 100), font, 2,
(0, 255, 0), 3)
79         countkanan=0
80         countkiri=0
81     elif gaze_ratio >= kiri:
82         countkiri +=1
83         countkanan= 0
84         if countkiri > 4:
85             cv2.putText(frame, "left", (50, 100), font,
2, (0, 255, 0), 3)
86         cv2.imshow("vid", cap)
87         key = cv2.waitKey(30)
88         if key == 27:
89             break
90 cv2.destroyAllWindows()

```

Penjelasan kode program pada tabel 5.7 :

1. Baris 1-5 : deklarasi untuk menentukan nilai deteksi pergerakan mata
2. Baris 7 – 11 : deklarasi untuk membuat suatu trackbar yang digunakan untuk segmentasi biner

3. Baris 16-53 : berisi fungsi untuk mendeteksi bagian mata hingga pengolahan segmentasi biner dan perhitungan nilai pada pixel putih.
4. Baris 56-58 : fungsi untuk membaca input gambar
5. Baris 59-61 : fungsi untuk mengubah ukuran dari gambar
6. Baris 62-63 : fungsi untuk mensegmentasi gambar menjadi grayscale
7. Baris 64-70 : berisi fungsi untuk mengambil nilai *facial landmark* pada bagian mata dan perhitungan pixel pada bagian mata
8. Baris 73-86 : Berisi pendeteksian pergerakan mata menggunakan nilai gaze rasio sebagai acuannya
9. Baris 87-90 : Fungsi untuk menampilkan hasil deteksi pada suatu window

5.2.2.4 Implementasi Deteksi Kedipan Mata

Implementasi deteksi kedipan mata berisi mengenai deteksi pergerakan mata saat mata berkedip dan melihat ke atas. Implementasi deteksi kedipan mata digunakan pada bagian sistem utama untuk mengubah sistem ketika dalam keadaan idle hingga menampilkan menu utama dan digunakan pada bagian implementasi pemilihan menu. Tabel 5.16 berisi kode program untuk implementasi deteksi kedipan mata.

Tabel 5.16 Kode Program Implementasi Deteksi Kedipan Mata

Deteksi Kedipan Mata	
1	#INISIALISASI
2	atas = 2.6
3	bawah = 4.9
4	countkedip = 1
5	countatas = 1
6	
7	
8	def midpoint(p1 ,p2):
9	return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)
10	
11	#DETEKSI KEDIP
12	def get_blinking_ratio(eye_points, facial_landmarks):
13	left_point = (facial_landmarks.part(eye_points[0]).x,
14	facial_landmarks.part(eye_points[0]).y)
15	right_point = (facial_landmarks.part(eye_points[3]).x,
16	facial_landmarks.part(eye_points[3]).y)
17	center_top =
18	midpoint(facial_landmarks.part(eye_points[1]),
19	facial_landmarks.part(eye_points[2]))
20	center_bottom =
21	midpoint(facial_landmarks.part(eye_points[5]),
22	facial_landmarks.part(eye_points[4]))
23	hor_line_lenght = hypot((left_point[0] - right_point[0]),
24	(left_point[1] - right_point[1]))

```

21 ver_line_lenght = hypot((center_top[0] -
center_bottom[0]), (center_top[1] - center_bottom[1]))
22 ratio = hor_line_lenght / ver_line_lenght
23 return rasio
24
25 #MAIN PROGRAM
26 while True:
27     #frame = cap
28     ret, frame = cap1.read()
29     width = 150
30     height = 100
31     cap = cv2.resize(frame, (width, height))
32     gray = cv2.cvtColor(cap, cv2.COLOR_BGR2GRAY)
33     faces = detector(gray, 0)
34     for face in faces:
35         landmarks = predictor(gray, face)
36         # Detect blinking
37         left_eye_ratio = get_blinking_ratio([36, 37, 38, 39,
38 40, 41], landmarks)
39         right_eye_ratio = get_blinking_ratio([42, 43, 44, 45,
40 46, 47], landmarks)
41         blinking_ratio = (left_eye_ratio + right_eye_ratio) /
42 2
43         if blinking_ratio < atas:
44             countatas +=1
45             countkedip = 0
46             if countkedip > 3 :
47                 cv2.putText(frame, "up", (50, 100),
48 font, 2, (255, 0, 0),3)
49                 countatas = 0
50         elif blinking_ratio > bawah:
51             countkedip +=1
52             countatas =0
53             if countkedip > 3 :
54                 cv2.putText(frame, "down", (50, 100),
55 font, 2, (255, 0, 0),3)
56                 countkedip = 0
57         cv2.imshow("vid", cap)
58         key = cv2.waitKey(30)
59         if key == 27:
60             break
61     cv2.destroyAllWindows()

```

Penjelasan kode program pada Tabel 5.8 :

1. Baris 1-5 : deklarasi variabel yang menjadi parameter deteksi
2. Baris 8-9 : fungsi untuk mendapatkan titik tengah pada mata
3. Baris 12-24 : berisi fungsi untuk mendeteksi kedipan mata menggunakan perhitungan pixel pada garis vertikal dan horizontal
4. Baris 27-29 : fungsi untuk membaca input gambar pada video streaming
5. Baris 30-32 : fungsi untuk mengubah ukuran gambar
6. Baris 33-34 : fungsi untuk menyegmentasi gambar menjadi grayscale

7. Baris 35-40 : berisi fungsi untuk mengambil nilai *facial landmark* pada bagian mata dan perhitungan pixel pada garis vertikal dan horizontal
8. Baris 40-51 : berisi pendeteksian mata ketika mata berkedip atau melihat ke atas
9. Baris 53-57 : Fungsi untuk menampilkan hasil deteksi pada suatu window

5.2.2.5 Implementasi Pemilihan Menu

Implementasi pemilihan menu berisi mengenai kode program untuk mendesain atau membuat suatu pemilihan menu pada suatu layar window yang akan digunakan oleh pengguna untuk memilih kebutuhan dari pengguna tersebut yang akan dioutputkan melalui suara. Implementasi pemilihan menu akan digunakan untuk pemilihan menu yang diterapkan pada bagian sistem utama. Tabel 5.17 merupakan kode program yang berisi implementasi pemilihan menu.

Tabel 5.17 Kode Program Implementasi Pemilihan menu

Desain Menu	
1	#INISIALISASI
2	keyboard = np.zeros((400, 400, 3), np.uint8)
3	widtha = 200
4	heighta = 200
5	th = 8
6	thmax= -1
7	i=1
8	
9	count = 1
10	
11	countatas = 1
12	countkedip = 1
13	countkanan = 1
14	countkiri = 1
15	
16	counter_start = 0
17	counter_stotal=0
18	
19	kondisiidle=1
20	
21	keyaa=0
22	keybb=0
23	key2=0
24	
25	#INISIALISASI MENU
26	def letter1(x,y,let):
27	cv2.rectangle(keyboard, (x + th, y + th), (x + widtha -
28	th, y + heighta - th), (148,105,0), th)
29	cv2.rectangle(keyboard, (x + th, y + th), (x + widtha -
30	th, y + heighta - th), (255, 255, 255), thmax)
31	font_letter = cv2.FONT_HERSHEY_PLAIN
32	font_scale = 2
33	font_th = 1
34	text_size = cv2.getTextSize(let, font_letter,
	font_scale, font_th)[0]
	width_text, height_text = text_size[0], text_size[1]
	text_x = int((widtha - width_text) / 2) + x

```

35     text_y = int((heighta + height_text) / 2) + y
36     cv2.putText(keyboard, let, (text_x, text_y),
37                 font_letter, font_scale, (0,0,0), font_th)
38 def mainMenu(index,ok, letter_bold):
39     # Keys
40     if index == 1:
41         x = 0
42         y = 0
43         letter1(200, 0, "makan")
44         letter1(0, 200, "toilet")
45         letter1(200, 200, "exit")
46     elif index == 2:
47         x = 200
48         y = 0
49         letter1(0, 0, "perawat")
50         letter1(0, 200, "toilet")
51         letter1(200, 200, "exit")
52     elif index == 3:
53         x = 0
54         y = 200
55         letter1(0, 0, "perawat")
56         letter1(200, 0, "makan")
57         letter1(200, 200, "exit")
58     elif index == 4:
59         x = 200
60         y = 200
61         letter1(0, 0, "perawat")
62         letter1(200, 0, "makan")
63         letter1(0, 200, "toilet")
64
65     if letter_bold is True:
66
67         if ok is True:
68             cv2.rectangle(keyboard, (th + x, y + th), (x +
69 widtha - th, y + heighta - th), (0, 0, 0), th)
70             if ok is False:
71                 cv2.rectangle(keyboard, ( th + x, y + th), (x +
72 widtha - th, y + heighta - th), (255, 255, 0), th)
73                 #if letter_bold is False:
74                 #cv2.rectangle(keyboard, (x + th, y + th), (x +
75 widtha - th, y + heighta - th), (255, 255, 255), th)
76
77 def menuMakan(index,ok, letter_bold):
78     # Keys
79     if index == 1:
80         x = 0
81         y = 0
82         letter1(200, 0, "sea food")
83         letter1(0, 200, "mie")
84         letter1(200, 200, "exit")
85     elif index == 2:
86         x = 200
87         y = 0
88         letter1(0, 0, "ayamm")
89         letter1(0, 200, "mie")
90         letter1(200, 200, "exit")
91     elif index == 3:

```

```

90     x = 0
91     y = 200
92     letter1(0, 0, "ayamm")
93     letter1(200, 0, "sea food")
94
95     letter1(200, 200, "exit")
96     elif index == 4:
97         x = 200
98         y = 200
99         letter1(0, 0, "ayamm")
100        letter1(200, 0, "sea food")
101        letter1(0, 200, "mie")
102
103
104        if letter_bold is True:
105
106            if ok is True:
107                cv2.rectangle(keyboard, (th + x, y + th), (x +
108                    widtha - th, y + heighta - th), (0, 0, 0), th)
109                if ok is False:
110                    cv2.rectangle(keyboard, ( th + x, y + th), (x +
111                        widtha - th, y + heighta - th), (255, 255, 0), th)
112
113            while True:
114                #CODE DETEKSI KEDIP
115                #CODE DETEKSI PERGERAKAN MATA
116                #CODE MENU IDLE
117
118                #DESAIN MENU
119                if increment > 100 :
120                    increment=0
121                    counter_stotal=0;
122                elif increment <= 300 :
123                    #MENU UTAMA
124                    cv2.rectangle(keyboard, (0,0), (400,400),
125                        (255,255,255), thmax)
126                    letter1(0, 0, "perawat")
127                    letter1(200, 0, "makan")
128                    letter1(0, 200, "toilet")
129                    letter1(200, 200, "exit")
130                    kondisiidle = 0
131                    enter = False
132
133                if blinking_ratio < atas:
134                    countatas += 1
135                    count += 1
136                    countkedip = 0
137                    countkanan = 0
138                    countkiri = 0
139                    cv2.putText(frame, "up", (50, 100), font, 2, (255,
140                        0, 0), 3)
141                    if i == 3 and countatas > 3:
142                        i = 1
143                        countatas=0
144                        count = 0
145                    elif i == 4 and countatas > 3:
146                        i = 2
147                        countatas = 0
148                        count = 0

```

```

145     if count > 10 and (i == 1 or i == 2):
146         enter = True
147     elif count == 0:
148         enter = False
149
150     elif blinking_ratio > bawah:
151         countkedip += 1
152         count += 1
153         countatas = 0
154         countkanan = 0
155         countkiri = 0
156         cv2.putText(frame, "down", (50, 100), font, 2, (255,
157     0, 0), 3)
158         if i == 1 and countkedip > 3:
159             i = 3
160             countkedip = 0
161             count = 0
162         elif i == 2 and countkedip > 3:
163             i = 4
164             countkedip = 0
165             count = 0
166         if count > 10 and (i == 4 or i == 3):
167             enter = True
168         elif count == 0:
169             enter = False
170
171     elif gaze_ratio <= kanan:
172         countkanan += 1
173         countatas = 0
174         countkedip = 0
175         countkiri = 0
176         count = 0
177         cv2.putText(frame, "right", (50, 100), font, 2, (0,
178     255, 0), 3)
179         if i == 1 and countkanan > 3:
180             i = 2
181             countkanan = 0
182         elif i == 3 and countkanan > 3:
183             i = 4
184             countkanan = 0
185         if count == 0:
186             enter = False
187
188     elif kanan < gaze_ratio < kiri:
189         cv2.putText(frame, "center", (50, 100), font, 2, (0,
190     255, 0), 3)
191         countatas = 0
192         count = 0
193         kondisiidle = 0
194         if count == 0:
195             enter = False
196
197     elif gaze_ratio >= kiri:
198         countkiri += 1
199         countatas = 0
200         countkedip = 0
201         countkanan = 0
202         count = 0

```

```

201 cv2.putText(frame, "left", (50, 100), font, 2, (0,
202 255, 0), 3)
203 if i == 2 and countkiri > 3:
204     i = 1
205     countkiri = 0
206 elif i == 4 and countkiri > 3:
207     i = 3
208     countkiri = 0
209 if count == 0:
210     enter = False
211 #PEMILIHAN MENU UTAMA
212 if i == 1 and key2 == 0:
213     bold = True
214     if enter == True:
215         enter = False
216     elif i == 2 and key2 == 0:
217         bold = True
218         if enter == True:
219             enter = False
220             keyaa += 1
221     else:
222         if keyaa >= 1:
223             key2 = 1
224             i = 1
225         elif i == 3 and key2 == 0:
226             bold = True
227             if enter == True:
228                 enter = False
229         elif i == 4 and key2 == 0:
230             bold = True
231             if enter == True:
232                 enter = False
233                 counter_stotal = 0
234                 increment = 0
235                 kondisiidle = 1
236                 i = 1
237         if key2 == 0:
238             mainMenu(i, enter, bold)
239
240 #MENU MAKANAN
241 if key2==1 :
242     letter1(200, 200, "exit")
243     letter1(0, 0, "ayamm")
244     letter1(200, 0, "sea food")
245     letter1(0, 200, "mie")
246
247     if blinking_ratio < atas:
248         countatas +=1
249         count +=1
250         countkedip = 0
251         countkanan = 0
252         countkiri = 0
253     cv2.putText(frame, "up", (50, 100), font, 2, (255,
254 0, 0), 3)
255     if i == 3 and countatas > 3 :
256         i=1
257         countatas=0
258         count=0

```

```

258 elif i == 4 and countatas > 3 :
259     i=2
260     countatas=0
261     count=0
262 if count > 10 and (i==1 or i==2):
263     enter = True
264     elif count == 0 :
265         enter = False
266
267 elif blinking_ratio > bawah:
268     countkedip +=1
269     count+=1
270     countatas = 0
271     countkanan = 0
272     countkiri = 0
273     cv2.putText(frame, "down", (50, 100), font, 2, (255,
274         0, 0), 3)
275 if i == 1 and countkedip > 3 :
276     i=3
277     countkedip=0
278     count=0
279     elif i == 2 and countkedip > 3 :
280         i=4
281         countkedip=0
282         count=0
283 if count > 10 and (i==4 or i==3):
284     enter = True
285     elif count == 0 :
286         enter = False
287
288 elif gaze_ratio <= kanan:
289     countkanan +=1
290     countatas = 0
291     countkedip = 0
292     countkiri = 0
293     count =0
294     cv2.putText(frame, "right", (50, 100), font, 2, (0,
295         255, 0), 3)
296 if i == 1 and countkanan > 3 :
297     i=2
298     countkanan=0
299     elif i == 3 and countkanan > 3 :
300         i=4
301         countkanan=0
302 if count == 0 :
303     enter = False
304
305 elif kanan < gaze_ratio < kiri :
306     cv2.putText(frame, "center", (50, 100), font, 2, (0,
307         255, 0), 3)
308     countatas=0
309     count=0
310     kondisiidle=0
311 if count== 0 :
312     enter = False
313
314 elif gaze_ratio >= kiri:
315     countkiri +=1
316     countatas = 0

```

```

314 countkedip = 0
315 countkanan = 0
316 count=0
317 cv2.putText(frame, "left", (50, 100), font, 2, (0,
    255, 0), 3)
318 if i == 2 and countkiri>3:
319     i=1
320     countkiri=0
321 elif i == 4 and countkiri>3:
322     i=3
323     countkiri=0
324 if count == 0 :
325     enter = False
326
327 cv2.putText(frame, str(countatas), (50, 250), font, 2, (0, 0, 275),
    3)
328
329
330 #PEMILIHAN MENU MAKANAN
331 if i == 1 and key2==1:
332     bold = True
333 elif i == 2 and key2==1:
334     bold = True
335 elif i == 3 and key2==1:
336     bold = True
337 elif i == 4 and key2==1:
338     bold = True
339 if enter==True :
340     keybb+=1
341 else :
342     if keybb>=1:
343         key2=0
344         i=1
345         keyaa=0
346 if key2==1 :
347     menuMakan(i, enter, bold)
    
```

Penjelasan kode program pada Tabel 5.9 :

1. Baris 1-23 : berisi deklarasi variabel yang digunakan untuk desain menu pada layar, acuan pendeteksian dan acuan untuk memilih menu
2. Baris 25-36 : berisi fungsi untuk membuat desain menu, berisi ukuran pixel dan warna dari menu
3. Baris 38-72 : berisi fungsi untuk membuat menu utama
4. Baris 73-110 : berisi fungsi untuk membuat menu makanan
5. Baris 118 – 209 : berisi kondisi pemilihan menu utama berdasarkan pergerakan mata
6. Baris 211- 238 : berisi output ketika pengguna memilih menu berdasarkan kondisi pergerakan mata
7. Baris 241-328 : berisi kondisi pemilihan menu makan berdasarkan pergerakan mata
8. Baris 331 – 347 : berisi output ketika pengguna memilih menu makan berdasarkan kondisi pergerakan mata

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian untuk mengetes sistem apakah dapat bekerja dengan baik sesuai yang diinginkan. Pengujian akan dilakukan dengan berbagai kondisi, kondisi yang diujikan adalah tingkat pencahayaan, jarak antara kamera dengan pengguna dan kondisi latar belakang gambar.

6.1 Pengujian Keakurasian Dalam Pendeteksian *Facial landmark* Pada Kondisi Latar Belakang Ramai Dan Sepi

6.1.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk mengetahui tingkat keakurasian dalam mendeteksi muka menggunakan *facial landmark*. Hal ini bertujuan untuk mengetahui apakah objek selain wajah dapat terdeteksi oleh metode *facial landmark* atau tidak.

6.1.2 Prosedur Pengujian

Prosedur untuk melakukan percobaan agar dapat berjalan dengan baik adalah :

1. Menghubungkan perangkat webcam dan LCD ke raspberry pi, lalu nyalakan raspberry pi dengan menghubungkannya ke adaptor.
2. Menjalankan program yang terdapat pada raspberry pi melalui thonny ide atau dapat melalui terminal.
3. Melakukan percobaan dengan cara mengambil gambar wajah atau video menggunakan perangkat webcam pada aplikasi yang telah diprogram.
4. Memperhitungkan tingkat keakurasian dari sistem dalam mendeteksi wajah menggunakan *facial landmark*.

6.1.3 Pelaksanaan Pengujian

Pengujian dilakukan dengan 15 subjek orang dengan menampilkan wajah mereka pada webcam. Pada tahap ini pergerakan dari mata tidak diperhitungkan karena hanya mengetes akurasi deteksi *facial landmark* dalam mendeteksi bagian wajah manusia. Jarak dan pencahayaan diperoleh secara acak untuk mengetahui tingkat akurasi sistem. Percobaan dilakukan dua kali yaitu pada latar belakang yang sepi dan latar belakang yang ramai.

6.1.4 Hasil dan Analisis Pengujian

Berikut ini merupakan hasil dan analisis dari pengujian yang telah dilakukan. Tabel 6.1 merupakan hasil yang diperoleh dari pengujian. Selanjutnya analisis hasil pengujian akan dilakukan.

Tabel 6.1 Pengujian Deteksi Facial Landmark Pada Latar Belakang Ramai Dan Sepi

No	Subject	Latar Belakang	
		Sepi	Ramai
1	Subject 1	V	V
2	Subject 2	V	V
3	Subject 3	V	V
4	Subject 4	V	V
5	Subject 5	V	V
6	Subject 6	V	V
7	Subject 7	V	V
8	Subject 8	V	V
9	Subject 9	V	V
10	Subject 10	V	V
11	Subject 11	V	V
12	Subject 12	V	V
13	Subject 13	V	V
14	Subject 14	V	V
15	Subject 15	V	V
Rata – rata deteksi		100%	100%

Keterangan :

V : Terdeteksi

X : Tidak Terdeteksi

Berdasarkan hasil pengujian pada tabel 6.1 membuktikan bahwa semua orang yang dilakukan pendeteksian dapat dideteksi dengan hasil yang sempurna yaitu 100%. Pada Gambar 6.1 dapat disimpulkan pada pengujian latar belakang bagian (b) bahwa *facial landmark* hanya mendeteksi bagian wajah dari manusia saja, tidak mendeteksi benda lain yang bukan wajah manusia. Dari hasil tersebut dapat dikatakan bahwa *facial landmark* dapat berfungsi dengan baik untuk mendeteksi wajah manusia dan hal ini menunjukkan bahwa metode ini dapat digunakan untuk mendeteksi bagian mata dari manusia yang digunakan untuk memprediksi pergerakan mata.



Gambar 6.1 Deteksi *Facial landmark*. (a) Pada Latar Belakang Sepi. (b) Pada Latar Belakang Ramai.

Dari Gambar 6.1 dapat dibuktikan bahwa pendeteksian menggunakan background sepi dan ramai tidak akan berpengaruh pada proses pendeteksian. Pada background sepi wajah dapat dideteksi dengan sempurna. Sedangkan pada background ramai wajah juga dapat dideteksi tanpa mendeteksi benda lain yang bukan wajah. Proses pendeteksian wajah akan menghasilkan titik – titik yang nantinya akan digunakan untuk melakukan proses cropping pada bagian mata. Pada Gambar 6.1 hasil proses deteksi ditunjukkan pada titik – titik hitam di area sekitar wajah.

6.2 Pengujian Keakurasian Dalam Pendeteksian Pergerakan Mata Pada Pencahayaan Terang (lux 50-400) dan Redup (lux 1-49)

6.2.1 Tujuan Pengujian

Pengujian dilakukan untuk mengetahui tingkat keakurasian dalam mendeteksi pergerakan mata pada keadaan terang dan redup. Hal ini dikarenakan program menggunakan segmentasi untuk melakukan pendeteksian mata.

6.2.2 Prosedur Pengujian

Prosedur pengujian yang harus dilakukan agar sistem dapat bekerja dengan baik adalah :

1. Menghubungkan perangkat webcam dan LCD ke raspberry pi, lalu nyalakan raspberry pi dengan menghubungkannya ke adaptor.
2. Mengambil data uji dengan perangkat dan mengukur pencahayaan menggunakan sensor lux.
3. Menjalankan program yang terdapat pada raspberry pi melalui thonny ide atau dapat melalui terminal.
4. Melakukan percobaan dengan cara mengambil gambar wajah atau video menggunakan perangkat webcam pada aplikasi yang telah diprogram.
5. Memperhitungkan tingkat keakurasian dari sistem dalam mendeteksi pergerakan mata.

6.2.3 Pelaksanaan Pengujian

Pengujian dilakukan dengan cara mengambil data gambar orang dengan posisi mata yang berbeda – beda yaitu ketika mata melihat ke kiri, ke kanan, ke depan, ke atas dan ke bawah. Pengujian ini dilakukan dengan 15 orang yang berbeda dengan mengujikan pada pencahayaan yang terang dan redup serta jarak antara subjek dengan kamera yaitu 30cm, 40cm dan 50cm.

6.2.4 Hasil dan Analisis Pengujian

Berikut ini merupakan hasil dan analisis dari pengujian yang telah dilakukan. Tabel 6.2 merupakan hasil yang diperoleh dari pengujian. Selanjutnya analisis hasil pengujian akan dilakukan. Analisis hasil pengujian akan dilakukan untuk mencari akurasi dari sistem dan proses pendeteksian pergerakan mata.

Tabel 6.2 Akurasi Pengujian Terhadap Cahaya Terang dan Redup

No.	Cahaya (lux)	Gerak Mata	Jarak	Subjek															Rata - rata Akurasi Gerak	Rata - rata Akurasi Cahaya
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	Terang (50 - 400)	Kiri	30	v	x	x	v	v	v	v	v	v	v	v	v	v	x	82%	91%	
2			40	v	x	v	v	v	v	v	v	v	v	v	v	v	v			
3			50	v	x	x	v	v	x	v	v	v	v	v	x	v	v			
4		Kanan	30	v	x	v	v	v	v	v	v	v	v	v	v	v	v	95%		
5			40	v	v	v	x	v	v	v	v	v	v	v	v	v	v			
6			50	v	v	v	v	v	v	v	v	v	v	v	v	v	v			v
7		Atas	30	v	v	v	v	v	x	v	v	v	v	v	v	v	v	93%		
8			40	v	v	v	x	v	v	v	v	v	v	v	v	v	v			
9			50	v	v	v	v	v	x	v	v	v	v	v	v	v	v			v
10		Bawah	30	v	v	v	v	v	v	v	v	v	v	v	v	v	v	97%		
11			40	v	v	v	v	v	v	v	v	v	v	v	v	v	v			
12			50	x	v	v	v	v	v	v	v	v	v	v	v	v	v			v
13		Depan	30	v	x	x	v	v	v	v	v	v	v	v	v	v	v	88%		
14			40	v	x	x	v	v	v	v	v	v	v	v	v	v	v			
15			50	x	v	v	v	v	v	v	v	v	v	v	v	v	v			v
16	Redup (1 - 49)	Kiri	30	v	x	x	v	v	v	v	v	v	x	v	x	v	v	82%		
17			40	v	x	v	v	v	v	v	v	v	x	v	v	v	v			
18			50	v	v	x	v	v	v	x	v	v	v	x	v	v	v		v	
19		Kanan	30	v	v	v	v	v	v	x	v	v	v	v	v	v	v	95%		
20			40	v	v	x	v	v	v	v	v	v	v	v	v	v	v			
21			50	v	v	v	v	v	v	v	x	v	v	v	v	v	v		v	
22		Atas	30	v	v	v	x	v	x	v	x	v	v	v	v	v	v	75%		
23			40	v	v	v	x	v	x	v	x	v	v	v	v	v	v			
24			50	v	v	v	x	v	x	v	x	v	v	v	v	x	x		v	
25		Bawah	30	v	v	v	v	v	v	v	v	v	v	v	v	v	v	97%		
26			40	v	v	v	v	v	v	v	v	v	v	v	v	v	v			
27			50	v	v	v	v	v	v	v	v	v	v	v	v	v	x		x	
28		Depan	30	v	v	x	v	v	v	v	v	v	v	v	x	v	v	86%		
29			40	v	v	x	v	v	v	v	v	v	v	v	x	v	v			
30			50	v	v	x	v	v	v	v	v	v	v	v	x	v	v		v	
Rata - rata Akurasi Total																	89%			

Keterangan :

V = Deteksi Berhasil

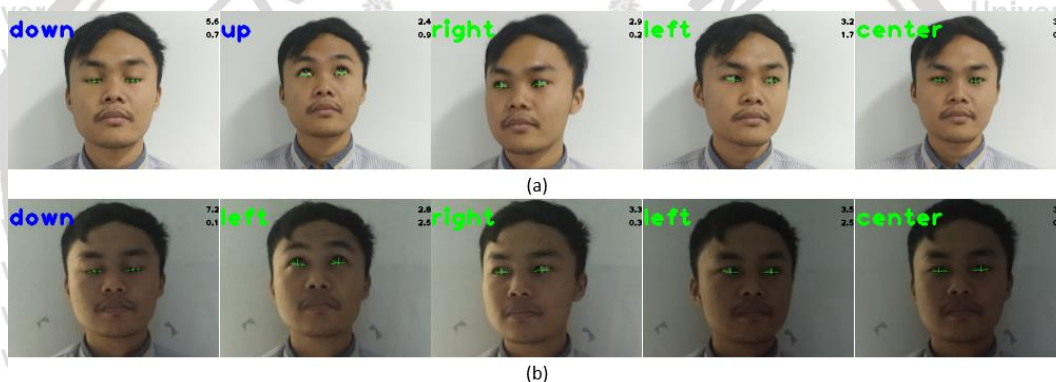
X = Deteksi Gagal

Dari hasil pengujian pada Tabel 6.2 dapat dibuktikan bahwa akurasi terbaik dalam mendeteksi yaitu terdapat pada cahaya terang yang memiliki lux berkisar antara 50 hingga 400. Pada pencahayaan terang mendapatkan akurasi yang tinggi yaitu 91% sedangkan pada pencahayaan redup mendapatkan akurasi 87%. Pergerakan mata dengan akurasi terbaik terdapat pada pergerakan melihat ke bawah dengan akurasi 97% pada pencahayaan terang dan gelap sedangkan pergerakan mata dengan akurasi terendah terdapat pada pergerakan melihat ke atas pada pencahayaan gelap dengan akurasi 75%. Berikut ini merupakan urutan posisi dari deteksi mulai dari yang terbaik hingga terendah :

1. Pergerakan ke bawah pada cahaya terang dengan akurasi 97%
2. Pergerakan ke bawah pada cahaya gelap dengan akurasi 97%
3. Pergerakan ke kanan pada cahaya terang dengan akurasi 95%
4. Pergerakan ke kanan pada cahaya gelap dengan akurasi 95%
5. Pergerakan ke atas pada cahaya terang dengan akurasi 93%

6. Pergerakan ke depan pada cahaya terang dengan akurasi 88%
7. Pergerakan ke depan pada cahaya gelap dengan akurasi 86%
8. Pergerakan ke kiri pada cahaya terang dengan akurasi 82%
9. Pergerakan ke kiri pada cahaya gelap dengan akurasi 82%
10. Pergerakan ke atas pada cahaya gelap dengan akurasi 75%

Pada proses deteksi terdapat bagian ketika gambar diproses dengan segmentasi. Hal tersebut sangat berpengaruh terhadap pencahayaan di sekitar karena ketika gambar memiliki cahaya yang redup maka proses segmentasi juga akan berdampak buruk dalam pendeteksiannya. Proses deteksi wajah menggunakan *facial landmark* juga akan berpengaruh ketika cahaya memiliki lux yang rendah seperti pada Gambar 6.2 dapat terlihat pada gambar bagian (b) ketika subjek melihat ke atas, sistem mendeteksi subjek melihat ke kiri, hal merupakan contoh proses deteksi yang gagal karena memiliki cahaya yang rendah sedangkan pada gambar kanan merupakan contoh proses deteksi yang berhasil karena memiliki cahaya yang mencukupi.



Gambar 6.2 Pengujian Pada Pencahayaan Terang dan Redup. (a) Pengujian Pada Pencahayaan Terang. (b) Pengujian Pada Pencahayaan Gelap

Pada Gambar 6.3 bagian atas kanan terdapat dua angka yang merupakan nilai dari deteksi. Pada kanan gambar angka tersebut memiliki nilai 3.1 dan 0.3 yang memiliki arti nilai yaitu nilai deteksi kedipan adalah 3.1 menunjukkan mata tidak melihat ke atas dan tidak berkedip, lalu deteksi pergerakan mata adalah 0.3 menunjukkan mata melihat ke kanan. Sedangkan pada kondisi gelap nilai tersebut rentan terganggu seperti pada gambar 6.3 bagian kiri, yang seharusnya nilai deteksi kedipan berkisar di antara 2,6 hingga 4,9 namun terdeteksi 2.5 yang berarti mata tersebut melihat ke atas. Untuk mengetahui deteksi pergerakan mata dan deteksi kedipan mata, nilai dari deteksi tersebut dapat digunakan sebagai acuan.



Gambar 6.3 Nilai Deteksi

6.3 Pengujian Keakurasian Dalam Pendeteksian Pergerakan Mata Pada Jarak 30cm, 40cm dan 50cm antara wajah dengan kamera

6.3.1 Tujuan Pengujian

Pengujian dilakukan untuk mengetahui tingkat keakurasian dalam mendeteksi pergerakan mata pada kondisi jarak kamera dengan wajah adalah 30cm, 40cm dan 50cm. Pengujian dilakukan untuk mengetahui apakah terdapat perbedaan akurasi ketika jarak wajah dengan kamera 30cm, 40cm dan 50cm.

6.3.2 Prosedur Pengujian

Prosedur pengujian yang harus dilakukan agar sistem dapat bekerja dengan baik adalah :

1. Menghubungkan perangkat webcam dan LCD ke raspberry pi, lalu nyalakan raspberry pi dengan menghubungkannya ke adaptor.
2. Mengambil data uji dengan menggunakan perangkat dan mengukur jaraknya menggunakan pengaris.
3. Menjalankan program yang terdapat pada raspberry pi melalui Thonny ide atau dapat melalui terminal.
4. Melakukan percobaan dengan cara mengambil gambar wajah atau video menggunakan perangkat webcam pada aplikasi yang telah diprogram.
5. Memperhitungkan tingkat keakurasian dari sistem dalam mendeteksi pergerakan mata.

6.3.3 Pelaksanaan Pengujian

Pengujian dilakukan dengan cara mengambil data gambar orang dengan posisi mata yang berbeda – beda yaitu ketika mata melihat ke kiri, ke kanan, ke depan, ke atas dan ke bawah. Pengujian ini dilakukan dengan 15 orang yang berbeda dengan mengujikan pada pencahayaan yang terang dan redup serta jarak antara subjek dengan kamera yaitu 30cm, 40cm dan 50cm.

6.3.4 Hasil dan Analisis Pengujian

Berikut ini merupakan hasil dan analisis dari pengujian yang telah dilakukan. Tabel 6.2 merupakan hasil yang diperoleh dari pengujian. Analisis hasil pengujian akan dilakukan untuk mencari akurasi dari sistem dan proses pendeteksian pergerakan mata.

Tabel 6.3 Akurasi Pengujian Terhadap Jarak 30cm, 40cm dan 50cm

No.	Jarak	Gerak Mata	Cahaya (lux)	Subjek															Rata - rata Akurasi Gerak	Rata - rata Akurasi Jarak
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	30	Kiri	Terang	v	x	x	v	v	v	v	v	v	v	v	v	v	v	x	80%	90%
2			Redup	v	x	x	v	v	v	v	v	v	v	x	v	x	v	v		
3		Kanan	Terang	v	x	v	v	v	v	v	v	v	v	v	v	v	v	v	96%	
4			Redup	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v		
5		Atas	Terang	v	v	v	v	v	v	x	v	v	v	v	v	v	v	v	86%	
6			Redup	v	v	v	x	v	x	v	x	v	v	v	v	v	v	v		
7		Bawah	Terang	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	100%	
8			Redup	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v		
9		Depan	Terang	v	x	x	v	v	v	v	v	v	v	v	v	v	v	v	86%	
10			Redup	v	v	x	v	v	v	v	v	v	v	v	v	x	v	v		
11	40	Kiri	Terang	v	x	v	v	v	v	v	v	v	v	v	v	v	v	v	90%	91%
12			Redup	v	x	v	v	v	v	v	v	v	v	x	v	v	v	v		
13		Kanan	Terang	v	v	v	x	v	v	v	v	v	v	v	v	v	v	v	93%	
14			Redup	v	v	x	v	v	v	v	v	v	v	v	v	v	v	v		
15		Atas	Terang	v	v	v	x	v	v	v	v	v	v	v	v	v	v	v	86%	
16			Redup	v	v	v	x	v	x	v	x	v	v	v	v	v	v	v		
17		Bawah	Terang	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	100%	
18			Redup	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v		
19		Depan	Terang	v	x	x	v	v	v	v	v	v	v	v	v	v	v	v	86%	
20			Redup	v	v	x	v	v	v	v	v	v	v	v	v	x	v	v		
21	50	Kiri	Terang	v	x	x	v	v	x	v	v	v	v	v	v	x	v	v	76%	87%
22			Redup	v	v	x	v	v	v	x	v	v	v	x	v	v	v	v		
23		Kanan	Terang	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	96%	
24			Redup	v	v	v	v	v	v	v	v	x	v	v	v	v	v	v		
25		Atas	Terang	v	v	v	v	v	x	v	v	v	v	v	v	v	v	v	80%	
26			Redup	v	v	v	x	v	x	v	x	v	v	v	v	x	x	v		
27		Bawah	Terang	x	v	v	v	v	v	v	v	v	v	v	v	v	v	v	93%	
28			Redup	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x		
29		Depan	Terang	x	v	v	v	v	v	v	v	v	v	v	v	v	v	v	90%	
30			Redup	v	v	x	v	v	v	v	v	v	v	v	v	x	v	v		
Rata - rata Akurasi Total																			89%	

Keterangan :

V = Deteksi Berhasil

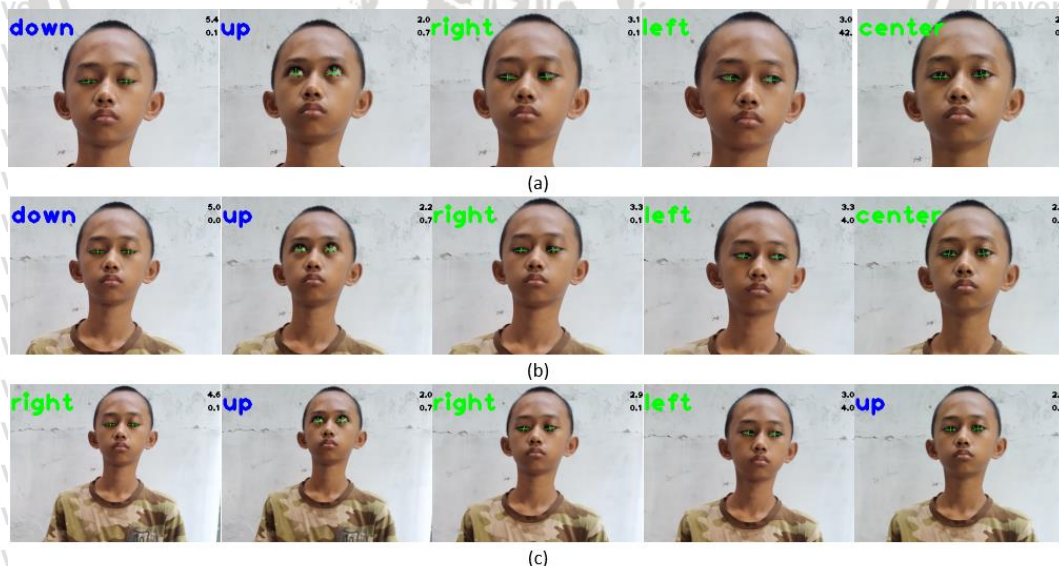
X = Deteksi Gagal

Dari hasil pengujian pada Tabel 6.3 dapat dibuktikan bahwa pengujian terbaik terdapat pada jarak 40cm yang mendapat akurasi sebesar 91% sedangkan pada jarak 30cm mendapat akurasi 90% dan pada jarak 50cm mendapat akurasi 87%. Pada akurasi pergerakan mata terbaik terdapat pada gerakan mata ke bawah pada jarak 30cm dan 40cm yang mendapatkan akurasi sebesar 100% sedangkan pergerakan mata dengan akurasi terburuk terdapat pada pergerakan mata ke kiri pada jarak 50cm dengan akurasi yang didapat 76%. Berikut ini merupakan urutan akurasi dari yang tertinggi hingga terendah berdasarkan jarak :

1. Pergerakan ke bawah pada jarak 40cm dengan akurasi 100%
2. Pergerakan ke bawah pada jarak 30cm dengan akurasi 100%
3. Pergerakan ke kanan pada jarak 30cm dengan akurasi 96%

4. Pergerakan ke kanan pada jarak 50cm dengan akurasi 96%
5. Pergerakan ke kanan pada jarak 40cm dengan akurasi 93%
6. Pergerakan ke bawah pada jarak 50cm dengan akurasi 93%
7. Pergerakan ke depan pada jarak 50cm dengan akurasi 90%
8. Pergerakan ke kiri pada jarak 40cm dengan akurasi 90%
9. Pergerakan ke atas pada jarak 30cm dengan akurasi 86%
10. Pergerakan ke depan pada jarak 30cm dengan akurasi 100%
11. Pergerakan ke atas pada jarak 40cm dengan akurasi 86%
12. Pergerakan ke depan pada jarak 40cm dengan akurasi 86%
13. Pergerakan ke atas pada jarak 50cm dengan akurasi 80%
14. Pergerakan ke kiri pada jarak 50cm dengan akurasi 80%
15. Pergerakan ke kiri pada jarak 50cm dengan akurasi 76%

Perbedaan akurasi ini dipengaruhi oleh jarak karena pada jarak yang jauh yaitu 50cm akan berbeda jumlah pixelnya dengan jarak yang 30cm dan 40cm. Seperti pada Gambar 6.4 bagian (a) dan (b) deteksi berhasil dilakukan sedangkan pada bagian (c) terdapat beberapa deteksi yang gagal dilakukan. Pada Gambar 6.4 pixel gambar mata pada bagian (a) dan (b) akan lebih besar jika dibandingkan gambar pada bagian (b), hal ini dapat mempengaruhi proses deteksi. Semakin besar pixel yang dideteksi maka semakin bagus juga proses pendeteksian pergerakan mata. Maka dari itu jarak ideal untuk mendeteksi pergerakan mata terdapat pada jarak diantara 30cm hingga 40cm.



Gambar 6.4 Pengujian Deteksi Pada Jarak 30cm dan 50cm. (a) Pengujian Pada Jarak 30cm. (b) Pengujian Pada Jarak 40cm. (c) Pengujian Pada Jarak 50cm.

6.4 Pengujian Kinerja Integrasi Pergerakan Mata Yang Telah Diterapkan Pada Pemilihan Menu Pada Layar LCD

6.4.1 Tujuan Pengujian

Pengujian dilakukan untuk mengetahui kinerja integrasi pergerakan mata terhadap pemilihan menu pada layar LCD.

6.4.2 Prosedur Pengujian

Prosedur pengujian yang harus dilakukan agar sistem dapat bekerja dengan baik adalah :

1. Menghubungkan perangkat webcam, speaker dan LCD ke raspberry pi, lalu nyalakan raspberry pi dengan menghubungkannya ke adaptor.
2. Menjalankan program yang terdapat pada raspberry pi melalui thonny ide atau dapat melalui terminal.
3. Melakukan percobaan dengan cara mengambil gambar wajah atau video menggunakan perangkat webcam pada aplikasi yang telah diprogram.
4. Memperhitungkan tingkat integrasi dari sistem dalam mendeteksi pergerakan mata yang diterapkan pada pemilihan menu.

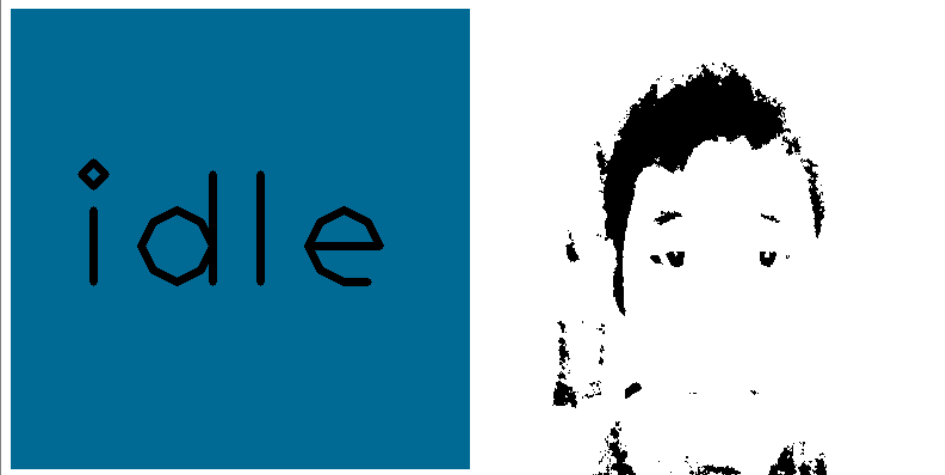
6.4.3 Pelaksanaan Pengujian

Pengujian dilakukan dengan cara melakukan pergerakan mata terhadap sistem pemilihan menu. Subjek akan diarahkan untuk melakukan pemilihan pada setiap menu. Sistem terdapat 4 pilihan menu dengan output yang berbeda beda, oleh karena itu subjek akan diarahkan untuk memilih 4 menu tersebut secara bergantian.

6.4.4 Hasil dan Analisis Pengujian

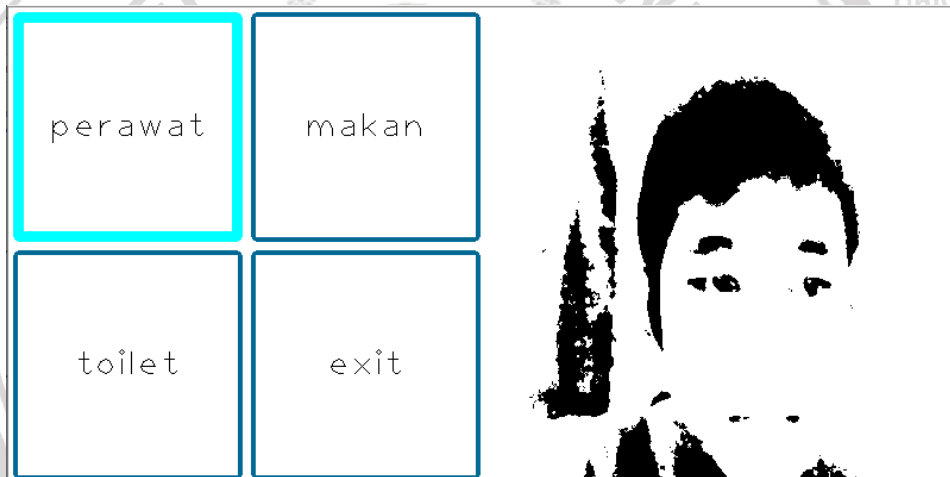
Berikut ini merupakan hasil dan analisis dari pengujian yang telah dilakukan :

1. Pada menu awal akan menampilkan tulisan idle seperti pada Gambar 6.5. Untuk dapat masuk ke menu utama, pengguna dapat mengedipkan mata sebanyak 3 kali.



Gambar 6.5 Menu Dalam Keadaan Idle

2. Menu utama akan muncul seperti pada Gambar 6.6. Untuk dapat memilih menu memanggil perawat, pengguna dapat mengarahkan matanya ke atas selama 3 detik.



Gambar 6.6 Tampilan Menu Utama

3. Ketika menu perawat dipilih maka sistem akan mengeluarkan output suara memanggil perawat seperti pada Gambar 6.7.



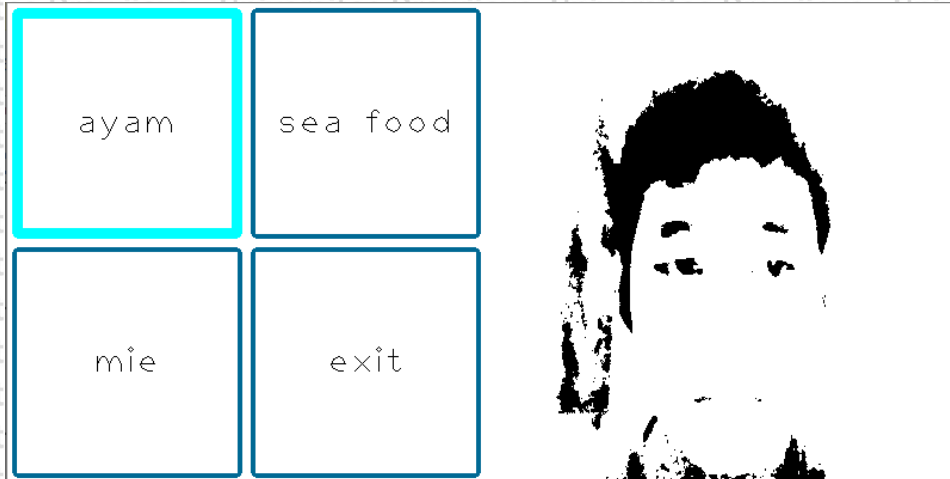
Gambar 6.7 Memilih Menu Perawat

4. Untuk mengarahkan ke menu makanan maka pengguna dapat melihat ke kanan seperti pada Gambar 6.8. Dan untuk memilih menu makanan, pengguna dapat mengarahkan mata ke atas selama 3 detik.



Gambar 6.8 Mengganti Pilihan Menu

5. Ketika memilih menu makanan maka menu baru yaitu menu makanan akan muncul seperti pada Gambar 6.9. Pengguna dapat mengarahkan matanya untuk mengganti pilihan menu makanan seperti ke kiri, ke kanan, ke atas dan ke bawah. Untuk memilih menu makanan ayam dan seafood pengguna dapat mengarahkan matanya ke atas selama 3 detik, dan untuk memilih menu mie pengguna dapat mengarahkan matanya ke bawah selama 3 detik. Menu exit berguna untuk kembali ke menu sebelumnya.



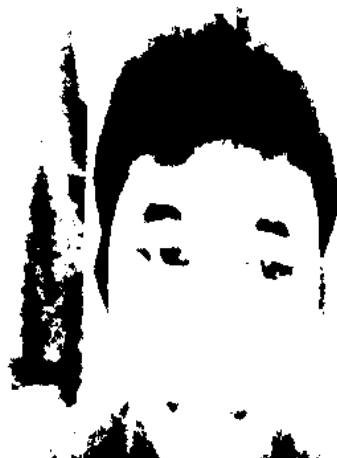
Gambar 6.9 Tampilan Menu Makanan

6. Ketika memilih menu toilet maka output seperti pada Gambar 6.10 yang akan muncul adalah suara memanggil perawat yang akan memberi pesan kepada perawat bahwa pengguna ingin ke toilet. Suara yang dikeluarkan menu toilet dan menu perawat akan berbeda

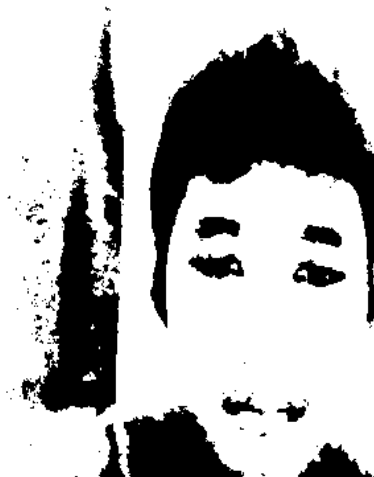
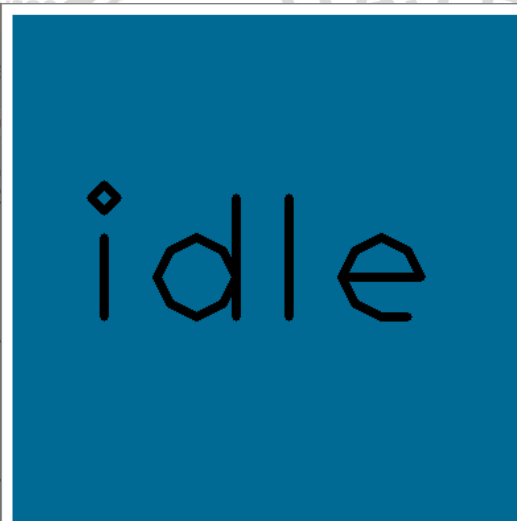


Gambar 6.10 Memilih Menu Toilet

7. Menu exit berfungsi untuk kembali ke kondisi idle, ketika menu tersebut dipilih maka sistem akan kembali ke kondisi idle seperti pada Gambar 6.11 ketika memilih menu idle dan Gambar 6.12 ketika menu kembali ke kondisi idle.



Gambar 6.11 Memindah Menu ke Exit



Gambar 6.12 Memilih Menu Exit

Dari hasil pengujian tersebut dapat disimpulkan bahwa sistem pendeteksian mata dapat bekerja dengan baik ketika di integrasikan dengan pemilihan menu. Sistem dapat mendeteksi mata dengan baik dan pemilihan menu dapat berganti posisi sesuai dengan pergerakan mata yang telah dilakukan pengguna. Pemilihan menu juga dapat mengeluarkan output sesuai dengan yang dipilih pengguna.

Dari analisis tersebut dilakukan delapan percobaan dengan memilih berbagai menu pada menu display. Hasil dari percobaan tersebut dapat dilihat pada Tabel 6.4 dan didapatkan akurasi sebesar 87,5%. Terdapat satu percobaan yang gagal pada saat memilih menu makanan ketika subjek melihat ke atas dideteksi oleh sistem subjek melihat ke kiri.

Tabel 6.4 Hasil Percobaan Integrasi Menu Display

Percobaan	Pergerakan Mata	Integrasi Menu Display		Hasil Deteksi		Akurasi
		Sebelum Eksekusi	Hasil Eksekusi	Benar	Salah	
1	Kedip 3 kali	idle	perawat makan toilet exit	v		87,5%
2	Lihat ke atas	perawat makan toilet exit	Suara memanggil perawat	v		
3	Lihat ke kanan	perawat makan toilet exit	perawat makan toilet exit	v		
4	Lihat ke atas	perawat makan toilet exit	perawat makan toilet exit		v	
5	Lihat ke bawah	perawat makan toilet exit	perawat makan toilet exit	v		
6	Lihat ke bawah	perawat makan toilet exit	Suara ingin ke toilet	v		
7	Lihat ke kiri	perawat makan toilet exit	perawat makan toilet exit	v		
8	Lihat ke bawah	perawat makan toilet exit	idle	v		

BAB 7 KESIMPULAN DAN SARAN

Pada bab ini akan membahas mengenai kesimpulan yang telah dilakukan pada bab sebelumnya. Kesimpulan tersebut mencakup akurasi kinerja dari sistem pendeteksian mata, sistem pemilihan menu dan output dari sistem. Saran dituliskan untuk memudahkan peneliti selanjutnya ketika ingin mengembangkan sistem agar lebih efektif, efisien dan mencapai akurasi yang tinggi sehingga dapat bermanfaat bagi masyarakat.

7.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan didapatkan beberapa kesimpulan sebagai berikut :

1. Sistem menggunakan metode *facial landmark* sebagai pendeteksi wajah manusia dan bagian mata dari wajah tersebut. *Facial landmark* hanya mendeteksi bagian wajah manusia dan tidak mendeteksi benda lainnya. Pada pengujian *facial landmark* didapatkan akurasi sebesar 100% pada background ramai dan sepi
2. Pada pendeteksian pergerakan mata dilakukan pengujian pada pencahayaan terang dengan lux 50-400 dan pencahayaan redup pada lux 1-49. Hasil dari pengujian didapatkan akurasi 91% pada kondisi pencahayaan terang dan 87% pada kondisi pencahayaan redup. Deteksi terbaik terdapat pada mata ketika melihat ke bawah dalam pencahayaan terang mendapat akurasi sebesar 97% dan deteksi dengan akurasi terendah terdapat pada mata ketika melihat ke atas dalam pencahayaan redup mendapat akurasi sebesar 75%
3. Pada pengujian jarak dilakukan tiga kali yaitu dengan jarak 30cm, 40cm dan 50cm. Akurasi terbaik terdapat pada jarak 40cm dengan nilai 91% lalu pada jarak 30cm akurasi yang didapat yaitu 90% dan pada jarak 50cm mendapat akurasi 87%. Deteksi dengan akurasi terbaik terdapat saat mata melihat ke bawah pada jarak 30cm dan 40cm mendapat akurasi sebesar 100% sedangkan akurasi yang paling rendah terdapat pada pergerakan mata ke kiri pada jarak 50cm mendapat akurasi 76%.
4. Ketika sistem diintegrasikan pada pemilihan menu, sistem dapat bekerja dengan baik dan mendapatkan akurasi 87,5%. Saat pengguna menggerakkan matanya sistem dapat bergerak sesuai dengan gerakan mata pengguna. Saat pengguna ingin memilih menu pengguna dapat mengedipkan atau mengarahkan matanya ke atas dan sistem akan mengeluarkan output berupa suara.

7.2 Saran

Metode ini menggunakan *facial landmark* dan segmentasi yang digunakan untuk mengolah citra. Hal ini akan berdampak pada cahaya, ketika pencahayaan yang digunakan sangat redup maka pendeteksian akan memiliki akurasi yang

rendah. Untuk penelitian selanjutnya diharapkan menggunakan metode lain selain segmentasi untuk mengolah citra, Hal ini akan berdampak pada akurasi sistem deteksi pergerakan mata.



DAFTAR REFERENSI

Alyushin, M.V., V.M. Alyushin, dan L.V. Kolobashkina. 2018. "Optimization of the Data Representation Integrated Form in the Viola-Jones Algorithm for a Person's Face Search." *Procedia Computer Science* 18-23.

Cadilhac, Dominique A., Monique F. Kilkenney, Natasha A. Lannin, Helen M. Dewey, Christopher R. Levi, Kelvin Hill, Brenda Grabsch, et al. 2019. "Outcomes for Patients With In-Hospital Stroke: A Multicenter Study From the Australian Stroke Clinical Registry (AuSCR)." *Journal of Stroke and Cerebrovascular Diseases* 1302 - 1310.

Chen, Ququ, dan Lei Sang. 2018. "Face-mask recognition for fraud prevention using Gaussian mixture model." *Journal of Visual Communication and Image Representation* 795 - 801.

Dinalankara, Lahiru. 2017. "Face Detection & Face Recognition Using Open Computer Vision Classifies." 1-7.

Dixit, Mahendra M., dan C. Vijaya. 2019. "Modelling and hardware implementation of quantization levels of digital cameras in DCT based image compression." *Engineering Science and Technology, an International Journal* 840 - 853.

Dwivedi, Divyansh. 2018. *Face Detection For Beginer*. 28 April. <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>.

Filimonau, Viachaslau, Christian Lemmer, David Marshall, dan Gisel Bejjani. 2017. "Restaurant menu re-design as a facilitator of more responsible consumer choice: An exploratory and preliminary study." *Journal of Hospitality and Tourism Management* 73-81.

freehanddesignprint. 2013. *freehanddesignprint*. 15 Oktober. Diakses Januari 14, 2020. www.freehanddesignprint.com.

gnovice. 2013. *stackoverflow*. 1 April. Diakses Januari 14, 2020. <https://stackoverflow.com/>.

Han, Shoudong, Ziqing Yang, Qianqian Li, dan Yang Chen. 2019. "Deformed landmark fitting for sequential faces." *Journal of Visual Communication and Image Representation* 381 - 393.

Hashemzadeh, Mahdi, Bahareh Asheghi, dan Nacer Farajzadeh. 2019. "Content-aware image resizing: An improved and shadow-preserving seam carving method." *Signal Processing* 233-246.

Irfan, Muhammad, Bakhtiar Alldino Ardi Sumbodo, dan Ika Candradewi. 2017. "Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra." *Indonesian Journal of Electronics and Instrumentation Systems* 139-148.

LinkedIn. 2003. *Slideshare*. 5 mei. Diakses Januari 14, 2020. www.slideshare.net.

Liput, Gusti Pandi. 2019. *gustinerz*. 8 Desember. Diakses Januari 14, 2020. www.gustinerz.com.

Logitech. 1981. *Logitech*. 2 Oktober. Diakses Januari 14, 2020. <https://www.logitech.com/id-id/video/webcams?filters=consumer>.

Luo, Deliang, Nengwu Zhu, Yao Li, Jiaying Cui, Pingxiao Wu, dan Jieyuan Wang. 2019. "Simultaneous leaching and extraction of indium from waste LCDs with acidic ionic liquids." *Hydrometallurgy* 105146.

Mohamed, Samar S., Wael A. Mohamed, A. T. Khalil, dan A. S. Mohra. 2019. "Deep Learning Face Detection and Recognition." *INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS* 1-6.

Mohammed, Assit. Prof. Aree A., dan MSc. Student Shereen A. Anwer. 2014. "Efficient Eye Blink Detection Method for disabled helping domain." *International Journal of Advanced Computer Science and Applications* 202-206.

Niu, Gang, dan Ququ Chen. 2018. "Learning an video frame-based face detection system for security fields." *Journal of Visual Communication and Image Representation* 457 - 463.

pamungkas, adi. 2018. *Pemrograman Matlab*. 1 November. Diakses Januari 14, 2020. pemrogramanmatlab.com.

Putri, Nila Novia, Mohamad Saiful Islam, dan Imam Subadi. 2018. "PERBANDINGAN LUARAN FUNGSIONAL PASIEN STROKE ISKEMIK AKUT PADA PEROKOK DAN." *Comparison of Acute Ischemic Stroke Functional Outcome* 65-71.

Raspberrypi. 2012. *raspberrypi*. 1 Februari. Diakses January 14, 2020. www.raspberrypi.org.

Rosebrock, Adrian. 2017. *pyimagesearch*. 3 April. Diakses Januari 14, 2020. www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/.

sdcard. 2014. *sdcard*. 1 Januari. Diakses januari 14, 2020. www.sdcard.org.

Setiawan, Puja. 2010. "STATISTIKA - PROBABILITAS." www.academia.edu 1-61.

Shiotsuki, Hiroyuki, Yasuaki Saijo, Yoichi Ogushi, dan Shotai Kobayashi. 2019. "Relationships between Alcohol Intake and Ischemic Stroke Severity in Sex Stratified Analysis for Japanese Acute Stroke Patients." *Journal of Stroke and Cerebrovascular Diseases* 1604 - 1617.

Sugano, Hiroki, dan Ryusuke Miyamoto. 2010. "Highly optimized implementation of OpenCV for the Cell Broadband Engine." *Computer Vision and Image Understanding* 1273 - 1281.

2019. support.hp.com. support.hp.com.

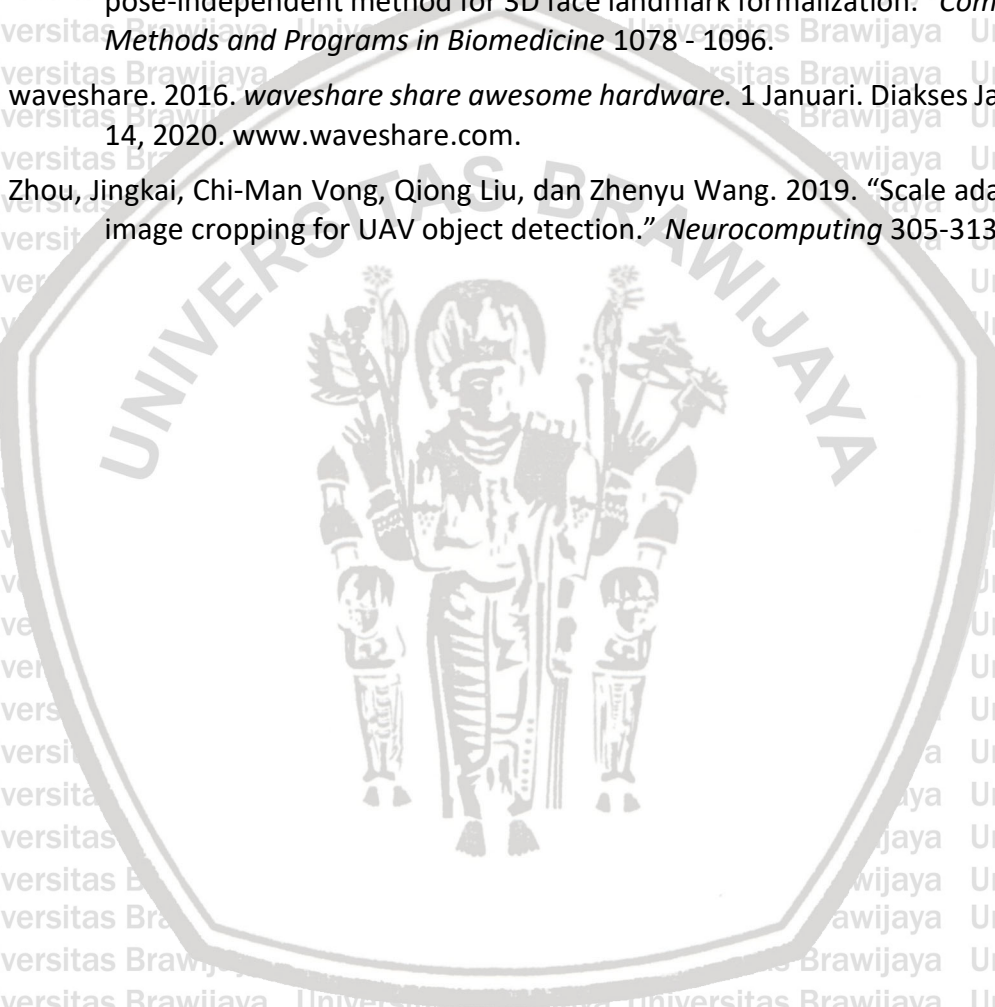
Thajee, Kajorngai, Pathinan Paengnakorn, Wasin Wongwilai, dan Kate Grudpan. 2018. "Application of a webcam camera as a cost-effective sensor with image processing for dual electrochemical – colorimetric detection system." *Talanta* 160 - 165.

Utaminingrum, Fitri, Yuita Arum Sari, Mochammad Ali Fauzi, dan Renaldi Primaswara Prasetya. 2016. "Eye Movement as Navigator for Disabled Person." 2016 *International Conference on Communication and Information Systems*. Bangkok.

Vezzetti, Enrico, Sandro Moos, Federica Marcolin, dan Vincenzo Stola. 2012. "A pose-independent method for 3D face landmark formalization." *Computer Methods and Programs in Biomedicine* 1078 - 1096.

waveshare. 2016. *waveshare share awesome hardware*. 1 Januari. Diakses Januari 14, 2020. www.waveshare.com.

Zhou, Jingkai, Chi-Man Vong, Qiong Liu, dan Zhenyu Wang. 2019. "Scale adaptive image cropping for UAV object detection." *Neurocomputing* 305-313.



LAMPIRAN A TABEL

A.1 Tabel Data Uji Pada Subjek 1 hingga 15

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 1 Anak-anak	Terang (50-400)	Kiri	30	4.2	3.0	Kiri	Benar	100%
2				40	4.0	3.3	Kiri	Benar	
3				50	4.0	3.0	Kiri	Benar	
4			Kanan	30	0.1	2.9	Kanan	Benar	100%
5				40	0.1	3.3	Kanan	Benar	
6				50	0.1	3.1	Kanan	Benar	
7			Atas	30	0.7	2.0	Atas	Benar	100%
8				40	0.7	2.2	Atas	Benar	
9				50	0.7	2.0	Atas	Benar	
10			Bawah	30	0.1	4.6	Kanan	Salah	66%
11				40	0.0	5.0	Bawah	Benar	
12				50	0.1	5.4	Bawah	Benar	
13			Depan	30	0.8	2.7	Bawah	Benar	66%
14				40	0.8	2.7	Bawah	Benar	
15				50	0.0	2.5	Atas	Salah	
16		Redup (1-49)	Kiri	30	13	2.6	Kiri	Benar	100%
17				40	13	2.6	Kiri	Benar	
18				50	18	2.7	Kiri	Benar	
19			Kanan	30	0.5	3.1	Kanan	Benar	100%
20				40	0.5	3.1	Kanan	Benar	
21				50	0.5	2.9	Kanan	Benar	
22			Atas	30	0.8	2.0	Atas	Benar	100%
23				40	0.8	2.0	Atas	Benar	
24				50	0.9	2.0	Atas	Benar	
25			Bawah	30	2.5	6.2	Bawah	Benar	100%
26				40	2.5	5.8	Bawah	Benar	
27				50	2.5	5.7	Bawah	Benar	
28			Depan	30	0.9	2.7	Depan	Benar	100%
29				40	1.0	2.9	Depan	Benar	
30				50	1.1	2.8	Depan	Benar	
Rata – rata									96%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1		Terang	Kiri	30	2.3	2.4	Atas	Salah	0%

2	Subjek 2 Anak-anak	(50-400)	Kanan	40	2.4	2.3	Atas	Salah	66%
3				50	2.6	2.4	Atas	Salah	
4				30	0.2	2.4	Kiri	Salah	
5				40	0.2	2.6	Kanan	Benar	
6				50	0.2	2.6	Kanan	Benar	
7				30	0.9	2.2	Atas	Benar	
8			Atas	40	0.9	2.3	Atas	Benar	100%
9				50	0.9	2.3	Atas	Benar	
10				30	3.3	10	Bawah	Benar	
11			Bawah	40	1.0	8.6	Bawah	Benar	100%
12				50	3.5	8.5	Bawah	Benar	
13			Depan	30	0.7	2.4	Atas	Salah	33%
14				40	0.7	2.5	Atas	Salah	
15				50	0.7	2.6	Depan	Benar	
16		Redup (1-49)	Kiri	30	3.2	2.5	Atas	Salah	33%
17				40	3.1	2.4	Atas	Salah	
18				50	3.5	2.6	Kiri	Benar	
19			Kanan	30	0.1	2.8	Kanan	Benar	100%
20				40	0.1	2.6	Kanan	Benar	
21				50	0.1	2.8	Kanan	Benar	
22			Atas	30	1.1	2.2	Atas	Benar	100%
23				40	1.1	2.3	Atas	Benar	
24				50	1.2	2.2	Atas	Benar	
25			Bawah	30	2.5	7.9	Bawah	Benar	100%
26				40	0.1	9.6	Bawah	Benar	
27				50	0.1	7.9	Bawah	Benar	
28			Depan	30	1.0	2.5	Atas	Salah	0%
29				40	0.9	2.5	Atas	Salah	
30				50	0.8	2.5	Atas	Salah	
Rata – rata									63%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1		Terang	Kiri	30	1.4	2.5	Atas	Salah	33%

2	Subjek 3 Anak-anak	(50-400)	Kanan	40	1.5	2.6	Kiri	Benar	100%
3				50	1.3	2.4	Atas	Salah	
4				30	0.4	2.8	Kanan	Benar	
5				40	0.3	3.1	Kanan	Benar	
6				50	0.4	3.1	Kanan	Benar	
7			Atas	30	0.6	2.1	Atas	Benar	100%
8				40	0.6	2.1	Atas	Benar	
9				50	0.5	2.2	Atas	Benar	
10			Bawah	30	0.4	7.6	Bawah	Benar	100%
11				40	2.9	8.4	Bawah	Benar	
12				50	1.6	7.1	Bawah	Benar	
13			Depan	30	1.8	3.1	Kiri	Salah	33%
14				40	2.2	3.3	Kiri	Salah	
15				50	1.1	3.7	Depan	Benar	
16		Redup (1-49)	Kiri	30	1.6	2.4	Atas	Salah	33%
17				40	2.0	2.7	Kiri	Benar	
18				50	1.6	2.4	Atas	Salah	
19			Kanan	30	0.4	2.6	Kanan	Benar	66%
20				40	0.3	2.5	Atas	Salah	
21				50	0.3	2.7	Kanan	Benar	
22			Atas	30	0.7	2.1	Atas	Benar	100%
23				40	0.7	2.1	Atas	Benar	
24				50	0.7	2.1	Atas	Benar	
25			Bawah	30	0.1	8.8	Bawah	Benar	100%
26				40	0.1	8.2	Bawah	Benar	
27				50	0.1	9.0	Bawah	Benar	
28			Depan	30	2.1	2.3	Atas	Salah	0%
29				40	2.2	2.3	Atas	Salah	
30				50	2.1	2.4	Atas	Salah	
Rata – rata									66%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 4 Anak-anak	Terang (50-400)	Kiri	30	2.1	4.6	Kiri	Benar	100%
2				40	1.9	4.6	Kiri	Benar	
3				50	2.4	4.6	Kiri	Benar	
4			Kanan	30	0.1	4.7	Kanan	Benar	66%
5				40	0.1	5.8	Bawah	Salah	
6				50	0.1	4.8	Kanan	Benar	
7			Atas	30	0.8	2.3	Atas	Benar	66%
8				40	0.9	2.6	Depan	Salah	
9				50	0.9	2.4	Atas	Benar	
10			Bawah	30	0.4	8.6	Bawah	Benar	100%
11				40	2.5	10	Bawah	Benar	
12				50	2.8	6.8	Bawah	Benar	
13			Depan	30	0.6	3.2	Depan	Benar	100%
14				40	0.6	3.5	Depan	Benar	
15				50	0.6	3.0	Depan	Benar	
16		Redup (1-49)	Kiri	30	1.4	4.3	Kiri	Benar	100%
17				40	1.2	3.9	Kiri	Benar	
18				50	1.5	3.6	Kiri	Benar	
19			Kanan	30	0.5	4.8	Kanan	Benar	100%
20				40	0.5	4.3	Kanan	Benar	
21				50	0.5	4.1	Kanan	Benar	
22			Atas	30	1.1	2.9	Depan	Salah	0%
23				40	1.0	2.9	Depan	Salah	
24				50	1.1	2.9	Depan	Salah	
25			Bawah	30	1.3	9.7	Bawah	Benar	100%
26				40	1.3	9.9	Bawah	Benar	
27				50	1.0	6.5	Bawah	Benar	
28			Depan	30	0.8	3.0	Depan	Benar	100%
29				40	0.8	2.9	Depan	Benar	
30				50	0.9	3.2	Depan	Benar	
Rata – rata									83%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 5 Remaja	Terang (50-400)	Kiri	30	1.3	4.6	Kiri	Benar	100%
2				40	1.3	4.8	Kiri	Benar	
3				50	1.3	4.3	Kiri	Benar	
4			Kanan	30	0.5	3.9	Kanan	Benar	100%
5				40	0.4	3.2	Kanan	Benar	
6				50	0.5	3.4	Kanan	Benar	
7			Atas	30	0.9	2.3	Atas	Benar	100%
8				40	0.8	2.3	Atas	Benar	
9				50	0.8	2.5	Atas	Benar	
10			Bawah	30	0.8	6.5	Bawah	Benar	100%
11				40	0.7	6.9	Bawah	Benar	
12				50	0.9	6.5	Bawah	Benar	
13			Depan	30	0.8	3.0	Depan	Benar	100%
14				40	0.8	3.2	Depan	Benar	
15				50	0.8	3.0	Depan	Benar	
16		Redup (1-49)	Kiri	30	1.5	4.2	Kiri	Benar	100%
17				40	1.5	4.2	Kiri	Benar	
18				50	1.2	3.8	Kiri	Benar	
19			Kanan	30	0.4	3.6	Kanan	Benar	100%
20				40	0.3	3.6	Kanan	Benar	
21				50	0.3	4.0	Kanan	Benar	
22			Atas	30	0.8	2.2	Atas	Benar	100%
23				40	0.8	2.3	Atas	Benar	
24				50	0.8	2.4	Atas	Benar	
25			Bawah	30	2.3	10.	Bawah	Benar	100%
26				40	2.7	9.9	Bawah	Benar	
27				50	3.0	9.5	Bawah	Benar	
28			Depan	30	0.7	3.1	Depan	Benar	100%
29				40	0.8	3.4	Depan	Benar	
30				50	0.8	3.2	Depan	Benar	
Rata – rata									100%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 6 Remaja	Terang (50-400)	Kiri	30	2.9	4.7	Kiri	Benar	66%
2				40	1.7	4.5	Kiri	Benar	
3				50	0.8	4.1	Depan	Salah	
4			Kanan	30	0.4	4.1	Kanan	Benar	100%
5				40	0.3	4.0	Kanan	Benar	
6				50	0.5	4.3	Kanan	Benar	
7			Atas	30	0.9	2.5	Atas	Benar	66%
8				40	0.8	2.5	Atas	Benar	
9				50	0.9	2.7	Depan	Salah	
10			Bawah	30	1.0	6.7	Bawah	Benar	100%
11				40	1.1	7.3	Bawah	Benar	
12				50	0.9	7.0	Bawah	Benar	
13			Depan	30	0.8	3.2	Depan	Benar	100%
14				40	0.8	3.2	Depan	Benar	
15				50	0.9	3.4	Depan	Benar	
16		Redup (1-49)	Kiri	30	2.4	3.4	Kiri	Benar	100%
17				40	1.7	3.8	Kiri	Benar	
18				50	1.5	3.3	Kiri	Benar	
19			Kanan	30	0.2	4.5	Kanan	Benar	100%
20				40	0.2	3.9	Kanan	Benar	
21				50	0.2	4.3	Kanan	Benar	
22			Atas	30	0.6	3.0	Depan	Salah	0%
23				40	0.6	2.6	Depan	Salah	
24				50	0.6	2.6	Depan	Salah	
25			Bawah	30	0.9	8.9	Bawah	Benar	100%
26				40	1.2	10	Bawah	Benar	
27				50	1.0	9.0	Bawah	Benar	
28			Depan	30	0.8	3.3	Depan	Benar	100%
29				40	0.8	3.1	Depan	Benar	
30				50	0.8	3.3	Depan	Benar	
Rata – rata									86%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 7 Remaja	Terang (50-400)	Kiri	30	1.3	3.3	Kiri	Benar	100%
2				40	1.4	3.5	Kiri	Benar	
3				50	1.4	3.6	Kiri	Benar	
4			Kanan	30	0.4	3.7	Kanan	Benar	100%
5				40	0.4	3.7	Kanan	Benar	
6				50	0.4	3.6	Kanan	Benar	
7			Atas	30	0.8	2.6	Depan	Salah	66%
8				40	0.8	2.5	Atas	Benar	
9				50	0.8	2.5	Atas	Benar	
10			Bawah	30	0.7	6.6	Bawah	Benar	100%
11				40	0.7	6.6	Bawah	Benar	
12				50	0.8	8.0	Bawah	Benar	
13			Depan	30	0.7	3.3	Depan	Benar	100%
14				40	0.7	3.3	Depan	Benar	
15				50	0.6	2.9	Depan	Benar	
16		Redup (1-49)	Kiri	30	2.2	4.2	Kiri	Benar	66%
17				40	2.6	4.1	Kiri	Benar	
18				50	2.5	5.5	Bawah	Salah	
19			Kanan	30	0.4	3.6	Kanan	Benar	100%
20				40	0.3	3.2	Kanan	Benar	
21				50	0.4	3.7	Kanan	Benar	
22			Atas	30	0.8	2.1	Atas	Benar	100%
23				40	0.8	2.2	Atas	Benar	
24				50	0.8	2.1	Atas	Benar	
25			Bawah	30	0.7	9.0	Bawah	Benar	100%
26				40	0.7	9.0	Bawah	Benar	
27				50	0.8	7.6	Bawah	Benar	
28			Depan	30	0.7	3.5	Depan	Benar	100%
29				40	0.7	3.3	Depan	Benar	
30				50	0.7	3.0	Depan	Benar	
Rata – rata									93%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 8 Remaja	Terang (50-400)	Kiri	30	1.7	3.2	Kiri	Benar	100%
2				40	2.0	3.6	Kiri	Benar	
3				50	2.0	3.4	Kiri	Benar	
4			Kanan	30	0.2	2.9	Kanan	Benar	100%
5				40	0.2	2.8	Kanan	Benar	
6				50	0.1	3.0	Kanan	Benar	
7			Atas	30	0.9	2.4	Atas	Benar	100%
8				40	0.9	2.3	Atas	Benar	
9				50	0.9	2.1	Atas	Benar	
10			Bawah	30	0.7	5.6	Bawah	Benar	100%
11				40	0.8	5.8	Bawah	Benar	
12				50	0.8	6.1	Bawah	Benar	
13			Depan	30	0.7	3.3	Depan	Benar	100%
14				40	0.7	3.0	Depan	Benar	
15				50	0.7	3.4	Depan	Benar	
16		Redup (1-49)	Kiri	30	2.5	3.5	Kiri	Benar	100%
17				40	3.7	3.7	Kiri	Benar	
18				50	3.6	3.6	Kiri	Benar	
19			Kanan	30	0.3	3.3	Kanan	Benar	100%
20				40	0.3	3.2	Kanan	Benar	
21				50	0.3	3.0	Kanan	Benar	
22			Atas	30	2.5	2.8	Kiri	Salah	0%
23				40	3.7	2.8	Kiri	Salah	
24				50	4.1	2.6	Kiri	Salah	
25			Bawah	30	0.1	7.2	Bawah	Benar	100%
26				40	0.1	6.1	Bawah	Benar	
27				50	0.1	6.9	Bawah	Benar	
28			Depan	30	0.9	3.4	Depan	Benar	100%
29				40	0.8	3.9	Depan	Benar	
30				50	0.9	3.6	Depan	Benar	
Rata – rata									90%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi	
1	Subjek 9 Remaja	Terang (50-400)	Kiri	30	2.4	3.0	Kiri	Benar	100%	
2				40	2.0	3.0	Kiri	Benar		
3				50	1.7	2.9	Kiri	Benar		
4			Kanan	30	0.4	2.7	Kanan	Benar	100%	
5				40	0.4	2.9	Kanan	Benar		
6				50	0.4	2.8	Kanan	Benar		
7			Atas	30	0.9	2.1	Atas	Benar	100%	
8				40	0.8	2.0	Atas	Benar		
9				50	0.9	2.1	Atas	Benar		
10			Bawah	30	0.8	5.2	Bawah	Benar	100%	
11				40	0.8	5.0	Bawah	Benar		
12				50	0.8	5.5	Bawah	Benar		
13		Depan	30	0.8	2.7	Depan	Benar	100%		
14			40	0.8	2.7	Depan	Benar			
15			50	0.8	2.9	Depan	Benar			
16		Redup (1-49)	Kiri	30	2.3	3.0	Kiri	Benar	100%	
17				40	2.1	3.1	Kiri	Benar		
18				50	1.8	2.7	Kiri	Benar		
19			Kanan	30	0.3	2.7	Kanan	Benar	66%	
20				40	0.3	2.8	Kanan	Benar		
21				50	0.3	2.5	Atas	Salah		
22			Atas	30	0.8	2.0	Atas	Benar	100%	
23				40	0.8	2.0	Atas	Benar		
24				50	0.8	2.1	Atas	Benar		
25			Bawah	30	0.6	6.9	Bawah	Benar	100%	
26				40	0.6	5.5	Bawah	Benar		
27				50	0.5	6.6	Bawah	Benar		
28			Depan	30	0.7	2.9	Depan	Benar	100%	
29				40	0.8	2.9	Depan	Benar		
30				50	0.8	2.7	Depan	Benar		
Rata – rata									96%	

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 10 Remaja	Terang (50-400)	Kiri	30	2.5	2.6	Kiri	Benar	100%
2				40	2.2	2.6	Kiri	Benar	
3				50	2.1	2.7	Kiri	Benar	
4			Kanan	30	0.4	3.2	Kanan	Benar	100%
5				40	0.4	3.2	Kanan	Benar	
6				50	0.4	3.3	Kanan	Benar	
7			Atas	30	1.1	2.4	Atas	Benar	100%
8				40	1.1	2.4	Atas	Benar	
9				50	1.1	2.2	Atas	Benar	
10			Bawah	30	2.8	7.1	Bawah	Benar	100%
11				40	3.0	7.7	Bawah	Benar	
12				50	6.3	6.4	Bawah	Benar	
13			Depan	30	1.1	3.1	Depan	Benar	100%
14				40	1.1	2.9	Depan	Benar	
15				50	1.1	2.8	Depan	Benar	
16		Redup (1-49)	Kiri	30	4.2	2.8	Kiri	Benar	100%
17				40	3.7	2.8	Kiri	Benar	
18				50	3.8	3.0	Kiri	Benar	
19			Kanan	30	0.2	3.4	Kanan	Benar	100%
20				40	0.4	3.6	Kanan	Benar	
21				50	0.3	3.2	Atas	Benar	
22			Atas	30	1.1	2.5	Atas	Benar	100%
23				40	1.1	2.3	Atas	Benar	
24				50	1.0	2.2	Atas	Benar	
25			Bawah	30	2.7	8.8	Bawah	Benar	100%
26				40	3.2	8.5	Bawah	Benar	
27				50	2.8	7.9	Bawah	Benar	
28			Depan	30	0.9	3.0	Depan	Benar	100%
29				40	0.8	2.9	Depan	Benar	
30				50	0.9	2.8	Depan	Benar	
Rata – rata									100%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 11 Remaja	Terang (50-400)	Kiri	30	4.8	3.2	Kiri	Benar	100%
2				40	4.0	3.4	Kiri	Benar	
3				50	4.9	3.3	Kiri	Benar	
4			Kanan	30	0.1	4.4	Kanan	Benar	100%
5				40	0.1	4.5	Kanan	Benar	
6				50	0.1	3.9	Kanan	Benar	
7			Atas	30	0.8	2.5	Atas	Benar	100%
8				40	0.8	2.5	Atas	Benar	
9				50	0.8	2.5	Atas	Benar	
10			Bawah	30	0.3	5.1	Bawah	Benar	100%
11				40	0.7	4.9	Bawah	Benar	
12				50	0.5	5.2	Bawah	Benar	
13			Depan	30	0.6	3.1	Depan	Benar	100%
14				40	0.6	3.0	Depan	Benar	
15				50	0.6	3.0	Depan	Benar	
16		Redup (1-49)	Kiri	30	8.2	5.2	Bawah	Salah	0%
17				40	6.1	5.4	Bawah	Salah	
18				50	4.8	7.3	Bawah	Salah	
19			Kanan	30	0.1	3.3	Kanan	Benar	100%
20				40	0.1	3.5	Kanan	Benar	
21				50	0.1	2.9	Atas	Benar	
22			Atas	30	0.9	2.5	Atas	Benar	100%
23				40	0.9	2.5	Atas	Benar	
24				50	0.9	2.4	Atas	Benar	
25			Bawah	30	1.6	6.4	Bawah	Benar	100%
26				40	1.0	6.3	Bawah	Benar	
27				50	0.8	6.0	Bawah	Benar	
28			Depan	30	0.7	2.7	Depan	Benar	100%
29				40	0.8	2.8	Depan	Benar	
30				50	0.8	2.6	Depan	Benar	
Rata – rata									90%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 12 Remaja	Terang (50-400)	Kiri	30	2.1	3.0	Kiri	Benar	100%
2				40	2.2	3.1	Kiri	Benar	
3				50	2.3	3.3	Benar	Benar	
4			Kanan	30	0.0	2.8	Kanan	Benar	100%
5				40	0.0	3.1	Kanan	Benar	
6				50	0.0	3.5	Kanan	Benar	
7			Atas	30	0.8	2.2	Atas	Benar	100%
8				40	0.7	2.3	Atas	Benar	
9				50	0.7	2.5	Atas	Benar	
10			Bawah	30	0.6	7.3	Bawah	Benar	100%
11				40	0.6	8.7	Bawah	Benar	
12				50	0.7	7.2	Bawah	Benar	
13			Depan	30	0.7	4.8	Depan	Benar	100%
14				40	0.7	4.6	Depan	Benar	
15				50	0.6	4.8	Depan	Benar	
16		Redup (1-49)	Kiri	30	1.2	3.0	Kiri	Benar	100%
17				40	1.2	3.1	Kiri	Benar	
18				50	1.4	3.1	Kiri	Benar	
19			Kanan	30	0.1	2.7	Kanan	Benar	100%
20				40	0.1	2.8	Kanan	Benar	
21				50	0.0	2.8	Kanan	Benar	
22			Atas	30	0.7	2.3	Atas	Benar	100%
23				40	0.7	2.3	Atas	Benar	
24				50	0.8	2.4	Atas	Benar	
25			Bawah	30	0.4	10	Bawah	Benar	100%
26				40	0.7	7.2	Bawah	Benar	
27				50	0.6	11	Bawah	Benar	
28			Depan	30	0.7	2.6	Depan	Benar	100%
29				40	0.7	2.6	Depan	Benar	
30				50	0.7	2.8	Depan	Benar	
Rata – rata									100%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 13 Dewasa	Terang (50-400)	Kiri	30	1.2	3.2	Kiri	Benar	66%
2				40	1.2	3.4	Kiri	Benar	
3				50	0.9	3.0	Depan	Salah	
4			Kanan	30	0.5	3.1	Kanan	Benar	100%
5				40	0.2	3.0	Kanan	Benar	
6				50	0.1	3.2	Kanan	Benar	
7			Atas	30	0.9	3.2	Atas	Benar	100%
8				40	0.9	2.2	Atas	Benar	
9				50	0.9	2.3	Atas	Benar	
10			Bawah	30	0.1	5.2	Bawah	Benar	100%
11				40	1.0	5.1	Bawah	Benar	
12				50	0.7	5.3	Bawah	Benar	
13			Depan	30	0.8	3.2	Depan	Benar	100%
14				40	0.9	2.8	Depan	Benar	
15				50	0.8	3.0	Depan	Benar	
16	Redup (1-49)		Kiri	30	1.3	4.9	Bawah	Salah	66%
17				40	1.2	4.4	Kiri	Benar	
18				50	1.2	3.8	Kiri	Benar	
19			Kanan	30	0.1	3.4	Kanan	Benar	100%
20				40	0.0	3.4	Kanan	Benar	
21				50	0.1	3.2	Kanan	Benar	
22			Atas	30	0.7	2.3	Atas	Benar	66%
23				40	0.7	2.3	Atas	Benar	
24				50	0.7	2.6	Depan	Salah	
25			Bawah	30	0.7	5.7	Bawah	Benar	100%
26				40	0.6	5.3	Bawah	Benar	
27				50	0.2	7.3	Bawah	Benar	
28			Depan	30	-	-	-	Salah	0%
29				40	0.5	2.9	Kanan	Salah	
30				50	-	-	-	Salah	
Rata – rata									80%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 14 Dewasa	Terang (50-400)	Kiri	30	1.3	3.8	Kiri	Benar	100%
2				40	1.5	4.1	Kiri	Benar	
3				50	1.4	4.3	Kiri	Benar	
4			Kanan	30	0.3	2.8	Kanan	Benar	100%
5				40	0.3	2.7	Kanan	Benar	
6				50	0.3	2.8	Kanan	Benar	
7			Atas	30	0.9	2.3	Atas	Benar	100%
8				40	0.9	2.3	Atas	Benar	
9				50	0.9	2.4	Atas	Benar	
10			Bawah	30	0.5	6.4	Bawah	Benar	100%
11				40	0.7	5.7	Bawah	Benar	
12				50	0.2	10	Bawah	Benar	
13			Depan	30	1.1	3.1	Depan	Benar	100%
14				40	0.6	3.7	Depan	Benar	
15				50	0.7	3.2	Depan	Benar	
16	Redup (1-49)	Kiri	30	15	3.5	Kiri	Benar	100%	
17			40	3.6	3.0	Kiri	Benar		
18			50	5.3	3.0	Kiri	Benar		
19			Kanan	30	0.3	4.4	Kanan	Benar	100%
20				40	0.2	4.3	Kanan	Benar	
21				50	0.2	4.7	Kanan	Benar	
22			Atas	30	0.8	2.4	Atas	Benar	66%
23				40	0.8	2.4	Atas	Benar	
24				50	0.8	2.6	Depan	Salah	
25		Bawah	30	0.2	7.6	Bawah	Benar	100%	
26			40	0.9	7.4	Bawah	Benar		
27			50	0.7	7.2	Bawah	Benar		
28		Depan	30	0.9	3.4	Depan	Benar	100%	
29			40	0.8	3.1	Depan	Benar		
30			50	0.8	3.4	Depan	Benar		
Rata – rata									96%

No	Subjek	Lux	Gerakan Mata	Jarak	Nilai Deteksi Arah	Nilai Deteksi Kedipan	Deteksi	Benar / Salah	Akurasi
1	Subjek 15 Dewasa	Terang (50-400)	Kiri	30	2.6	2.0	Atas	Salah	66%
2				40	2.6	2.6	Kiri	Benar	
3				50	2.6	3.0	Depan	Benar	
4			Kanan	30	0.4	3.0	Kanan	Benar	100%
5				40	0.2	2.9	Kanan	Benar	
6				50	0.2	2.9	Kanan	Benar	
7			Atas	30	0.4	2.1	Atas	Benar	100%
8				40	0.3	2.1	Atas	Benar	
9				50	0.3	2.1	Atas	Benar	
10			Bawah	30	0.1	7.0	Bawah	Benar	100%
11				40	0.0	5.3	Bawah	Benar	
12				50	0.1	5.6	Bawah	Benar	
13			Depan	30	0.7	2.8	Depan	Benar	100%
14				40	0.7	2.8	Depan	Benar	
15				50	0.7	2.8	Depan	Benar	
16		Redup (1-49)	Kiri	30	7.0	2.6	Kiri	Benar	100%
17				40	6.1	2.6	Kiri	Benar	
18				50	4.0	2.6	Kiri	Benar	
19			Kanan	30	0.5	3.0	Kanan	Benar	100%
20				40	0.5	3.0	Kanan	Benar	
21				50	0.5	2.9	Kanan	Benar	
22			Atas	30	0.9	2.2	Atas	Benar	100%
23				40	0.9	2.2	Atas	Benar	
24				50	1.0	2.1	Atas	Benar	
25			Bawah	30	0.1	5.1	Bawah	Benar	66%
26				40	2.5	4.9	Bawah	Benar	
27				50	0.3	4.5	Kanan	Salah	
28			Depan	30	0.9	2.7	Depan	Benar	100%
29				40	0.9	2.7	Depan	Benar	
30				50	0.9	2.7	Depan	Benar	
Rata – rata									93%

LAMPIRAN B GAMBAR

B.1 Gambar Hasil Deteksi *Facial landmark*



a (1)



a (2)



a (3)



a (4)



a (5)



a (6)



a (7)



a (8)



a (9)



a (10)



a (11)



a (12)



a (13)



a (14)



a (15)



b (1)



b (2)



b (3)



b (4)



b (5)



b (6)



b (7)



b (8)



b (9)



b (10)



b (11)



b (12)



b (13)



b (14)



b (15)

B.2 Gambar Hasil Deteksi Pergerakan Mata

