

**DETEKSI PERGERAKAN KEPALA BERDASARKAN ANALISIS
DETEKSI TEPI WAJAH BERBASIS RASPBERRY PI UNTUK
IMPLEMENTASI PEMILIHAN MENU DISPLAY**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

Budi Atmoko

NIM: 145150300111125



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

MALANG

2020



PENGESAHAN

**DETEKSI PERGERAKAN KEPALA BERDASARKAN ANALISIS DETEKSI TEPI WAJAH
BERBASIS RASPBERRY PI UNTUK IMPLEMENTASI PEMILIHAN MENU DISPLAY**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik**

**Disusun Oleh :
Budi Atmoko
NIM: 14515030011125**

**Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2020
Telah diperiksa dan disetujui oleh:**

Dosen Pembimbing I

**Dr. Eng. Fitri Utaminingrum, S.T, M.T
NIP: 198207102008122001**

Mengetahui

Ketua Jurusan Teknik Informatika



**Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 197105182003121001**

PERNYATAAN 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 pustaka.

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, 3 Januari 2020



Budi Atmoko

NIM: 145150300111125

PRAKATA

Puja dan puji syukur atas kehadiran Allah SWT karena berkat rahmat serta hidayah-Nya, penulis dapat menyelesaikan penelitian untuk memperoleh gelar Sarjana Teknik yang berjudul “Deteksi Pergerakan Kepala Berdasarkan Analisis Deteksi Tepi Wajah Berbasis Raspberry Pi Untuk Implementasi Pemilihan Menu Display”.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan-bantuan yang diberikan oleh berbagai pihak, maka penulis mengucapkan banyak-banyak terima kasih kepada :

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Deskan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
3. Bapak Dahniyal Syauqy, S.T., M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
4. Ibu Dr. Eng.Fitri Utamingrum, S.T, M.T selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingan, sehingga dapat menyelesaikan skripsi ini dengan baik.
5. Kedua orangtua saya atas segala nasehat, semangat dan dorongan baik berupa do’a maupun materi selama penulis melakukan penelitian.
6. Teman-teman Teknik Komputer angkatan 2014 yang telah banyak membantu dan mendukung dalam proses pengerjaan skripsi.
7. Faiz,Johannes,Giri,Xavierro,Syifau,Royyanul,Radea,Agra,Mantiqo,Rezaqi ,Wahyu Ilyas, Primastyo yang telah memberi semangat sehingga skripsi dapat terselesaikan dengan baik
8. Irfan Reza,Riyyan Royhan, Aliffandi, Faris, Firmansyah yang telah banyak memberi dukungan kepada penulis sehingga penulis mampu menyelesaikan skripsi.
9. Para subjek penelitian dan orang-orang yang tidak dapat disebutkan satu persatu yang telah memberikan doa serta dukungannya.

Penulis menyadari dalam penyusunan skripsi ini terdapat banyak kekurangan, sehingga saran dan kritik yang membangun sangat dibutuhkan untuk ke depannya agar lebih baik lagi. Semoga penelitian ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Malang, 3 Januari 2020

Penulis

budiatmoko6@gmail.com

ABSTRAK

Budi Atmoko, Deteksi Pergerakan Kepala Berdasarkan Analisis Deteksi Tepi Wajah Berbasis Raspberry Pi Untuk Implementasi Pemilihan Menu Display

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

Dalam dunia komputasi terdapat fitur penting dalam mendukung tampilan suatu program guna memudahkan pengoperasian program yakni fitur menu. Menu selalu disediakan baik pada program GUI (Graphical User Interface) dan juga CLI (Command Line Interface). Dengan adanya menu maka akan lebih mudah untuk mengakses suatu fungsi program dengan kesesuaian kehendak user. Namun kebanyakan dari fitur navigasi yang disediakan dalam menu program, masih sebagian besar menggunakan tangan sebagai media untuk mengoperasikan opsi dari menu. Sehingga memiliki sedikit batasan, khususnya bagi pengguna atau user yang mengalami cedera pada organ tangan baik berat maupun ringan atau pengguna yang mengalami disabilitas fisik pada tangan. Solusi yang ditawarkan adalah image processing dengan konversi nilai dari RGB ke *HSV* dan *YCbCr* selanjutnya dibentuk bounding box serta ditentukan nilai dari x, y, w, h sehingga terbentuk titik centroid dan nilai kuadran, dari perbandingan nilai parameter tersebut sistem mampu mengklasifikasikan gerakan sehingga dapat diubah kedalam navigasi menu LCD. Hasil menunjukkan bahwa akurasi untuk tingkat pencahayaan pada kondisi *luminance* 1500-1800 mencapai 96 % dan pada *luminance* 1250-1400 mencapai 92% pada jarak perbedaan jarak menunjukkan hasil pada jarak 40cm memiliki jumlah rata-rata akurasi 94%, pada kondisi jarak 50cm sebesar 93%, jarak 60 memiliki akurasi 96% dan pada jarak 80cm rata rata untuk jarak tersebut adalah adalah 95%. Bisa diambil kesimpulan bahwa untuk jarak terjauh yakni 60 dan 80 cm dapat memperoleh akurasi tinggi karena pada saat objek dekat tingkat sensitifitas terhadap warna akan lebih besar, fitur wajah seperti alis dan rambut dapat mempengaruhi hasil dari percobaan karena warna yang dideteksi pada penelitian ini hanya warna kulit sehingga alis dan rambut tidak termasuk dan akan mempengaruhi hasil deteksi. Visualisasi gambar sangat baik dapat dilihat pada hasil yang diperoleh dalam percobaan, hasil menunjukkan minimal area noise . dan oval terbentuk secara sempurna.

Kata kunci : *YCbCr*, *HSV*, Visual Studio, Raspberry Pi, Kuadran , Centroid

ABSTRACT

Budi Atmoko, Analysis Of Head Movement Detection Based On Face Edge Detection Based On Raspberry Pi For Implementation Of Display Menu Selection

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

In the world of computing there are important features in supporting the appearance of a program in order to facilitate the operation of the program namely the menu feature. Menus are always provided in both the GUI (Graphical User Interface) program and also the CLI (Command Line Interface). With the menu, it will be easier to access a program function according to the user's will. However, most of the navigation features provided in the program menu, still most use the hand as a medium to operate options from the menu. So it has a few restrictions, especially for users or users who have injuries to both heavy and light hand organs or users who have physical disabilities on the hands. Research on navigation is considered important in terms of developing technological features with the same goal of being able to cover more users and obstacles such as physical limitations. By connecting to the Raspberry Pi module it is developed with the movement of the head which is able to navigate the menu of the desired object through the LCD display. Through the conversion of values from HSV and YCbCr, then bounding boxes are formed and the values of x, y, w, h are formed so that centroid points are formed again and quadrants are formed from comparison of these parameter values the system is able to classify movements so that they can be changed into LCD navigation menus. Results that show verified values at daytime conditions reach 96% and nightly reach 92% at distances distinguishing distances show results at distances of 40cm having an average amount of 94%, at conditions of 50cm distance of 93%, distances 60 having an accuracy of 96% and at an 80cm distance the average distance is 95%. Conclusions can be reached for the farthest distance of 60 and 80 cm because when the object is near the level of sensitivity to color will be greater, facial features such as eyebrows and hair can affect the experimental results. Very good image visualization can be seen in the results obtained in the experiments, the results show minimal area noise. and the oval is perfectly formed.

Keyword : YCbCr, HSV, Visual Studio, Raspberry Pi, kuadran , centroid

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	7
2.2.1 Disabilitas	8
2.2.2 Citra Digital.....	9
2.2.3 Macam - macam Warna pada Citra Digital	11
2.2.4 Konversi ruangan warna	15
2.2.5 Deteksi Kult	16
2.2.6 <i>Bouding Rect</i>	17
BAB 3 METODOLOGI	18
3.1 Tipe Penelitian	18
3.2 Strategi dan Rancangan Penelitian.....	18
3.2.1 Subjek Penelitian.....	20
3.2.2 Lokasi Penelitian.....	20
3.2.3 Metode Pengumpulan Data.....	20

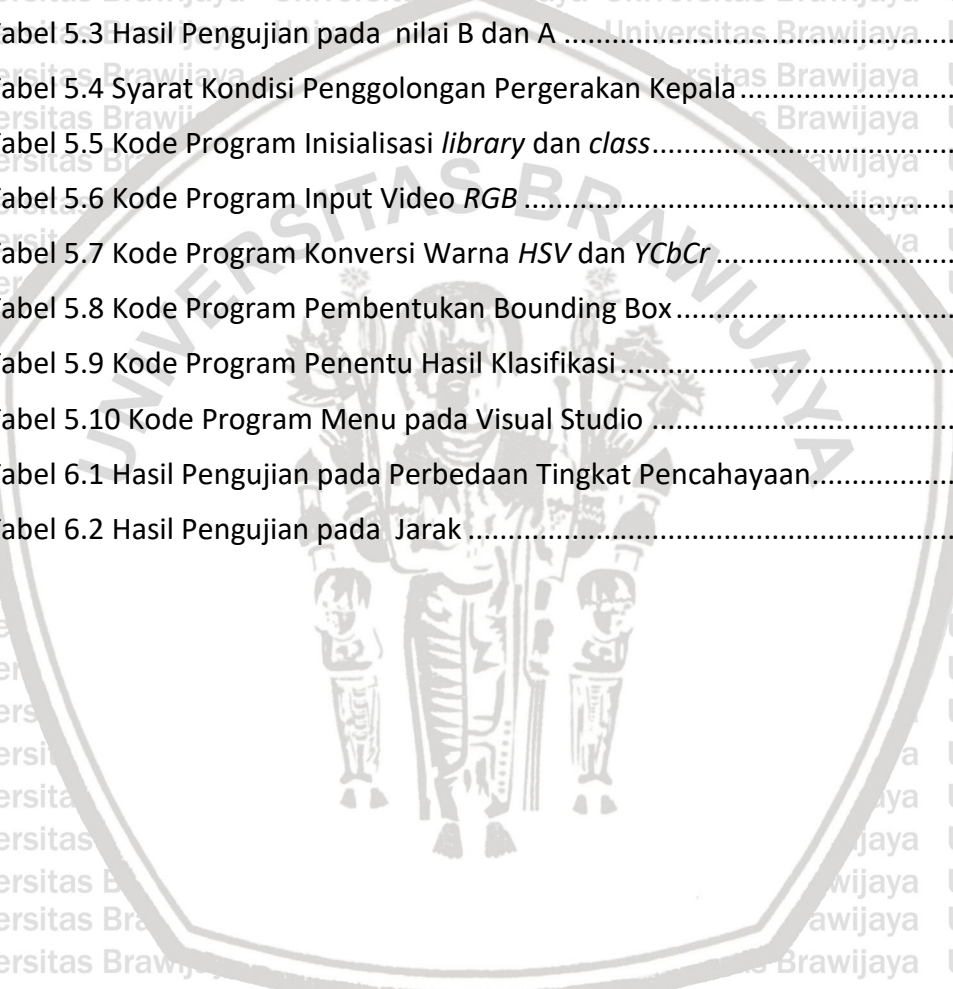


3.2.4 Metode Analisis Data	20
3.2.5 Peralatan Pendukung Penelitian.....	21
BAB 4 REKAYASA KEBUTUHAN.....	22
4.1 Deskripsi Umum Sistem.....	22
4.2 Analisis Kebutuhan	23
4.2.1 Kebutuhan Fungsional.....	23
4.2.2 Kebutuhan Non-Fungsional	24
4.3 Analisis Kebutuhan Perangkat	25
4.3.1 Kebutuhan Perangkat Lunak	25
4.3.2 Kebutuhan Perangkat Keras.....	26
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan Sistem.....	30
5.1.1 Perancangan Perangkat Keras	30
5.1.2 Perancangan Perangkat Lunak.....	31
5.2 Implementasi Sistem	38
5.2.1 Implementasi Perangkat Keras	38
5.2.2 Implementasi Perangkat Lunak.....	39
BAB 6 PENGUJIAN DAN ANALISIS.....	48
6.1 Pengujian pengaruh tingkat pencahayaan pada deteksi kepala	48
6.1.1 Strategi Pengujian	48
6.1.2 Hasil dan Analisis Pengujian.....	48
6.2 Pengujian terhadap jarak pada implementasi sistem deteksi kepala ..	49
6.2.1 Hasil dan Analisis Pengujian.....	49
6.3 Visualisasi masing masing pergerakan kepala dan deteksi kepala pada sistem.....	51
6.3.1 Hasil dan Analisis Pengujian.....	51
BAB 7 Kesimpulan dan saran.....	54
7.1 Kesimpulan.....	54
7.2 Saran	54
DAFTAR REFERENSI	55
LAMPIRAN	56



DAFTAR TABEL

Tabel 2.1 Data Penyandang Disabilitas pada tahun 2011 dikutip dari situs WHO .	8
Tabel 2.2 Komposisi warna berdasarkan ruang warna <i>HSV</i>	14
Tabel 4.1 Spesifikasi Raspberry Pi B+	27
Tabel 5.1 Tabel Pin Perancangan Hardware	31
Tabel 5.2 Hasil Pengujian pada nilai h dan w	37
Tabel 5.3 Hasil Pengujian pada nilai B dan A	37
Tabel 5.4 Syarat Kondisi Penggolongan Pergerakan Kepala.....	38
Tabel 5.5 Kode Program Inisialisasi <i>library</i> dan <i>class</i>	40
Tabel 5.6 Kode Program Input Video <i>RGB</i>	40
Tabel 5.7 Kode Program Konversi Warna <i>HSV</i> dan <i>YCbCr</i>	41
Tabel 5.8 Kode Program Pembentukan Bounding Box	42
Tabel 5.9 Kode Program Penentu Hasil Klasifikasi	45
Tabel 5.10 Kode Program Menu pada Visual Studio	46
Tabel 6.1 Hasil Pengujian pada Perbedaan Tingkat Pencahayaan.....	48
Tabel 6.2 Hasil Pengujian pada Jarak	49



DAFTAR GAMBAR

Gambar 2.1 Elemen Matriks pada Pikel	9
Gambar 2.2 Pembagian nilai RGB pada Citra Digital	10
Gambar 2.3 Matriks pada nilai RGB	10
Gambar 2.4 Logo OpenCV	11
Gambar 2.5 Representasi model warna $YCbCr$ dalam Plane YUV	13
Gambar 2.6 Representasi model warna $YCbCr$ dalam Plane XYZ	13
Gambar 2.7 Representasi model warna HSV	14
Gambar 2.8 Bounding Rect	17
Gambar 3.1 Diagram Alir Metodologi Penelitian	18
Gambar 4.1 Gambaran Umum pada Sistem	22
Gambar 4.2 Logo Microsoft Visual Studio Software	25
Gambar 4.3 Logo OpenCV	26
Gambar 4.4 Modul Raspberry Pi	27
Gambar 4.5 Modul Kamera Raspberry Pi	28
Gambar 4.6 Panel LCD 14 inch	29
Gambar 5.1 Gambar Instalasi LCD dan kamera pada Raspberry-Pi	30
Gambar 5.3 Diagram alur Proses Utama Sistem	32
Gambar 5.4 Diagram alur Proses Konversi Warna	33
Gambar 5.5 Diagram alur Bounding Box	34
Gambar 5.6 Flowchart Klasifikasi Pergerakan Kepala	35
Gambar 5.7 Tampilan <i>Bounding Box</i>	36
Gambar 5.8 Pola Bounding Box	38
Gambar 5.8 Tampilan Prototype Alat Nampak Samping	39
Gambar 5.9 Tampilan Prototype Alat	39
Gambar 5.10 Tampilan Menu dengan 4 Opsi	46
Gambar 6.1 Hasil Visualisasi dari Pergerakan ke Atas	51
Gambar 6.2 Hasil Visualisasi dari Pergerakan ke Bawah	51
Gambar 6.3 Hasil Visualisasi dari Pergerakan ke Kanan	52
Gambar 6.4 Hasil Visualisasi dari Pergerakan ke Kiri	52

DAFTAR LAMPIRAN

Lampiran Kode Program	56
Lampiran Foto Percobaan.....	59
Lampiran Tabel Hasil Percobaan	62



BAB 1 PENDAHULUAN

Bagian utama skripsi terdiri dari beberapa komponen atau bab yang tersusun dengan alur yang logis. Pendahuluan merupakan komponen/bab pertama yang harus menjelaskan apa yang akan dikerjakan dalam skripsi dan mengapa ini perlu dikerjakan.

1.1 Latar belakang

Dalam dunia komputasi terdapat fitur penting dalam mendukung tampilan suatu program guna memudahkan pengoperasian program yakni fitur menu. Menu selalu disediakan baik pada program GUI (Graphical User Interface) dan juga CLI (Command Line Interface). Dengan adanya menu maka akan lebih mudah untuk mengakses suatu fungsi program dengan kesesuaian kehendak user. Dan dengan adanya menu ini maka dapat memudahkan user yang awam sekalipun untuk menjalankan program tanpa perlu memanggil fungsi secara manual pada program. Penggunaan menu sendiri dalam pengaplikasiannya digunakan pada banyak hal (Konrad Baumann,2001). Contohnya, pada program sistem operasi, pada program berbasis web maupun aplikasi perangkat lunak (software) guna meningkatkan kualitas visualisasi, mengatur berbagai konten dan juga membuat kategori tertentu pada tiap fungsi ataupun fitur yang disediakan oleh program tersebut.

Pada pengaplikasian menu menurut Advent Jose (2014) ada beberapa tipe pengoperasian, yang umum dilakukan adalah dengan pengoperasian melalui tombol ataupun dengan menggunakan cursor. Bantuan alat atau hardware seperti mouse, keyboard dan juga joystick dan lain sebagainya. Untuk pengimplementasian menu ini juga terdapat beragam jenis seperti teks, pop-up, konteks dengan kombinasi antara teks dan simbol dan juga dapat berupa pull-down menu. Dengan demikian, sistem navigasi dari menu juga akan beragam. Karena pembuatan menu ini pada dasarnya adalah untuk memudahkan interaksi antara program dan penggunaannya maka menu dibuat se fleksibel mungkin dan se ramah mungkin agar mudah dioperasikan oleh banyak pengguna.

Namun kebanyakan dari fitur navigasi yang disediakan dalam menu program, masih sebagian besar menggunakan tangan sebagai media untuk mengoperasikan opsi dari menu. Sehingga memiliki sedikit batasan, khususnya bagi pengguna atau user yang mengalami cedera pada organ tangan baik berat maupun ringan atau pengguna yang mengalami disabilitas fisik pada tangan. Dilansir dari data WHO (2011) data yang didapat memiliki berbagai varian usia dari mulai 0-14 tahun, 15-59 tahun, lebih dari 60 tahun dan total pada umur lebih dari 15 tahun (usia produktif). Pada data tersebut persentase jumlah didapatkan dari pengkategorian negara yang di bedakan berdasarkan tipe pendapatan, dalam data tersebut terdapat data dari negara berpendapatan tinggi dan negara berpendapatan sedang dan rendah. Dari data yang diperoleh dari WHO negara dengan tingkat pendapatan sedang dan rendah memiliki persentase penyandang disabilitas yang relatif lebih tinggi daripada negara

dengan tingkat penghasilan tinggi. Di negara Indonesia yang dilansir dalam Analisis LPEM FEB Universitas Indonesia (2016) didapat estimasi dari jumlah penduduk disabilitas 12,5% dari keseluruhan jumlah penduduk Indonesia. Dengan rincian 10,29% untuk penyandang tingkat sedang dan 1,87% untuk kategori berat. Dari persentase tersebut terdapat disabilitas fisik dalam hal ini disabilitas tangan yang menjadi konsentrasi dari penelitian ini. Dari persentase juga dapat diketahui bahwa di Indonesia memiliki jumlah penyandang difabel yang juga tidak sedikit dan dari penjabaran ini dapat menjadi alasan tentang pentingnya mengembangkan sistem teknologi yang mampu membantu sehingga penyandang disabilitas mendapat fasilitas yang sama dan dapat mengoperasikan suatu alat dengan mandiri sehingga ketika suatu teknologi dioperasikan secara umum, teknologi tersebut dapat mencakup berbagai kalangan masyarakat dan lebih luas. Penelitian tentang navigasi ini dirasa penting dalam hal pengembangan fitur teknologi dengan tujuan yang sama yakni mampu mencakup lebih banyak kalangan user dan halangan seperti keterbatasan fisik.

Untuk sistem navigasi ini diusulkan fitur yang memanfaatkan gerakan kepala sebagai sumber untuk melakukan proses navigasi pada sebuah menu program. Salah satu gestur tubuh yang dapat diaplikasikan yakni bagian kepala manusia. Kepala sering kali merupakan langkah pertama dalam aplikasi seperti pengamatan video, antarmuka human computer, face recognition, dan image database management (Kumbhar, 2017). Telah banyak penelitian yang diusulkan di bidang Pengenalan Wajah dan Deteksi Wajah untuk membuatnya lebih maju dan akurat, revolusi di bidang ini adalah ketika Viola-Jones menciptakan Real-Time Face Detector, yang mampu mendeteksi wajah secara real-time dengan akurasi tinggi. Proses analisis piksel untuk mendeteksi wajah yang cukup memakan waktu dan sulit untuk mencapai hasil yang sesuai keinginan karena perbedaan bentuk ukuran kepala dan pigmentasi kulit manusia. Maka pada penelitian ini digunakan metode yang menggunakan pola sementara sehingga dapat digunakan sebagai metode pengenalan gerakan dan jenis keilmuan bioinformatika (Starner, 1995). Juga dikarenakan kepala adalah organ paling penting dan merupakan organ dengan chance paling sedikit mengalami gangguan di sistem pergerakannya. Tanpa memandang ukuran, posisi dan kondisi termasuk warna, tekstur. Dengan dihubungkan pada modul raspberry pi dikembangkan dengan pergerakan kepala yang mampu melakukan proses navigasi menu yang ada pada objek yang diinginkan melalui tampilan LCD.

1.2 Rumusan masalah

Berdasarkan penjelasan pada latar belakang, maka dapat dirumuskan rumusan masalah sebagai berikut:

1. Bagaimana pengaruh tingkat pencahayaan dengan luminance antara (1200-1400) dan (1500-1800) pada proses rekognisi pergerakan kepala ?
2. Bagaimana pengaruh jarak pada pengimplementasian sistem rekognisi pergerakan kepala pada jarak 40cm, 50cm, 60cm serta 80cm ?

3. Bagaimana visualisasi masing masing rekognisi pergerakan kepala dan deteksi kepala pada sistem?

1.3 Tujuan

Berdasarkan perumusan masalah maka, tujuan dari penelitian adalah sebagai berikut :

1. Mengetahui dan menganalisa pengaruh tingkat pencahayaan pada proses rekognisi pergerakan kepala
2. Mengetahui dan menganalisa pengaruh jarak pada pengimplementasian sistem rekognisi pergerakan kepala pada jarak 40cm, 50cm, 60cm serta 80cm.
3. Mengetahui visualisasi masing masing rekognisi pergerakan kepala pada sistem.

1.4 Manfaat

Pada bagian ini akan menjelaskan manfaat dari sistem yang akan dibuat dan penjelasan

1. Pembuatan sistem ini dapat menghasilkan alat yang dapat melakukan navigasi menu hanya dengan pergerakan kepala.
2. Diharap dengan adanya sistem ini, penyandang disabilitas termudahkan dalam menggunakan fasilitas umum yang berbasis pilihan menu.
3. Peneliti mampu membuat sistem yang relevan untuk mengenali pergerakan pada kepala dengan color filtering HSV dan YcbCr berbasis pada pembagian cross area.

1.5 Batasan masalah

Adapun batasan masalah pada penelitian kali ini yaitu :

1. Sistem hanya mendeteksi pergerakan kepala melalui deteksi kulit YCbCr dan HSV .
2. Sistem hanya dapat melakukan navigasi pada menu aplikasi yang juga dibuat sendiri oleh penulis.
3. Pergerakan kepala saat pengujian hanya pergerakan keatas, kebawah, ke samping kiri dan samping kanan saja.
4. Jarak yang digunakan pada penelitian ditentukan oleh peneliti dan tingkat pencahayaan hanya memiliki 2 tingkat yang berbeda

1.6 Sistematika pembahasan

Sistematika pembahasan dalam skripsi ini sebagai berikut:

BAB 1 Pendahuluan

Pada tahap pendahuluan terdapat berbagai komponen yang mengawali kegiatan penelitian. Penjabaran meliputi latar belakang, rumusan masalah, tujuan, manfaat, serta batasan masalah. Berbagai komponen awal ini dijabarkan dan menjadi proses yang mengawali terjadinya penelitian.

BAB 2 Landasan Kepustakaan

Tahap bab ini merupakan bab yang berisikan tinjauan pustaka yang menjabarkan tentang penelitian yang sudah berjalan sebelumnya dan dasar teori yang merupakan kumpulan dari berbagai sumber sebagai bahan literatur penelitian.

BAB 3 Metodologi Penelitian

Pada tahap bab 3 penjabaran yang berlangsung adalah penjabaran tentang tipe penelitian yang berisikan pengenalan pada jenis penelitian yang dilaksanakan dan Strategi dan Rancangan penelitian yang berisikan informasi tentang subjek, lokasi dan metode pengumpulan pada penelitian, metode analisis data serta penjabaran tentang peralatan pendukung penelitian.

BAB 4 Analisis Kebutuhan

Tahap bab analisis kebutuhan dibagi menjadi 3 poin yaitu deskripsi umum sistem, analisis kebutuhan yang dibagi menjadi kebutuhan fungsional dan kebutuhan non fungsional dan yang terakhir adalah analisis kebutuhan perangkat .

BAB 5 Perancangan dan Implementasi

Pada tahap ini merupakan suatu tahapan yang berisikan gambaran umum sistem yang berfungsi sebagai *blueprint* seluruh komponen sistem baik software dan hardware yang dirangkai pada penelitian.

BAB 6 Pengujian dan Analisis

Tahap ini merupakan bab yang menjelaskan tahap pengujian dari proses yang telah dilaksanakan. Berdasarkan parameter pengujian yang telah dicantumkan dalam rumusan masalah.

BAB 7 Kesimpulan dan Saran

Tahapan bab ini merupakan penjabaran dari seluruh program yang telah dirancang dan diuji, serta menguji perbedaan antara hasil yang telah didapatkan dan dianalisis dengan perumusan dan tujuan penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Pada umumnya , terdapat penelitian yang saling mempunyai keterkaitan dengan penelitian ini, keterkaitan berupa kesamaan pada teknologi yang digunakan, maupun secara metode serta environment. Berikut adalah penelitian yang dinilai penulis memiliki keterkaitan terhadap penelitian ini.

2.1 Tinjauan Pustaka

Adapun beberapa penelitian sebelumnya yang menjadi literatur penulis dalam mengerjakan laporan ini yang berkaitan dengan analisis pergerakan kepala.

JUDUL	KETERANGAN
Real Time Face Detection And Tracking Using OpenCV	Penelitian pertama adalah penelitian dari Prof. P Y Kumbhar, dkk (2017) yang berjudul "Real Time Face Detection And Tracking Using OpenCV" yang mengimplementasikan pendeteksi wajah secara real-time dan melakukan <i>tracking</i> posisi kepala melalui video berkualitas tinggi menggunakan Raspberry-Pi. Library OpenCV dan SimpleCV digunakan dalam pengerjaan penelitian ini menggunakan Haar Cascade dalam proses tracking tipe wajah dengan memilih fitur wajah yang memiliki kriteria tertentu. Kemudian menggabungkan fitur yang telah di filter tersebut dengan algoritma AdaBoost sehingga mampu menentukan klasifikasi tracking kepala pada target ataupun objek penelitian. Jadi, pertama gambar akan di tangkap oleh kamera untuk kemudian dicocokkan dengan Database kemudian jika cocok maka di verifikasi sebagai objek wajah untuk kemudian di ekstraksi kembali untuk penentuan fitur objek wajah tersebut. Jika terdapat kesamaan dengan template maka hasil akan ditampilkan pada proses video <i>real-time</i> tersebut. Kekurangan dalam penelitian ini yaitu

	<p>penggunaan Haar-Cascade untuk menentukan tracking sehingga data latih yang diperlukan akan sangat banyak bergantung pada seberapa banyak klasifikasi dan tipe wajah objek penelitian.</p>
<p>Real Time Hand Gesture Recognition for Human Machine Communication Using ARM Cortex A-8</p>	<p>Penelitian selanjutnya dari Vignesh. SPG Scholar (2014) yang menggunakan Haar Classifier sebagai metode untuk aplikasi pengenalan pergerakan tangan yang ditampilkan dalam HDMI monitor. Menurut peneliti berbagai variasi gerakan yang semakin banyak di masukkan dalam database data uji maka membuat <i>computer vision</i> mampu mengukur tiap pergerakan tersebut. Tujuan dari peneliti ialah menciptakan “<i>Mouse Less</i>” atau mouse kasat mata yang mampu beroperasi tanpa hardware dalam bentuk mouse fisik. Penelitian menggunakan BeagleBoard sebagai mini cpu dengan integrasi sistem operasi ubuntu dan objek ditangkap melalui kamera USB untuk kemudian pengenalan pola yang ditangkap oleh kamera USB diproses oleh board sehingga mampu membuat keputusan dalam hal ini pergerakan pointer mouse. Pada penelitian ini objek penelitian menggunakan tangan dan Haar Cascade.</p>
<p>Human Skin Detection Using RGB, HSV and YCbCr Color Models</p>	<p>Berikutnya ialah penelitian dari S.Kolkur, dkk (2017) yang membahas tentang deteksi kulit manusia menggunakan metode RGB, HSV dan YCbCr menurut penelitiannya, kunci utama dari deteksi pengenalan kulit manusia adalah berdasarkan pada warna kulit. Tapi warna dalam penelitian ini tidak hanya difaktorkan dari jenis perbedaan warna dan ras dari variabelnya. Selain hal tersebut</p>



masih ada faktor lain yakni tingkat pencahayaan . Untuk mengurangi *Threshold* maka peneliti menggunakan metode penggabungan antara ketiga metode tersebut menjadi satu. Proses yang digunakan ialah pertama sistem menampilkan atau *load* gambar sampel, untuk kemudian dikonversi kedalam *array* 2 dimensi piksel untuk kemudian dimasukkan kedalam 2 metode HSV dan YCBCR. Data hasil yang ditampilkan oleh peneliti memiliki tingkat akurasi yang tinggi sekitar 85-90 persen. Namun pada penelitian ini objek yang digunakan masih berupa gambar random yang didapatkan oleh peneliti melalui google dan bukan berupa video real-time .

Berdasarkan beberapa penelitian diatas terdapat beberapa kesimpulan yakni dari Prof. P Y Kumbhar, dkk (2017) Kekurangan dalam penelitian ini yaitu penggunaan Haar-Cascade untuk menentukan tracking sehingga data latih yang diperlukan akan sangat banyak bergantung pada seberapa banyak klasifikasi dan tipe wajah objek penelitian. Sehingga program tidak mampu menentukan sendiri klasifikasi data secara otomatis dan ketika muncul klasifikasi baru maka data latih harus ditambahkan kembali. Kurang efisiensi inilah sehingga perlu digunakan metode lain yang memungkinkan deteksi objek kulit secara otomatis tanpa perlu memasukkan banyak data latih kedalam sistem. S.Kolkur, dkk (2017) menggunakan metode yang cukup baik dalam hal definisi objek berdasarkan warna kulit yakni YCbCr dengan pengurangan *threshhold* dari pencahayaan objek menggunakan HSV namun pada penelitiannya objek yang digunakan masih berupa gambar dan bukan proses secara real-time video. Dari Vignesh. SPG Scholar (2014) sendiri pengklasifikasian gerakan dengan penampilan output pada monitor namun masih menggunakan objek berupa tangan dan membutuhkan daya latih karena menggunakan Haar-Cascade. Pada penelitian ini akan menggunakan deteksi pergerakan kepala dengan menggunakan metode HSV dan YCbCr sehingga tidak memerlukan banyak data latih dan output ditampilkan kedalam menu LCD yang berjalan secara real-time.

2.2 Dasar Teori

Penulisan persamaan, tabel, gambar, dan symbol-simbol memiliki aturan khusus seperti yang dijelaskan dalam seksi-seksi berikut.

2.2.1 Disabilitas

Disabilitas menurut sumber disability.wa.gov.au merupakan segala kondisi berkelanjutan yang membatasi aktivitas sehari-hari. The Disability Service Act (1993) juga mendefinisikan disabilitas sebagai beberapa kondisi diantaranya :

- 1) Yang disebabkan oleh gangguan intelektual, kejiwaan, kognitif, neurologis, sensorik atau fisik atau kombinasi dari gangguan tersebut
- 2) Yang permanen atau cenderung permanen
- 3) Yang mungkin atau mungkin tidak bersifat kronis atau episodik
- 4) Yang menghasilkan kapasitas orang secara substansial berkurang untuk komunikasi, interaksi sosial, pembelajaran atau mobilitas dan kebutuhan akan layanan berkelanjutan

Dikutip dari situs WHO pada tahun 2011 terdapat data berupa tabel yang menunjukkan kondisi penyandang disabilitas pada berbagai negara di dunia. Dalam tabel berikut ini diambil sampel negara dengan dibagi menjadi 2 tipe dasar berdasarkan tingkat pendapatan yakni, negara dengan tingkat pendapatan tinggi dan negara berpendapatan sedang dan rendah yang juga mencakup daerah WHO. Dari persentase yang disebutkan terdapat disabilitas tangan yang menjadi konsentrasi dari penelitian ini sehingga data tabel dibawah penting sebagai referensi data disabilitas. Negara berpendapatan sedang dan rendah mencakup 6 wilayah besar yaitu : Afrika, Amerika, Asia Tenggara, Eropa, Mediterania Timur, dan Pasifik Barat. Data penyandang disabilitas juga dibagi dalam 2 kategori Disabilitas kategori berat juga sedang dan berat yang memiliki rentang umur objek observasi yakni umur 0 sampai 14 tahun, 15 sampai 59 tahun dan lebih dari 60 tahun. Berikut adalah hasil observasi tersebut :

Tabel 2.1 Data Penyandang Disabilitas pada tahun 2011 dikutip dari situs WHO

Usia	Dunia	Dalam Persen (%)						
		Negara berpendapatan tinggi	Negara berpendapatan sedang dan rendah , daerah WHO					
			Afrika	Amerika	Asia Tenggara	Eropa	Mediterania Timur	Pasifik Barat
Disabilitas tingkat berat								
0-14 tahun	0,7	0,4	1,2	0,6	0,7	0,8	0,9	0,5
15-59 tahun	2,7	2,3	3,3	2,6	2,9	2,7	3,0	2,4
≥ 60 tahun	10,2	8,5	16,9	9,2	12,6	7,2	12,4	10,0
≥ 15 tahun	3,8	3,8	4,5	3,4	4,0	3,6	3,9	3,4
Semua umur	2,9	3,2	3,1	2,6	2,9	3,0	2,8	2,7
Tingkat sedang dan berat								
0-14 tahun	5,1	2,8	6,4	4,5	5,2	4,2	5,2	5,3
15-59 tahun	14,9	12,4	19,1	14,6	16,3	14,3	15,5	13,7
≥ 60 tahun	46,1	36,8	53,3	44,3	58,8	41,4	53,7	46,7
≥ 15 tahun	19,4	18,3	22,0	18,3	21,1	19,5	19,1	18,1
Rata-rata semua umur	15,3	15,4	15,3	14,1	16,0	16,4	14,0	15,0

Sumber : (<https://www.who.int/>)

2.2.1.1 Jenis - jenis Disabilitas

Kategori utama disabilitas adalah fisik, sensorik, psikiatris, neurologis, kognitif, dan intelektual. Kecacatan fisik adalah jenis kecacatan yang paling umum, diikuti oleh kecacatan intelektual dan indera. Cacat fisik umumnya



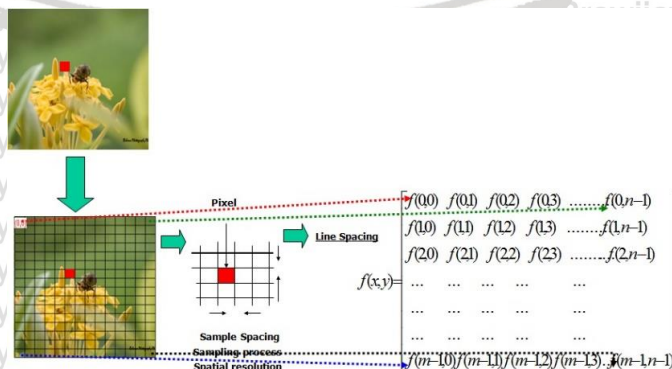
berkaitan dengan gangguan pada sistem muskuloskeletal, sirkulasi, pernapasan, dan saraf. Cacat indera melibatkan gangguan pendengaran dan penglihatan. Kecacatan neurologis dan kognitif termasuk kecacatan yang didapat seperti multiple sclerosis atau cedera otak traumatis. Kecacatan intelektual mencakup kecacatan intelektual dan perkembangan yang berkaitan dengan kesulitan dengan proses berpikir, belajar, berkomunikasi, mengingat informasi dan menggunakannya dengan tepat, membuat penilaian dan penyelesaian masalah. Kecacatan intelektual adalah hasil interaksi antara gangguan kognitif yang disebabkan oleh perkembangan, hambatan sikap dan lingkungan. Gangguan kejiwaan yang mengakibatkan kecacatan mungkin termasuk kelainan kecemasan, fobia atau depresi.

2.2.2 Citra Digital

Pengertian pengolahan citra digital adalah proses yang ditujukan guna memanipulasi juga melakukan proses analisis citra menggunakan bantuan komputasi komputer. Pengolahan citra digital dikelompokkan menjadi dua jenis kegiatan (Iswindarty.Peni, 2013):

1. Memperbaiki kualitas gambar, sehingga memudahkan proses interpretasi oleh mata manusia.
2. Sebagai pengolah informasi yang terdapat di suatu gambar sebagai keperluan pengenalan objek otomatis.

Citra digital merupakan fungsi intensitas cahaya $f(x,y)$ dalam bidang berbentuk 2D, perhitungan x dan y menggambarkan koordinat spasial dan perhitungan fungsi tersebut di setiap titik (x,y) menunjukkan tingkat keberhasilan citra pada titik tersebut. Citra digital merupakan citra $f(x,y)$ yang mana akan dilakukan diskritisasi koordinat sampling dan diskritisasi tingkat kwantisasi. Citra digital menggambarkan sebuah matriks dimana ketika indeks baris dengan kolom memberitahukan sebuah titik di citra dan elemen matriks menggambarkan tingkat keabuan pada titik area tersebut. Citra digital bisa dibayangkan berbentuk 4 persegi panjang lalu dimensi ukurannya dinyatakan dengan tinggi x lebar. Citra digital yang berukuran $N \times M$ dinyatakan dengan sebuah matriks berukuran N untuk baris dan M untuk kolom. Setiap elemen yang ada di citra digital dapat disebut sebagai piksel.



Gambar 2.1 Elemen Matriks pada Piksel



Sumber : (<https://pemrogramanmatlab.com/>)

Sebuah citra berukuran $N \times M$ pixel dimana memiliki intensitas yang beragam pada setiap pixel tersebut, hal ini direpresentasikan dalam bentuk numerik dan bentuk matriks berukuran N baris dan M kolom.

B (0-255)								G (0-255)								R (0-255)											
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}

Sehingga Total Jumlah RGB = 16,777,216 (256^3).



B (0-255)								G (0-255)								R (0-255)											
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}



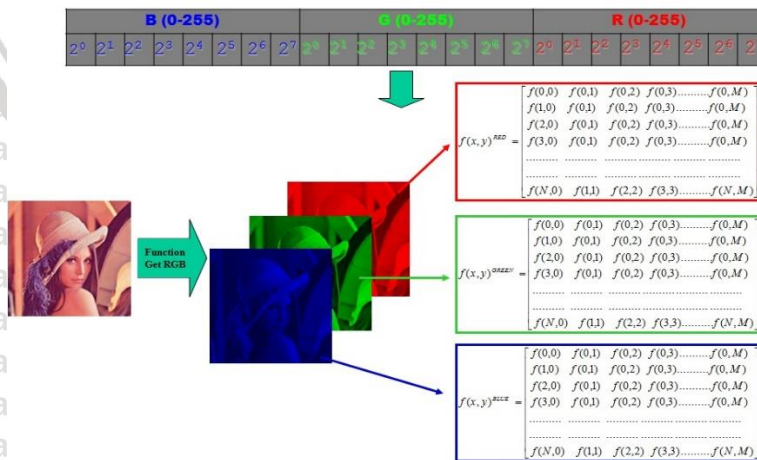
Gambar 2.2 Pembagian nilai RGB pada Citra Digital

Sumber : (<https://pemrogramanmatlab.com/>)

Dari gambar di atas dalam satu titik citra memiliki warna Red, Green, Blue.

Dapat di sederhanakan menjadi 1 citra yang berukuran M Baris x N Kolom yang memiliki kapasitas berupa 3 penyimpanan dalam sebuah bentuk matrik yang berjenis :

1. Data_Citra_Red(M Baris x N Kolom), sebagai penyimpanan perhitungan diantara antara 0-255 kategori Red
2. Data_Citra_Green(M Baris x N Kolom), sebagai penyimpanan perhitungan diantara antara antara 0-255 kategori Green
3. Data_Citra_Blue(M Baris x N Kolom), sebagai penyimpanan perhitungan diantara antara antara 0-255 kategori Blue



Gambar 2.3 Matriks pada nilai RGB.

Sumber : (<https://pemrogramanmatlab.com/>)



Bidang aplikasi yang kedua sangat berhubungan dengan pattern recognition yang ditujukan sebagai pengenalan objek dengan mengekstrak informasi yang penting dalam suatu citra. Bila sistem pengenalan pola terhubung dengan pengolahan citra, maka akan terbentuk sistem yang mampu memproses citra masukan hingga citra tersebut dapat mengenali suatu pola. Proses inilah yang disebut dengan pengenalan citra (Conzales, 2007). Pengolah citra adalah fitur penting dalam penelitian yang dilakukan.



Gambar 2.4 Logo OpenCV.

Sumber : (<https://opencv.org>)

OpenCV (Open Source Computer Vision Library) dirilis di bawah lisensi BSD dan gratis untuk penggunaan akademis dan komersial. Library ini mendukung bahasa pemrograman C ++, Python dan antarmuka Java dan mendukung Windows, Linux, Mac OS, iOS dan Android. OpenCV dirancang untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi real-time. Ditulis dalam C / C ++ yang dioptimalkan, perpustakaan dapat memanfaatkan pemrosesan multi-core. Diaktifkan dengan OpenCL, dapat memanfaatkan akselerasi hardware dari platform komputasi heterogen yang mendasarinya (Pulli et al. 2012).

2.2.3 Macam - macam Warna pada Citra Digital

Secara default parameter warna yang digunakan oleh kamera saat gambar di rekam ialah berupa gambar dalam warna RGB dengan masing masing nilai pada warna Red, Green dan Blue. Warna ini masih butuh disederhanakan kembali dalam bentuk ekstraksi ataupun diganti dengan parameter yang lebih sederhana guna meringankan analisis untuk mengidentifikasi objek ataupun ekstraksi pada citra. Dalam penelitian ini menggunakan 2 macam konversi warna yang akan digunakan yaitu YcbCr dan HSV.

2.2.3.1 RGB

Suatu model warna yang didalamnya terdiri dari 3 buah warna yaitu merah, hijau, dan biru, dimana kemudian ditambahkan berbagai komposisi untuk menghasilkan warna itu adalah RGB. RGB biasanya digunakan untuk menampilkan citra di dalam sebuah perangkat elektronik. Kelebihan dari RGB adalah gambar yang mudah dipindah ke alat lain tanpa harus di ubah ke mode

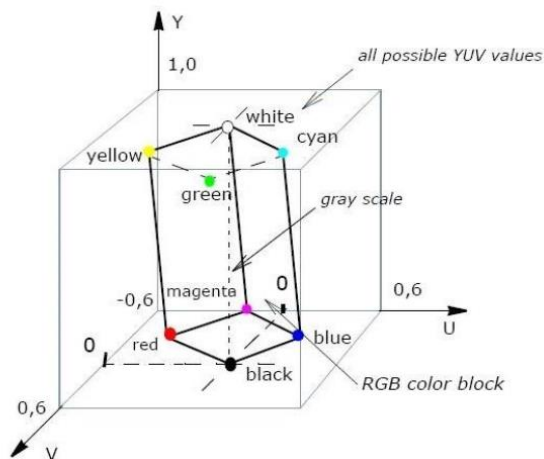
warna lain, karena sudah banyak peralatan yang telah memakai mode warna ini. Kelemahan dari RGB adalah tidak bisa untuk mencetak dengan sempurna di printer karena printer menggunakan mode CMYK sehingga harus mengubahnya terlebih dahulu.

Metode warna sering dipakai untuk warna layar yang merupakan cahaya yang sedang dipancarkan. Warna ini bisa dibilang hampir mirip dengan teori warna dasar, akan tetapi menggunakan warna Hijau daripada warna Kuning. Pemberian hitungan maksimal untuk RGB yaitu bernilai RGB (255,255,255) atau RGB(FF,FF,FF) yang akan menghasilkan warna putih. Sedangkan pemberian perhitungan minimal RGB(0,0,0) akan menjadi warna hitam. Untuk menghasilkan warna abu-abu, cukup memberikan perhitungan yang sama di unsur R, G, dan B. Seperti misalnya : RGB(20,20,20) menghasilkan abu-abu gelap dan RGB(200,200,200) akan menghasilkan warna abu-abu yang lebih cerah.

2.2.3.2 YCBCR

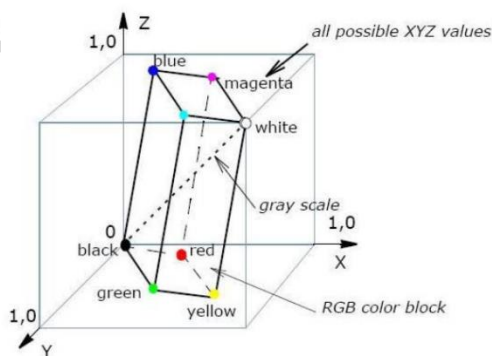
Pengolahan citra yang dilakukan pada pengujian ini menggunakan ruang warna *YCbCr* yang umumnya banyak digunakan dalam pemrosesan citra gambar maupun video. Pengolahan citra yang menggunakan lingkup *YCbCr* ini memiliki *channel* warna sendiri, dengan *Y* sebagai komponen tunggal untuk pengaturan cahaya atau *luminance*, *Cb* dan *Cr* sebagai komponen pewarnaan berupa warna pada *channel* biru untuk *Cb* dan *channel* merah pada *Cr* yang tergolong dalam komponen warna *chrominance*.

Menurut (Ghinmine, 2017) Penggunaan model pewarnaan *YCbCr* memiliki skala data sebesar 8-bit dengan nominal 16-235 untuk *channel* warna yang dimiliki oleh *Y* sedangkan pada *Cb* dan *Cr* memiliki skala dengan nominal 16-240. Pengembangan dari pewarnaan *YCbCr* banyak digunakan sebagai klasifikasi dari pewarnaan kulit, pendeteksian warna dasar kematangan buah, hingga pendeteksian warna api dan masih banyak yang lain. Dalam penelitian ini, ruang warna *YCbCr* digunakan sebagai bahan pendeteksi kulit dimana kulit yang telah terdeteksi digunakan sebagai navigasi menu pada LCD. Penggunaan lingkup warna *YCbCr* untuk pengujian sistem ini dapat meniru penglihatan mata manusia karena sifat dari mata manusia yang peka terhadap intensitas cahaya ketimbang warna. Hal ini lah yang digunakan penguji sebagai bahan untuk pemrosesan citra secara real-time karena kebutuhan akan informasi yang diperlukan lebih banyak. Ilustrasi dari implementasi ruang warna *YCbCr* pada plane *YUV* dapat ditunjukkan pada Gambar 2.5 sedangkan Gambar 2.6 merupakan representasi plane *XYZ*.



Gambar 2.5 Representasi model warna YCbCr dalam Plane YUV

Sumber : Ibraheem (2012)



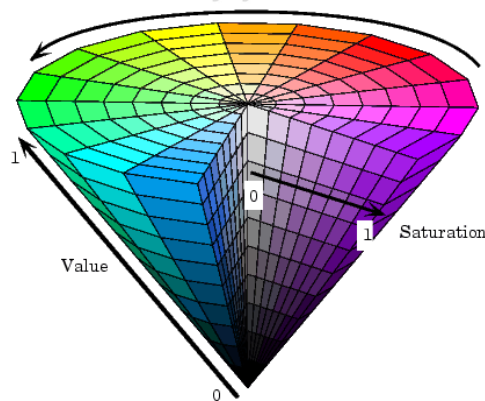
Gambar 2.6 Representasi model warna YCbCr dalam Plane XYZ

Sumber : Ibraheem (2012)

2.2.3.3 HSV

Salah satu pemodelan warna yang dilakukan untuk pengujian pada sistem ini yaitu menggunakan HSV atau *Hue Saturation Value*. HSV merupakan dasar dari perubahan warna RGB dimana nilai dari channel warna RGB ini memiliki beda lebih kesegi pencahayaannya. Bagian pewarnaan pada nilai Hue memiliki komposisi warna dengan rentang nilai derajat mulai dari 0 hingga 360 derajat pada bagian pemilihan warnanya. Nilai saturation pada HSV memiliki komposisi warna berupa tingkat kecerahan warna, dengan skala nilai 0 sampai dengan 100. Kecarahan warna yang digunakan pada saturation berupa tingkatan warna mulai dari warna putih hingga warna yang dituju. Pembagian warna HSV pada bagian value hampir sama dengan penggunaan saturation, perbedaannya terletak jika pemilihan warna menggunakan nilai value dilakukan mulai tingkat warna gelap atau hitam menuju ke warna yang dituju. Pewarnaan komposisi dari hue memiliki struktur ruang pixel yang menyangkut pewarnaan seperti pewarnaan pada pelangi. Komposisi yang digunakan pada ruang pewarnaan saturation

dinyatakan dengan tingkat akurasi kekuatan warna. Nilai *value* yang dibutuhkan pada komposisi warna *HSV* memiliki rentang 0-100% pada tingkat kecerahan warnanya. Semakin besar tingkat kecerahan warnanya, semakin banyak pilihan variasi pewarnaan dari *HSV*. Ketiga komponen pewarnaan *HSV* sangat cocok digunakan untuk pengolahan citra yang membutuhkan tingkat kecerahan warna seperti halnya pendeteksian kulit karena warna dari kulit memiliki tingkat kecerahan yang berbeda-beda. Gambar 2.7 ilustrasi skala pewarnaan dari model *HSV*



Gambar 2.7 Representasi model warna HSV

Sumber : northwestern.edu (2018)

Pewarnaan pada ruang warna *HSV* memiliki bentuk menyerupai kerucut terbalik dengan pusat sudut berwarna putih dan berwarna hitam pada bagian ujung dari pemvisualan seperti pada Gambar 2.7. Bentuk kerucut paling sering digunakan sebagai pengolahan citra karena berkaitan dengan desain grafis komputer (Ghinmine, 2017). Rentang sekala pada ruang warna *HSV* dapat dilihat pada Tabel 2.2 berikut

Tabel 2.2 Komposisi warna berdasarkan ruang warna HSV

Sudut	Color
0-60	Red
60-120	Yellow
120-180	Green
180-240	Cyan
240-300	Blue
300-360	Magenta

2.2.4 Konversi ruangan warna

2.2.4.1 Konversi warna RGB ke HSV

Konversi warna rgb ke hsv memiliki nilai transformasi komposisi yang digunakan pada trAVIS, dapat ditunjukkan pada Persamaan (2.2)sampai (2.3)

$$\begin{aligned} R' &= \frac{R}{255} \\ G' &= \frac{G}{255} \\ B' &= \frac{B}{255} \end{aligned} \tag{2.1}$$

Pada pengkondisian dalam komposisi RGB yang akan dirubah menjadi HSV, persamaan (2.1) diperlukan untuk merubah komposisi elemen warna merah, hijau dan biru dibagi dengan nilai 255 dimana nilai tersebut adalah satuan dari nilai maksimal berukuran 8 bit. Selanjutnya, hal yang diperlukan dalam perubahan warna ini, menggunakan komposisi nilai dengan variabel C_{max} yang diperoleh dari hasil perhitungan R', G', B' pada persamaan (2.1). Perumusan C_{max} dapat dilihat pada persamaan (2.2) berikut

$$\begin{aligned} C_{max} &= \max(R' \times G' \times B') \\ C_{min} &= \min(R' \times G' \times B') \\ \Delta &= C_{max} - C_{min} \end{aligned} \tag{2.2}$$

Setelah nilai delta didapatkan dari selisih pengurangan antara C_{max} dengan C_{min} , komposisi nilai dari Hue bisa didapatkan. Cara perolehan nilai Hue dapat dihasilkan dari persamaan (2.3) berikut

$$H = \begin{cases} 0 & , \Delta = 0 \\ 60 \times \frac{G' - B'}{\Delta} \text{ mod } 6 & , C_{max} = R' \\ 60 \times \frac{B' - R'}{\Delta} + 2 & , C_{max} = G' \\ 60 \times \frac{R' - G'}{\Delta} + 4 & , C_{max} = B' \end{cases} \tag{2.3}$$

Dari persamaan (2.3) diatas, kondisi dari nilai H , mempunyai beberapa kondisi yang berbeda-beda menyesuaikan hasil dari nilai C_{max} . Jika hasil seleksi delta C_{max} dikurangi dengan C_{min} mengeluarkan nilai 0, maka nilai H juga menjadi nilai 0. Pada pencarian nilai saturation, memiliki 2 kondisi dimana kondisi pertama jika nilai C_{max} tidak sama dengan 0 maka variabel delta akan dibagi dengan nilai C_{max} . Kondisi kedua yaitu jika nilai komposisi pada C_{max} bernilai 0 maka otomatis nilai S juga berwarna 0 atau warna yang dihasilkan berwarna putih. Persamaan ini dapat dirumuskan sebagai persamaan (2.4) berikut

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases} \tag{2.4}$$



Perhitungan segmentasi pada bagian *RGB* ke *value* dari *HSV* menggunakan nilai yang didapat dari *Cmax*. Nilai dari *Cmax* sendiri merupakan representasi dari tingkat kecerahan warna. Rumus dari nilai *value HSV* dapat disimpulkan menjadi persamaan (2.5) berikut

$$V = Cmax \quad (2.5)$$

2.2.4.2 Konversi ruangan warna *RGB* ke *YCbCr*

Pada pengujian sistem ini, menggunakan segmentasi perubahan warna *RGB* ke *YCbCr* sebagai bahan untuk pendeteksian kulit. Pewarnaan *YCbCr* memiliki kekurangan pada warna yang dihasilkan, karena pewarnaan *YCbCr* bukanlah warna yang absolut. Proses perubahan warna dari *RGB* ke *YCbCr* merupakan hasil pengkodean wujud dari warna *RGB*. Warna tampilan yang sebenarnya menyesuaikan dengan warna-warna primer penyusun *RGB*. Pada Persamaan (2.6) merupakan perumusan untuk segmentasi warna dari *RGB* ke lingkup warna *YCbCr*

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299R & + 0.587G & + 0.114B \\ -0.16874R & + 0.33126G & + 0.5B \\ 0.5R & - 0.41869G & + 0.0813B \end{bmatrix} \begin{bmatrix} +16 \\ +128 \\ +128 \end{bmatrix} \quad (2.6)$$

2.2.5 Deteksi Kulit

Skin detection adalah salah satu proses segmentasi yang digunakan untuk memisahkan region pada objek di dalam citra yang berdasarkan dari perbedaan warna. Objek yang memiliki warna tertentu dipisahkan dari objek yang memiliki warna berbeda. Nantinya hasil dari segmentasi digunakan sebagai proses selanjutnya yaitu klasifikasi citra. Warna kulit didefinisikan dalam ruang warna *YCbCr* dengan nilai *Cb* antara 77 sampai dengan 127 dan nilai *Cr* di antara 133 sampai dengan 173 (S.Kolkur,2017).

Deteksi kulit adalah salah satu tahapan awal dalam computer vision yang digunakan sebagai mendeteksi hal yang berkaitan dengan manusia. *Skin detection* digunakan sebagai metode segmentasi yang berguna sebagai pengenalan wajah ataupun anggota tubuh lainnya.

Langkah proses segmentasi warna kulit sebagai berikut:

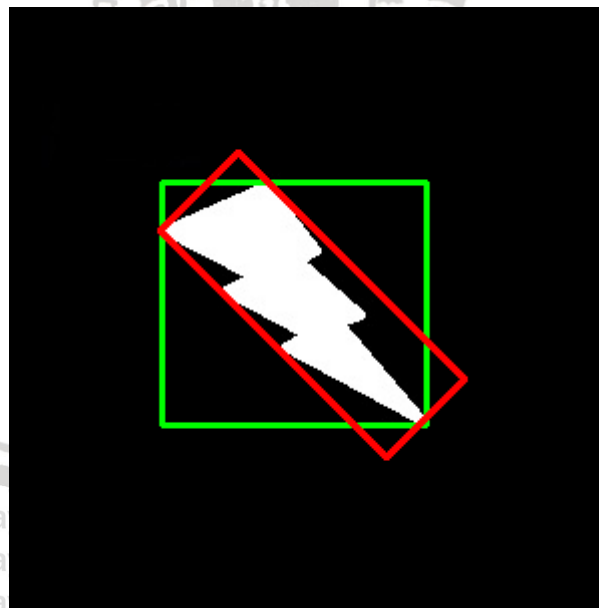
1. Melakukan penyeimbangan warna *RGB*
2. Melakukan transformasi ruang warna *RGB* menjadi *YCbCr*
3. Melakukan segmentasi warna kulit berdasarkan nilai *Cb* antara 77 sampai dengan 127 dan nilai *Cr* antara 133 sampai dengan 173
4. Menampilkan hasil segmentasi

Di dalam pendeteksian citra, warna memiliki kepekaan yang cukup tinggi bila terjadi perubahan cahaya, untuk mengatasinya maka dilakukanlah transformasi citra *RGB* ke dalam ruang warna yang memiliki komponen luminasi

dan kromatiknya dipisahkan sehingga cukup digunakan kromatik saja untuk proses deteksi warna kulit. Bila ingin mendapatkan nilai area kulit dibangun suatu pengklasifikasi piksel yang menunjukkan yang mana kulit dan mana yang bukan kulit. Ketika ingin membuat model warna kulit diperlukan sampel warna kulit yang mana diperoleh dari beberapa potongan kulit dari sejumlah citra. Labelisasi dilakukan kepada kulit yang sudah tersegmentasi yang akan dievaluasi terhadap berbagai macam karakter berbeda dari suatu wajah. Dengan menggunakan metode statistic sederhana dilakukan sebuah pembagian kelas pada pencahayaan kepada sampel warna kulit hingga diketahui jangkauannya atau jarak untuk kulit dengan berbagai macam kondisi pencahayaan. Bila ingin mensegmentasi daerah kulit manusia dengan daerah yang bukan kulit dengan berdasarkan warna, dibutuhkan model warna kulit yang nantinya disesuaikan pada warna kulit manusia yang tentunya bervariasi warnanya dan juga kondisi tingkat pencahayaan.

2.2.6 Bouding Rect

Bouding Rect atau yang dikenal juga sebagai bounding box merupakan kotak imajiner yang berada di sekitar objek yang sedang menjadi konsentrasi ataupun objek yang sedang di periksa. Bouding rect memiliki sistem koordinat 2D dan juga 3D . Dalam pemrosesan digital, bounding box berbentuk persegi yang menjadi wadah objek deteksi yang memiliki garis panjang dan lebar.



Gambar 2.8 Bouding Rect

Sumber : (<https://opencv.org>)

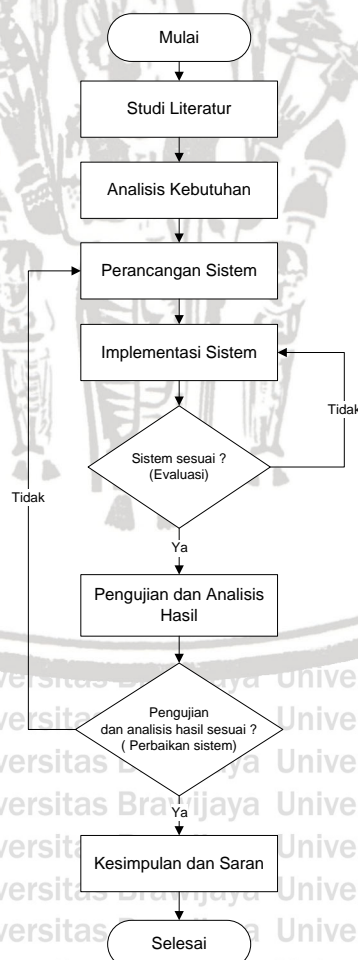
BAB 3 METODOLOGI

3.1 Tipe Penelitian

Penelitian ini merupakan penelitian bidang skripsi Rekayasa Perangkat Cerdas (RPC) dengan tipe penelitian berupa Implementatif Pengembangan sebuah kegiatan penelitian implementatif untuk membuat sebuah alat deteksi pergerakan kepala sebagai navigasi pada menu LCD dengan menerapkan konversi warna HSV dan YCbCr dengan metode deteksi tepi wajah meliputi analisis, perancangan, konstruksi dan pengujian.

3.2 Strategi dan Rancangan Penelitian

Tahapan awal dari proses penelitian ini adalah dengan studi literatur terkait dengan kajian pustaka serta dasar teori. Sifat penelitian ini merupakan implementatif. Penentuan alur penelitian guna langkah yang akan ditempuh sebagai tahapan dalam menyelesaikan penelitian agar berjalan sistematis. Tahapan alur metode proses penelitian yang ditempuh dapat dilihat pada diagram alir Gambar 3.1



Gambar 3.1 Diagram Alir Metodologi Penelitian.

Pada penelitian ini tahapan studi literatur akan menjadi suatu landasan perancangan sistem dan implementasi pada sistem. Tahapan studi literatur berisi teori yang akan membantu penelitian seperti materi tentang pemrosesan citra sebagai dasar pengetahuan tentang pemrosesan gambar, dengan menggunakan perpaduan YCbCr dan HSV sebagai pemroses deteksi warna. Kemudian deteksi tepi Canny Edge sebagai dasar dari metode inti dari pemrosesan gambar yang diimplementasikan ke dalam system dan sebagai parameter dari definisi tipe gerakan yang akan diproses, dan Raspberry Pi sebagai modul utama pemrosesan data dan visual studio sebagai aplikasi untuk membuat tampilan menu nantinya berbasis visual basic dan LCD display sebagai *output*. Dan juga membantu pada tahap implementasi sistem menentukan keluaran sistem.

Pada tahapan analisis kebutuhan dilakukahi macam-macam kebutuhan sistem agar sistem dapat bekerja sesuai tujuan sistem. Kebutuhan sistem dari penelitian ini antara lain rekayasa kebutuhan secara fungsional terdiri dari kebutuhan antarmuka perangkat keras juga perangkat lunak. Lainnya merupakan kebutuhan non fungsional yang merupakan batasan sistem supaya sistem mampu bekerja secara baik dan tidak mejadi gangguan dari kebutuhan fungsional.

Pada tahapan perancangan sistem, proses dilakukan sebelum implementasi supaya penelitian dapat dilakukan berjalan efektif dan efisien. Hal pertama yaitu dengan membuat diagram blok sistem yang tujuannya digunakan untuk merancan sistem sebagai rancangan komunikasi system dan deteksi pada objek. Bagian kedua merupakan alur program yang berkomponen macam perintah untuk menjalankan sistem.

Implementasi merupakan proses dari perancangan suatu sistem penelitian hingga hasil akhir penelitian. Tahapan implementasi sistem pada penelitian ini adalah implementasi pemrograman python dengan metode deteksi tepi dalam menentukan pola gerakan kepala dan aplikasi berbasis visual basic sebagai objek dari penelitian.

Pada tahapan pengujian dan analisis proses yang berjalan adalah pengecekan berjalannya sistem agar dapat berjalan dengan baik dan sesuai dengan kehendak spesifikasi penulis sesuai kebutuhan yang ditetapkan.

Hasil yang telah diperoleh dan juga proses analisis merupakan tahapan selanjutnya yaitu kesimpulan penelitian. Kesimpulan penelitian ini merupakan ringkasan proses berjalan dan tahapan yang telah dilaksanakan saat proses penelitian. Guna membandingkan relevansi antara rumusan masalah dengan hasil yang telah diperoleh maka poin ini dinilai penting sebagai salah satu parameter penelitian.

Kesimpulan dan saran merupakan tahap akhir dari laporan penelitian yang juga tahapan selanjutnya dari proses analisis kebutuhan, implementasi

serta pengujian. Kesimpulan merupakan jawaban dari rumusan masalah yang diujikan melalui pengujian sistem. Dan saran merupakan sub bab terakhir dengan berisi usulan yang dapat menjadi parameter penelitian selanjutnya jika ingin dilakukan pengembangan pada program. Dengan adanya kesimpulan dan saran maka proses penelitian dianggap sudah selesai dan telah dilaksanakan hingga akhir.

3.2.1 Subjek Penelitian

Subjek yang digunakan pada penelitian tidak terfokus pada gender atau kondisi tertentu, objek diklasifikasikan laki-laki dan perempuan dengan jumlah 10 untuk menguji ke akuratan sistem ditambahkan aksesoris pada subjek. Dengan masing-masing subjek diujikan berdasarkan 4 klasifikasi gerakan yaitu kanan, kiri, menunduk dan tegak. Dan pengujian juga dilakukan pada tingkat pencahayaan yang berbeda serta jarak yang berbeda untuk menguji sistem deteksi berdasarkan jarak subjek dan *luminance* tingkan pencahayaan yang paling tepat untuk digunakan.

3.2.2 Lokasi Penelitian

Lokasi penelitian dilaksanakan pada ruang tertutup (*indoor*) dengan tujuan untuk memastikan tingkat pencahayaan antar subyek tidak berbeda antar satu dan lainnya. Lokasi yang digunakan hanya pada satu lokasi ruang berukuran studio dengan sumber pencahayaan dengan lampu yang memiliki varian tingkat cahaya yang berbeda.

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data penelitian yang digunakan adalah tipe observasi, dengan mengobservasi masing masing objek pada saat melakukan gerakan. Penelitian dilakukan dengan beberapa sesi yang dibagi atas jarak dan tingkat pencahayaan. Satu persatu subjek berada pada jarak posisi yang ditentukan dan mulai melakukan pergerakan sesuai klasifikasi yaitu kanan, kiri, tegak dan menunduk. Setelah satu subjek selesai diganti dengan objek selanjutnya hingga selesai. Setelah semua subjek selesai pada sesi pertama, jarak diganti dan subjek kembali diuji. Setelah sesi jarak selesai kemudian tingkat pencahayaan yang diganti hingga semua sesi selesai dan semua subjek telah diuji. Data tersebut yang kemudian digunakan sebagai analisis hasil.

3.2.4 Metode Analisis Data

Metode yang digunakan untuk menganalisis hasil secara kuantitatif adalah dengan meneliti hasil uji yang telah diperoleh untuk kemudian dijelaskan tentang akurasi dari setiap masing-masing pergerakan baik berdasarkan jarak subjek maupun tingkat pencahayaan. Sehingga diperoleh statistik yang dapat digunakan sebagai parameter akurasi dari alat yang dibuat. Untuk pengujian kualitatif dibuktikan dengan penggunaan deteksi tepi dengan metode deteksi warna YcbCr dan HSV ini dibandingkan dengan teori yang telah diperoleh pada

jurnal. Kemudian akan diperoleh hasilnya berupa perbandingan akurasi sistem yang telah didapat dengan teori.

3.2.5 Peralatan Pendukung Penelitian

Terdapat beberapa peralatan pendukung yang digunakan pada penelitian.

Pada sub bab ini dijelaskan tentang daftar alat yang digunakan guna mendukung proses penelitian antara lain :

Kebutuhan Software :

1. Raspbian
2. Visual Studio
3. OpenCV
4. Python
5. Thorny

Kebutuhan Hardware :

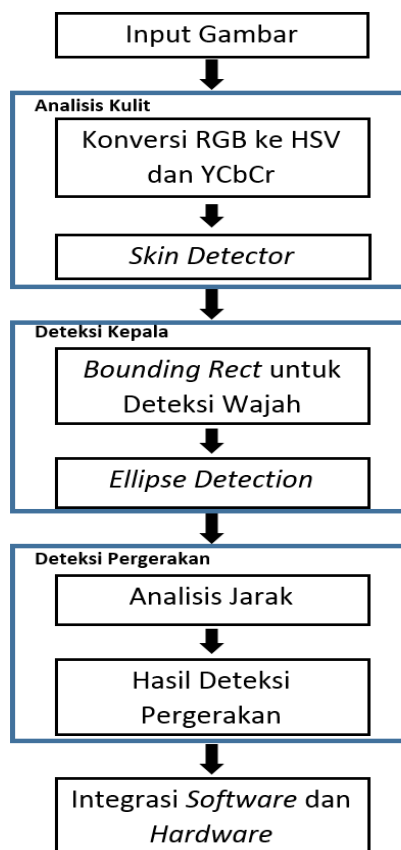
1. Layar LCD 14 inch dilengkapi dengan mainboard SKR.03 8501 dan universal inverter 12V sebagai penyeimbang daya yang akan masuk pada LCD
2. Raspberry Pi B+
3. Sensor HC-SR04
4. Adapter 12V 2A sebagai sumber listrik untuk LCD
5. Adapter 5V 2A sebagai sumber listrik untuk Raspberry Pi
6. Modul kamera webcam mini 720p 5MP Raspberry Pi
7. Kabel jumper
8. Kabel HDMI 1,5 meter

BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum Sistem

Pada tahap deskripsi umum dijelaskan tentang penggambaran seluruh sistem pada saat program dijalankan. Dengan tujuan akhir dari sistem adalah mampu mengklasifikasikan gerakan kepala dan membuat keputusan berdasarkan klasifikasi untuk menggerakkan menu pada LCD. Dengan perangkat yang digunakan yakni kamera sebagai penangkap gambar subjek dan sensor jarak untuk parameter jarak subjek dengan tujuan agar jarak antar subjek adalah sama. Dengan penempatan subjek di depan layar LCD dan dengan jarak yang telah ditentukan, subjek bergerak sesuai klasifikasi gerakan yakni kanan, kiri, tegak dan menunduk dan kamera akan memproses gambar tersebut sebagai input dari sistem yang dibuat.

Bagian ini akan menampilkan tentang proses dasar atau gambaran umum dari mulai input, proses yang berjalan dan output dari sistem secara sederhana dan menjelaskan proses berjalan program secara garis besar.



Gambar 4.1 Gambaran Umum pada Sistem

Hasil yang ditangkap oleh kamera selanjutnya diproses pertama filter warna *HSV* dan *YcbCr*. Setelah gambar dari filter tersebut terproses maka selanjutnya adalah mengatur erosi yakni guna mengurangi *noise* pada gambar hasil filter dan dilasi untuk mempertajam bentuk objek sehingga hasil yang diperoleh untuk proses selanjutnya menjadi lebih baik dan *noise* berkurang. Selanjutnya setelah hasil dari dilasi dan erosi diperoleh dibuat bentuk *ellipse* dari objek untuk merepresentasikan bentuk kepala subjek selain itu guna memudahkan pada proses pengklasifikasian objek. Setelah diperoleh gambar *ellipse* subjek maka selanjutnya sistem menganalisa garis tepi pada *ellipse*, garis ini kemudian dipergunakan sebagai parameter instruksi sistem navigasi pada menu yang telah dibuat dengan aplikasi Visual Studio merupakan menu sederhana dengan 4 pilihan menu yaitu menu kanan, menu kiri, menu atas dan menu bawah. Dengan menentukan titik masing masing koordinat pada keempat pilihan menu tersebut sistem pergerakan kepala ini berjalan.

4.2 Analisis Kebutuhan

Analisis kebutuhan pada sistem memiliki tujuan sebagai analisa seluruh kebutuhan yang diperlukan oleh sistem yang hendak dibangun. Analisis kebutuhan berdasarkan identifikasi kebutuhan dapat dibagi menjadi berikut :

4.2.1 Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan-kebutuhan sistem agar sistem dapat berjalan sesuai dengan yang diharapkan. Kebutuhan fungsional antara lain:

1. Sistem mendapatkan sebuah data yang telah didapatkan oleh *Input* atau masukan.

Input yang akan didapatkan oleh sistem akan melakukan pendataan yang berupa gambar objek dari kamera dan juga menampilkan jarak sensor.

2. Sistem dapat menerjemahkan data yang telah didapat dari *input* atau masukan menjadi sebuah instruksi.

Sistem yang telah mendapatkan data dari *input* selanjutnya diproses dalam filter *HSV* dan *YCbCr* dan di kurangi *noise* nya menggunakan erosi dan dilasi untuk menebalkan titik nilai dari deteksi filter, kemudian hasil tersebut dimasukkan kedalam bentuk oval melalui modul *cv2.ellipse* dan kemudian diambil dua garis antara centroid dengan dua titik pada pertengahan nilai *x* dan *y* untuk dihitung garis menggunakan rumus linier dan hasil perbandingan kedua garisnya digunakan sebagai navigasi menu kanan dan kiri . Untuk posisi tegak dan menunduk digunakan perbandingan nilai antara *h* (*high*) dan *w* (*width*) .

3. Sistem dapat melakukan sebuah eksekusi perintah pada *ouput* atau navigasi menu pada LCD.

Output pada sistem ini akan melakukan pergerakan. Pergerakan akan dilakukan setelah mendapat sebuah instruksi dari pemrosesan. Setelah data intruksi didapatkan akan dilakukan pergerakan yang sesuai dengan intruksi yang di dapat. Instruksi ini menggunakan titik koordinat sebagai penentu eksekusi dari program. Pada masing masing menu ditentukan terlebih dahulu titik koordinatnya.

4. Sistem dapat menganalisis warna kulit melalui konversi *RGB* ke *HSV* dan *RGB* ke *YCbCr*

Sistem mengkonversi warna yang diperoleh dari input kamera menjadi dua tipe warna yakni *HSV* dan *YCbCr* yang kemudian dua warna ini memiliki nilai minimum dan maksimum, sehingga mampu menjadi filter inputan warna sebagai area yang dibutuhkan nantinya sebagai *ROI* (*Region of Interest*).

5. Sistem dapat mendeteksi pergerakan kepala dan mengurangi *noise*.

Sistem memiliki filter terhadap *noise* sehingga objek yang memiliki warna menyerupai warna kulit dengan ukuran yang lebih kecil tidak ikut terdeteksi dan mempengaruhi nilai dari parameter yang telah ditentukan oleh sistem. Filter ini menggunakan ukuran area dari *BoundingRect* sehingga akan mengabaikan objek dengan nilai piksel lebih kecil daripada nilai yang ditentukan pada fungsi *centroid*.

6. Deteksi pergerakan pada sistem dan membandingkan nilai dari titik tepi wajah yang dapat diklasifikasikan kedalam macam gerakan.

Sistem mampu mendeteksi gerakan pada sistem dan membandingkan nilai dari titik tepi wajah yang dapat diklasifikasikan kedalam macam gerakan yakni tegak, menunduk, kiri dan kanan yang kemudian dicocokkan dengan titik koordinat dari menu navigasi pada LCD.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan berikut adalah suatu kebutuhan yang menitikberatkan properti perilaku yang dipunyai sistem. kebutuhan yang mempunyai batasan pada layanan juga fungsi yang telah ditawarkan oleh sistem pada penelitian ini merupakan desain, batasan waktu, batasan pada pengembangan proses, dan standarisasi.

1. Desain menu LCD yang sesuai pada perancangan
2. Alat dipasang di depan subjek menghadap pada subjek dengan posisi kamera tepat di depan layar monitor LCD
3. Pada sensor jarak menggunakan jarak dengan range perbedaan antar subjek 1-5 cm.
4. Background pada proses percobaan berwarna polos dan tidak termasuk dalam warna yang terdeteksi sebagai *noise* yakni warna coklat, merah dan warna lain yang mendekati warna kulit.

4.3 Analisis Kebutuhan Perangkat

Pada bagian ini akan menjelaskan analisis kebutuhan perangkat. Kebutuhan perangkat terdiri dari dua bagian yaitu perangkat lunak dan perangkat keras. Perangkat lunak dibutuhkan pada sistem ini adalah sebagai berikut :

4.3.1 Kebutuhan Perangkat Lunak

4.3.1.1 Raspbian

Raspbian adalah sistem operasi gratis berbasis Debian yang dioptimalkan untuk perangkat keras Raspberry Pi. Sistem operasi raspbian berisikan serangkaian program dasar dan utilitas yang dibutuhkan untuk menjalankan Raspberry Pi. Raspbian menyediakan 35.000 paket, Raspbian masih dalam pengembangan aktif dengan penekanan pada peningkatan stabilitas dan kinerja sebanyak mungkin paket Debian. Sehingga dengan digunakannya Raspbian pada Raspberry Pi dapat berkomunikasi dengan perangkat socket maupun pin pin yang ada pada Raspberry Pi guna menjalankan program sistem yang dibuat.

4.3.1.2 2. Microsoft Visual Studio

Untuk aplikasi yang digunakan sebagai objek penelitian adalah aplikasi berbasis visual basic yang disediakan oleh Microsoft Visual Studio. Microsoft Visual Studio merupakan sebuah *software suite* mampu digunakan sebagai pengembang aplikasi, pengembangan dapat dalam bidang aplikasi personal, aplikasi bisnis, atau komponen aplikasinya, dalam bentuk console, berupa aplikasi berbasis Windows, atau aplikasi berbasis web. Visual Studio mencakup *compiler*,



Gambar 4.2 Logo Microsoft Visual Studio Software

Sumber: (<https://visualstudio.microsoft.com/>)

IDE (*Integrated Development Environment*), SDK, dan dokumentasi yang umumnya MSDN Library. *Compiler* yang mendukung dalam aplikasi Visual Studio antara lain Visual C#, Visual Basic.NET, Visual Basic, Visual C++, Visual J#, Visual J++, Visual InterDev, Visual FoxPro, dan Visual SourceSafe.

Pada sistem ini digunakan bahasa C++ untuk membentuk aplikasi Windows dengan tampilan 4 label yang ketika menerima kursor maka warna label akan berubah dan ketika tidak menerima klik kursor maka label dalam aplikasi menjadi warna putih. Dengan bantuan *library OpenCV* yaitu pyautogui yang dapat melakukan autoklik pada koordinat yang ditentukan maka kedua fitur ini dapat menjadi output sistem dengan interface sederhana dan mampu menampilkan secara visual tentang parameter pengujian sistem.

4.3.1.3 OpenCV

Open Source Computer Vision Library (OpenCV) berlisensi BSD dan gratis jika digunakan untuk penggunaan akademis dan sebagai komersial. Library ini telah mendukung Bahasa dengan tingkat pemrograman C ++, Python serta antarmuka Java juga mendukung untuk sistem operasi Windows, Mac OS ,Linux, iOS dan juga Android.



Gambar 4.3 Logo OpenCV.

Sumber : (<https://opencv.org>)

OpenCV dirancang sebagai efisiensi komputasi memiliki fokus yang kuat pada aplikasi secara real-time. Ditulis dalam bahasa pemrograman C / C ++ yang telah dioptimalkan, *Library* mampu memanfaatkan proses multi-core. Diaktifkan dengan OpenCL, yang memanfaatkan akselerasi dari hardware platform komputasi heterogen yang mendasarinya (Pulli et al. 2012).

4.3.2 Kebutuhan Perangkat Keras

Kemudian kebutuhan perangkat keras yang dibutuhkan sistem ini yaitu :

4.3.2.1 Raspberry Pi B+

Raspberry Pi yang digunakan dalam penelitian merupakan tipe Raspberry Pi B+ . Dengan spesifikasi dapat dilihat pada tabel 4.1, Raspberry Pi B+ merupakan sebuah mini komputer yang merupakan perkembangan dari versi sebelumnya, yakni Raspberry Pi tipe B. Dengan model yang sama namun memiliki keunggulan performa yang lebih baik sehingga dipilih sebagai alat

penelitian. Memiliki Broadcom BCM2873B0 Cortex A53 64-bit dengan kecepatan prosesor 1,4 GHz, support pada Wireless 802.11ac juga Bluetooth 4.2, Chipset yang memiliki sistem pengaturan suhu yang lebih baik dan port Ethernet dengan support PoE (Power over Ethernet) merupakan macam fitur yang dimiliki versi B+ yang jika dibandingkan dengan versi sebelumnya tentu performanya dari Raspberry Pi B+ lebih unggul sehingga diharapkan dengan memakai single board komputer ini dapat meningkatkan efisiensi alat serta performanya yang baik saat sistem program dijalankan.



Gambar 4.4 Modul Raspberry Pi

Sumber : (<https://www.raspberrypi.org/>)

Raspberry Pi tipe B+ yang memiliki spesifikasi sebagai berikut :

Tabel 4.1 Spesifikasi Raspberry Pi B+

Prosesor :	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Jenis Memori :	1GB LPDDR2 SDRAM
Konektifitas :	2.4GHz dan 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports
Fitur Akses:	Memiliki 40-pin GPIO
Tampilan Video dan Audio:	1 × full size HDMI MIPI DSI display port MIPI CSI camera port



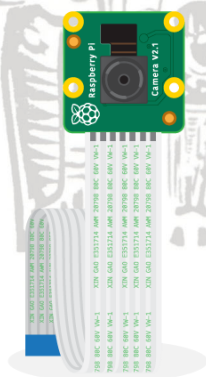
Tabel 4.1 (Lanjutan)

Multimedia:	H.264, MPEG-4 <i>decode</i> (1080p30); H.264 <i>encode</i> (1080p30); OpenGL ES 1.1, 2.0 <i>graphics</i>
Media Penyimpanan :	Kartu Micro SD sebagai penyimpan sistem operasi
Sumber Daya:	5V/2.5A DC <i>micro USB connector</i> 5V DC <i>GPIO header</i> <i>Power over Ethernet</i> (PoE)– enabled
Ketahanan:	Temperatur pada kisaran 0–50°C

Sumber : (<https://www.raspberrypi.org/>)

4.3.2.2 Kamera Raspberry Pi

Kamera Raspberry yang digunakan dalam penelitian merupakan tipe REV 1.3 yang memiliki resolusi kamera 5mp. Kamera merupakan komponen penting dalam penelitian sebagai alat input gambar video yang selanjutnya diproses oleh sistem.



Gambar 4.5 Modul Kamera Raspberry Pi

Sumber : (<https://projects.raspberrypi.org/>)

Modul kamera Raspberry Pi ini memiliki beberapa fitur yang dimiliki.

Antara lain :

1. Sepenuhnya Kompatibel dengan Raspberry Pi Model A dan Model B.
2. Modul kamera 5MP Omnivision 5647 .
3. Resolusi Gambar: 2592 x 1944.

4. Rekaman Video: Supports 1080p @ 30fps, 720p @ 60fps dan 640x480p 60/90.

5. 15-pin MIPI *Camera Serial Interface* – Memiliki soket yang khusus untuk terhubung pada modul *Raspberry*.

4.3.2.3 Layar LCD

Layar LCD pada penelitian ini berguna sebagai media tampilan output sebagai navigasi menu dan juga menampilkan hasil input dari kamera setelah diproses oleh filter. Layar LCD yang digunakan dalam penelitian merupakan tipe LCD dengan layar IPS memiliki ukuran 14 inch dan beresolusi HD 1366 x 768.



Gambar 4.6 Panel LCD 14 inch

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini dijelaskan tentang proses perancangan juga implementasi dari sistem yang akan dijabarkan mengenai deteksi pergerakan kepala berdasarkan cross area konversi warna HSV dan YCbCr Berbasis Raspberry-Pi. Penjabaran tentang perancangan dan implementasi dijelaskan secara runtut dari mulai gambaran umum dari sistem hingga sub proses pada saat output sistem berjalan.

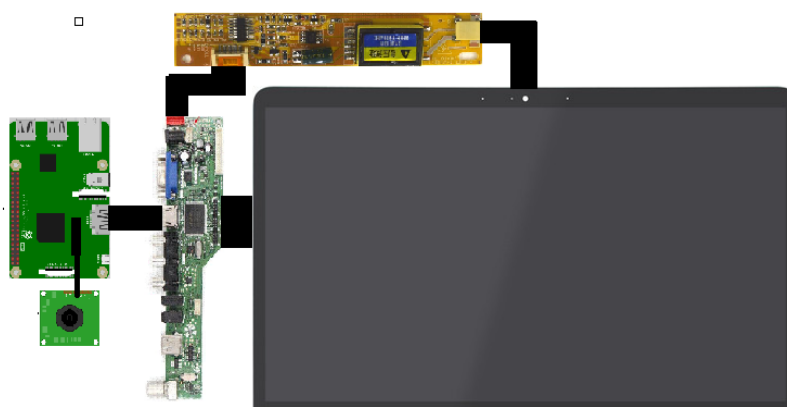
5.1 Perancangan Sistem

5.1.1 Perancangan Perangkat Keras

Sub bab ini akan membahas tentang perancangan *hardware* dari sistem mencakup pemasangan lcd pada Raspberry-Pi, inialisasi sensor jarak pada Raspberry-pi dan kamera secara mendetil untuk menjelaskan tentang *hardware* keseluruhan yang digunakan pada program. Untuk kemudian diimplementasikan ke dalam program sesuai tujuan dan dikoordinasikan melalui software dan kode program.

5.1.1.1 Instalasi LCD 14" pada Raspberry-Pi

Pada bab ini dilakukan instalasi LCD agar dapat terkoneksi pada perangkat Raspberry Pi. LCD memerlukan 2 komponen utama agar dapat berfungsi sebagai layar output, yakni *inverter* dan *controller board*. Inverter berfungsi sebagai penyeimbang daya yang masuk pada LCD dan pengubah daya dari 220v ke 9v dan controller board merupakan pusat kendali layar LCD sebagai output. Controller Board yang digunakan adalah Universal Controller Board dengan tipe mesin T.V56.031. Setelah socket LDVS interface dan Inverter dihubungkan pada board. Selanjutnya port HDMI pada Raspberry Pi dihubungkan pada port HDMI pada controller board dan instalasi layar LCD sebagai output dapat digunakan.



Gambar 5.1 Gambar Instalasi LCD dan kamera pada Raspberry-Pi

Tabel 5.1 Tabel Pin Perancangan Hardware

Raspberry Pi	LCD 14"	Controller Board	Inverter	Kamera Raspberry Pi
CSI Camera Port Out	-	-	-	Camera Port In
HDMI In	-	HDMI Out	-	-
-	LDVS Out	LDVS In	-	-
-	-	IR & Key In	-	-
-	V in	Inverter In	Inverter Out	-

Pada tabel 5.1 tersebut dijelaskan tentang pin yang digunakan pada penelitian ini, Pin Kamera raspberry dihubungkan pada pin kamera output pada board, port kabel HDMI input pada modul raspberry sebagai input dihubungkan pada controller board sebagai output. Kemudian LDVS untuk port tampilan pada LCD dari board sebagai input, IR & Key interface untuk menyalakan remote pada LCD. Dan untuk mengatur arus listrik yang masuk pada LCD digunakan inverter.

5.1.2 Perancangan Perangkat Lunak

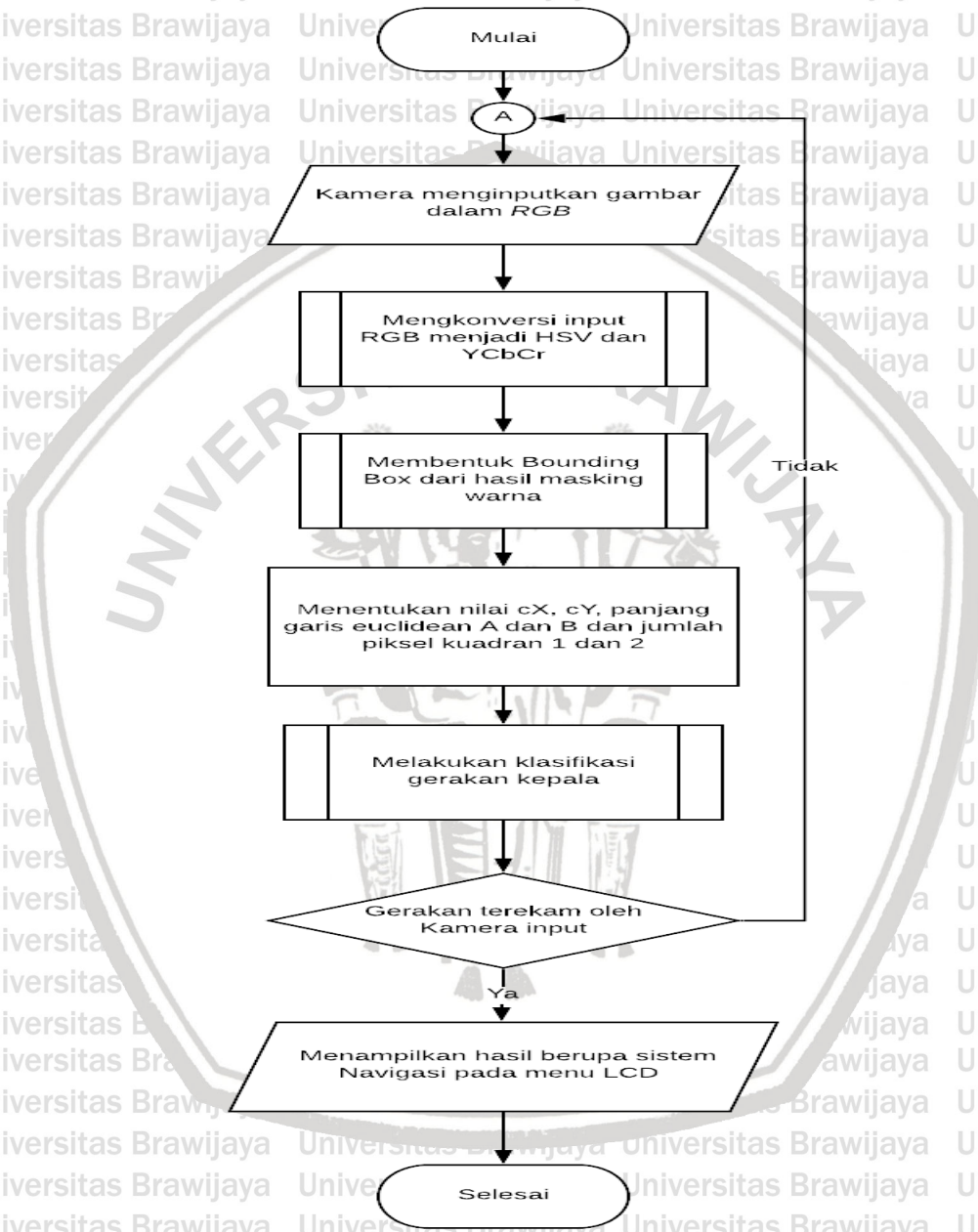
Sub bab ini menjelaskan tentang perancangan dari perangkat lunak yang digunakan pada penelitian. Dimulai dari perancangan proses utama pada sistem, proses konversi warna, pembentukan bounding box, hingga klasifikasi pergerakan.

5.1.2.1 Rancangan Proses Utama pada Sistem

Pada proses utama perancangan perangkat ini ditampilkan dalam flowchart yang menunjukkan proses berjalannya perangkat lunak mulai dari aliran data dan juga proses berjalannya sistem secara umum. Diawali dengan berjalannya proses melalui inputan data dari kamera dalam bentuk citra RGB yang selanjutnya dikonversikan input tersebut kedalam format yang lain yakni format HSV dan YCbCr. Selanjutnya hasil dari konversi tersebut diolah ke Bounding Box untuk menentukan area deteksi secara spesifik, untuk berikutnya sistem menentukan masing-masing koordinat dari *centroid* dan 2 titik lain dari x dan y yang bertujuan untuk menghitung jumlah jarak garis *euclidean* dari 4 koordinat tersebut juga jumlah piksel yang dimiliki oleh kuadran 1 dan 2. Setelah



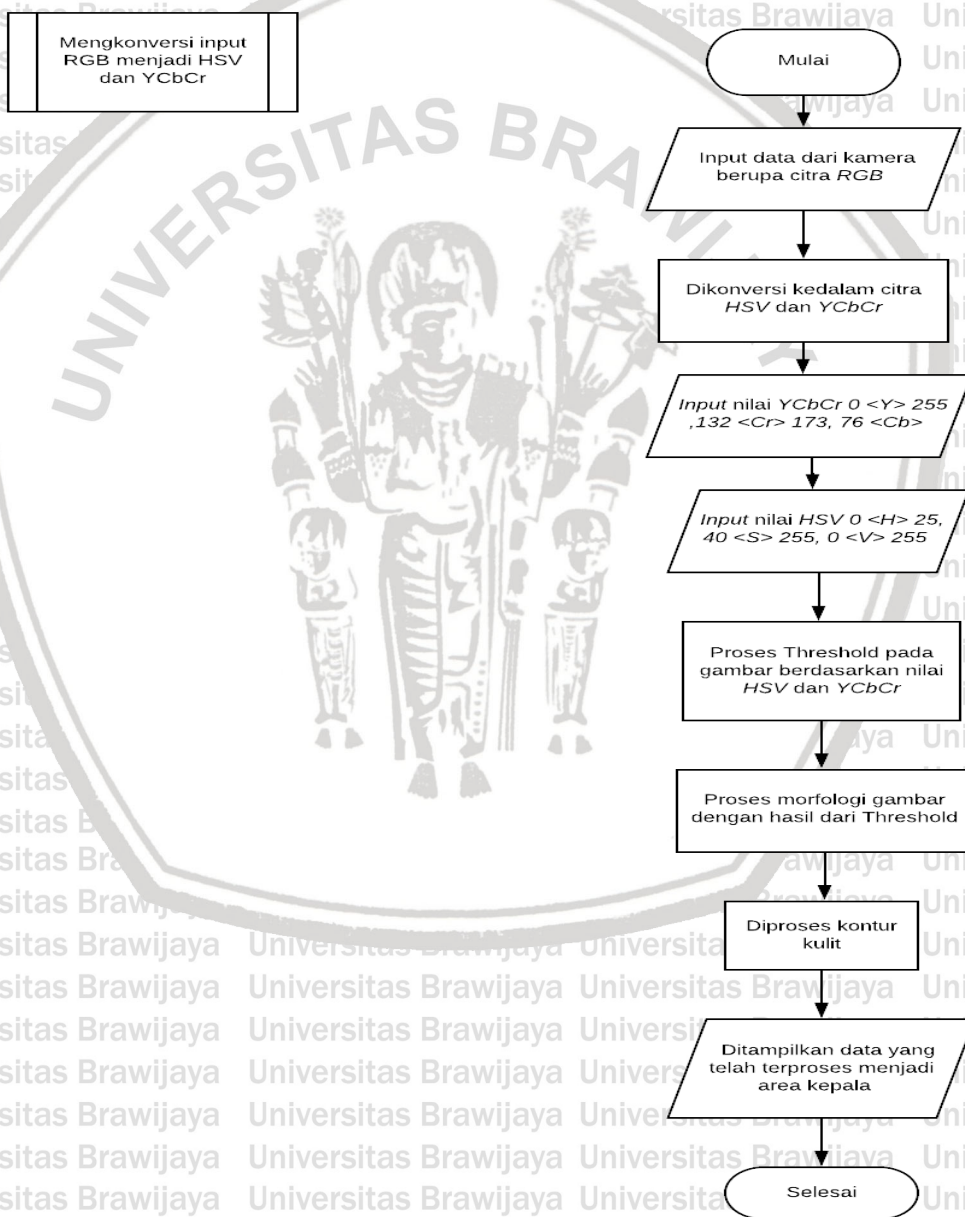
nilai terkumpul maka dilakukan klasifikasi gerakan kepala menggunakan perbandingan nilai h/w garis *euclidean* A dan B dan jumlah piksel kuadran 1 dan 2. Jika input masih terdeteksi oleh sistem maka proses akan kembali pada tahap proses pembacaan citra. Hasil keluaran output ini berupa navigasi menu pada LCD. Seluruh alur sistem tersebut ditampilkan pada gambar 5.3.



Gambar 5.2 Diagram alur Proses Utama Sistem

5.1.2.2 Proses Konversi Input RGB menjadi HSV dan YCbCr

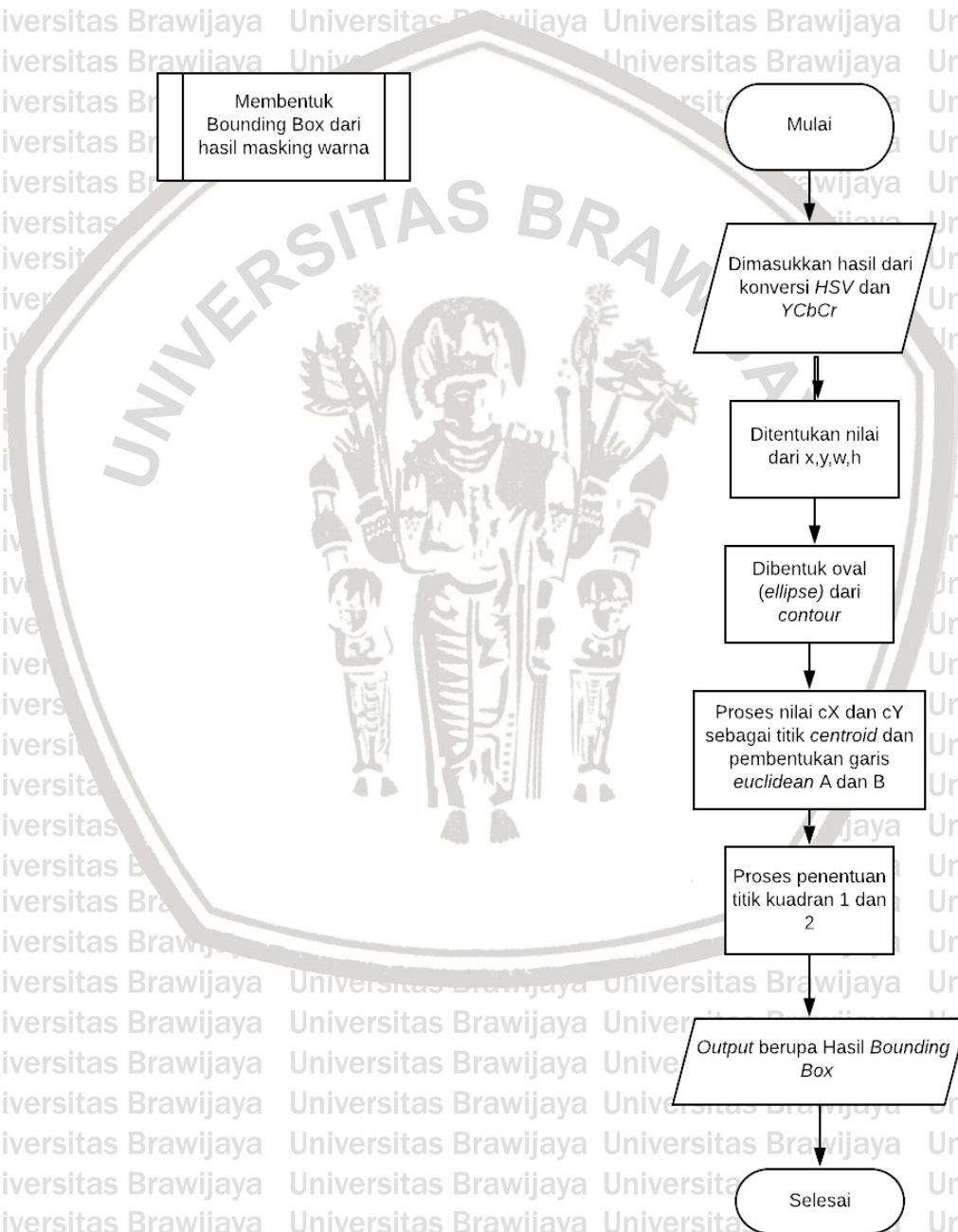
Gambar 5.4 adalah diagram yang menunjukkan proses konversi input dari RGB kedalam HSV dan YCbCr. Proses ini diperuntukkan sebagai fitur deteksi objek kepala yang merupakan fitur penting dalam proses. Dimulai dari input data melalui kamera yang kemudian dikonversi dari RGB kedalam HSV dan YCbCr setelah dikonversi dengan tingkat threshold yang ditetapkan kemudian terbentuk area deteksi yang merupakan bentuk kepala dalam bentuk biner. Setelah diproses morfologi maka dicari kontur untuk selanjutnya ditampilkan sebagai objek deteksi.



Gambar 5.3 Diagram alur Proses Konversi Warna

5.1.2.3 Proses Membentuk Bounding Box

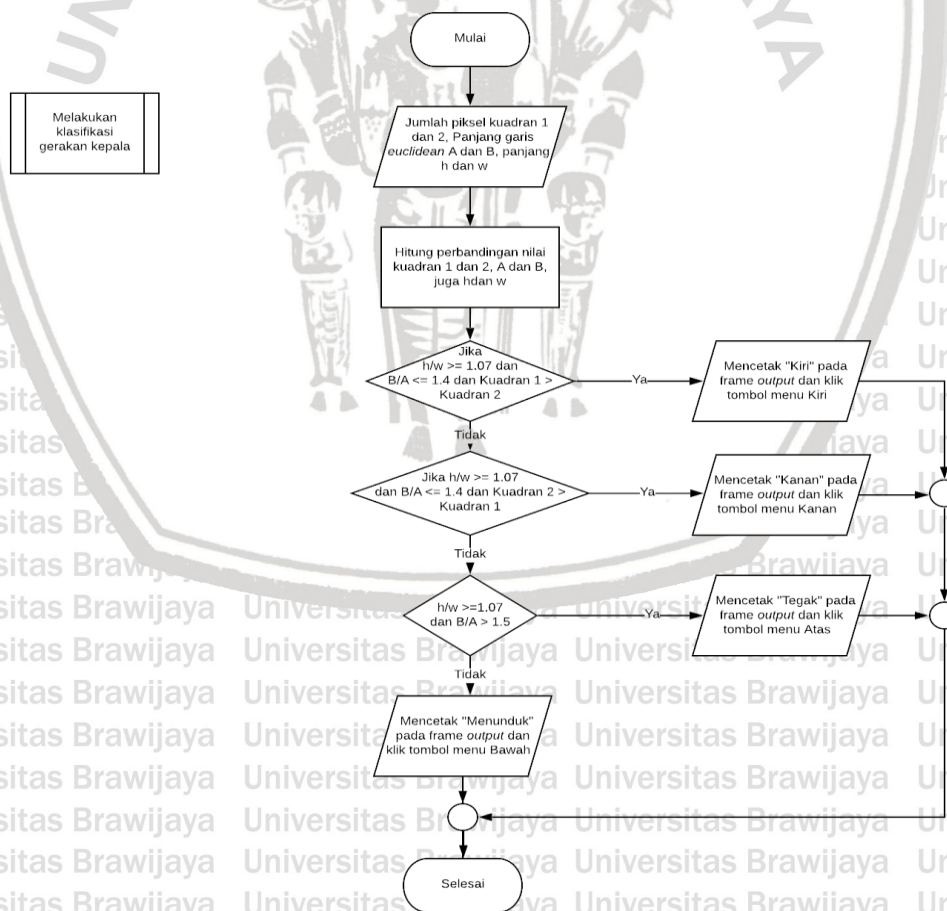
Pada gambar 5.5 berikut menunjukkan alur diagram alir pada pembentukan *bounding box*. Data hasil konversi *HSV* dan *YCbCr* selanjutnya diproses guna menentukan nilai dari 4 fitur yakni x, y, w dan h yang berfungsi x sebagai baris, y sebagai kolom, w untuk lebar dan h untuk nilai tinggi. Sehingga diperoleh nilai minimum dan maksimumnya, kemudian melalui parameter dari *contours* dibentuk ellipse. Setelah bentuk oval terbentuk maka ditentukan nilai centroid dan 2 garis *euclidean* dan wilayah kuadran 1 dan 2 untuk dibandingkan.



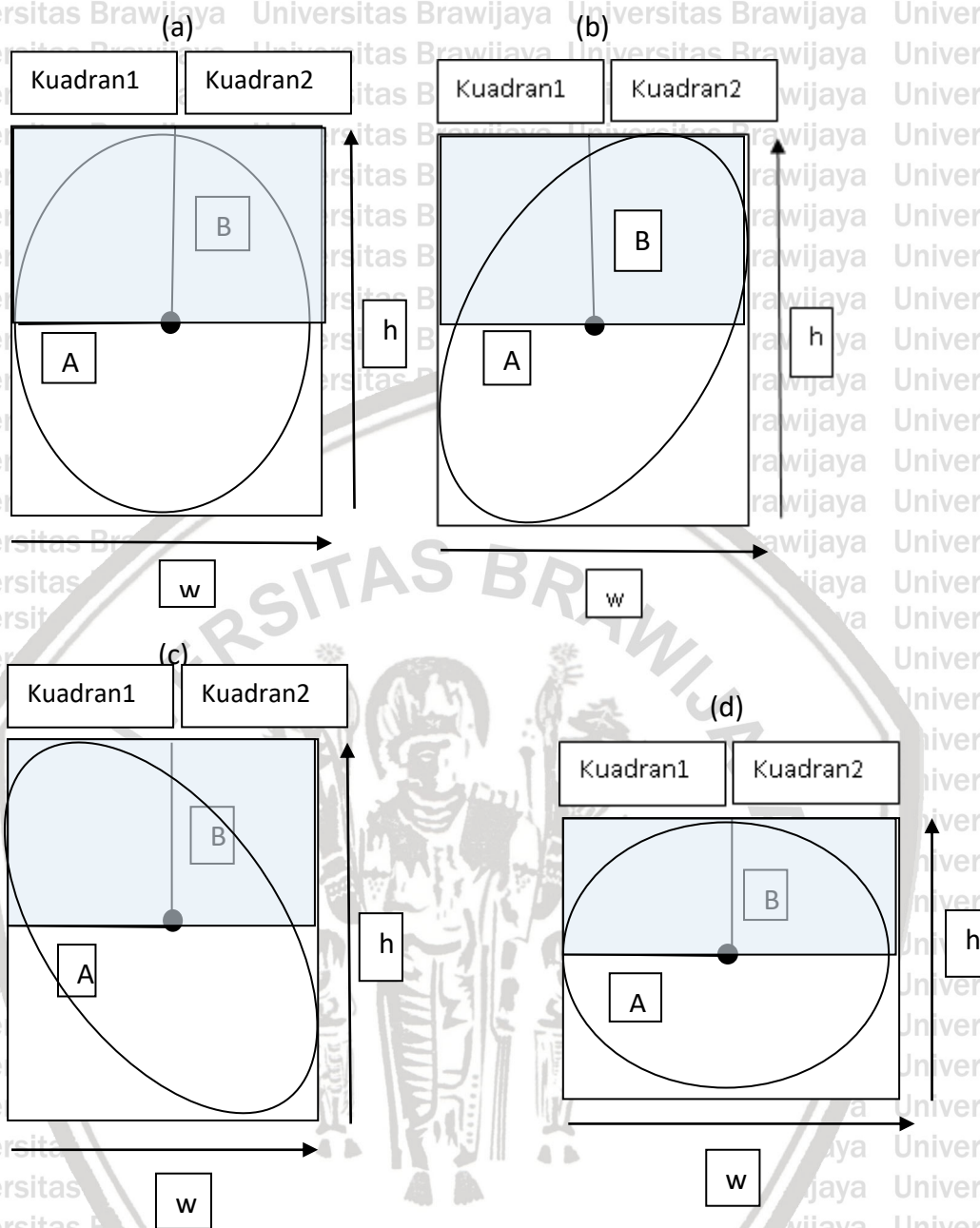
Gambar 5.4 Diagram alur Bounding Box

5.1.2.4 Proses Klasifikasi Pergerakan Kepala

Pada gambar 5.6 menjelaskan tentang diagram alir untuk klasifikasi pada pergerakan kepala. Inputan pada tahap ini merupakan 3 aspek yaitu jumlah piksel putih yang berada pada kuadran 1 dan 2, panjang garis euclidean dari garis A dan B juga nilai panjang dari tinggi (h) dan lebar (w). Ketiga elemen tersebut dibuat perbandingan dengan kondisi yang pertama yaitu ketika h/w bernilai lebih dari samadengan 1,07 maka selanjutnya dicek nilai piksel dari kuadran 1 lebih besar dari kuadran 2 dan perbedaan panjang garis *euclidean* dari B/A adalah kurang dari sama dengan 1,4 maka cetak string "Kiri" pada layar output dan *pyautogui* klik pada koordinat menu kiri, yang kedua yaitu ketika h/w bernilai lebih dari samadengan 1,07 maka dicek nilai piksel dari kuadran 2 lebih besar dari kuadran 1 dan perbedaan panjang garis *euclidean* dari B/A adalah kurang dari sama dengan 1,4 maka cetak string "Kanan" pada layar output dan *pyautogui* klik pada koordinat menu kanan, yang ketiga yaitu ketika h/w bernilai lebih dari samadengan 1,07 dan perbedaan panjang garis *euclidean* dari B/A adalah lebih dari 1,5 maka cetak string "Tegak" pada layar output dan *pyautogui* klik pada koordinat menu atas, dan ketika h/w bernilai kurang dari 1,07 maka cetak string "Menunduk" pada layar output dan *pyautogui* klik pada koordinat menu bawah.



Gambar 5.5 Flowchart Klasifikasi Pergerakan Kepala



Gambar 5.6 Tampilan *Bounding Box*

Nilai perbandingan h dan w juga nilai perbandingan B dan A didapatkan dari nilai threshold yang diperoleh dari data latih tabel 5.2 dan 5.3 dibawah ini. Dari nilai threshold yang telah ditunjukkan threshold paling baik untuk perbandingan nilai h dan w adalah menggunakan nilai antara 1,07 dan nilai untuk kanan dan kiri dengan menggunakan perbandingan nilai B dan A adalah antara 1,3 dan 1,4 dengan persentase sama yakni 60%. Dengan menggunakan nilai ini sebagai nilai threshold maka sistem dapat menentukan definisi pergerakan kepala yang digunakan pada penelitian ini. Data latih dilakukan menggunakan 5 subjek yang berbeda dengan tingkat pencahayaan dan jarak yang sama.

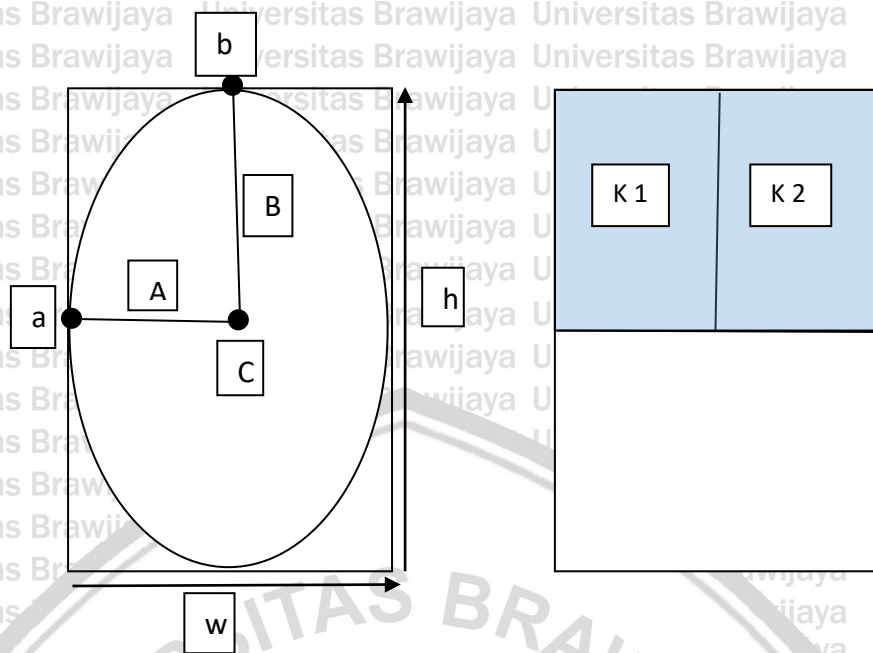
Tabel 5.2 Hasil Pengujian pada nilai h dan w

Pergerakan	Nilai h/w	Jumlah Percobaan	Benar	Salah	Rata-rata Akurasi
Tegak	1,12	5	3	2	60%
	1,07		4	1	80%
	0,9		5	0	100%
Menunduk	1,12	5	4	1	80%
	1,07		4	1	80%
	0,9		2	3	40%

Tabel 5.3 Hasil Pengujian pada nilai B dan A

Pergerakan	Nilai B/A	Jumlah Percobaan	Benar	Salah	Rata-rata Akurasi
Kanan	1,3	5	3	2	60%
	1,4		3	2	60%
	1,5		2	3	40%
Kiri	1,3	5	3	2	60%
	1,4		3	2	60%
	1,5		2	3	40%

Pada gambar 5.7 merupakan hasil output *Bounding Box* terdapat 4 opsi yaitu untuk (a) adalah gambar output ketika hasil terdeteksi sebagai gerakan tegak, (b) adalah hasil dari output gerakan kanan, (c) adalah ketika output gerakan terdeteksi sebagai gerakan kekiri dan untuk gambar (d) adalah gerakan menunduk. Deteksi berdasarkan nilai h/w terlebih dahulu, jika syarat h/w memenuhi yakni hasil perbandingannya 1,07 maka selanjutnya digunakan parameter kedua yaitu nilai A/B ketika nilai dari A/B adalah kurang dari atau samadengan 1,4 maka perbandingan jumlah piksel pada kuadran 1 dan 2. Jika jumlah piksel pada kuadran 1 yang lebih tinggi maka klasifikasi menunjukkan kiri, jika sebaliknya maka klasifikasi pergerakan kanan. Jika nilai dari A/B ternyata lebih dari 1,4 maka pergerakan didefinisikan sebagai tegak dan jika h/w kondisi pertamanya adalah tidak memenuhi atau kurang dari 1,07 maka hasilnya adalah pergerakan menunduk.



Gambar 5.7 Pola Bounding Box.

Pada gambar 5.8 terdapat visualisasi *bounding box* yang digunakan dalam penelitian dengan berbagai parameter yang digunakan sebagai klasifikasi dari pergerakan kepala. A merupakan hasil dari panjang garis *euclidean* yang menghubungkan titik C dan a dan B merupakan panjang garis *euclidean* yang menghubungkan titik C dan b, K merupakan simbol untuk daerah kuadran. Tabel 5.4 dibawah ini merupakan kondisi syarat penggolongan pergerakan kepala.

Tabel 5.4 Syarat Kondisi Penggolongan Pergerakan Kepala

Kondisi	$h/w \geq 1,07$	$B/A \leq 1,4$	$K 1 > K2$	$K2 > K1$
Tegak	Ya	Tidak	Ya atau Tidak	Ya atau Tidak
Kanan	Ya	Ya	Tidak	Ya
Kiri	Ya	Ya	Ya	Tidak
Menunduk	Tidak	Ya	Ya atau Tidak	Ya atau Tidak

5.2 Implementasi Sistem

Implementasi sistem merupakan bab yang menjelaskan tentang perwujudan dari rancangan sistem. Pembuatan alat diilustrasikan kedalam prototype dan pengimplementasian perangkat lunak juga perangkat keras.

5.2.1 Implementasi Perangkat Keras

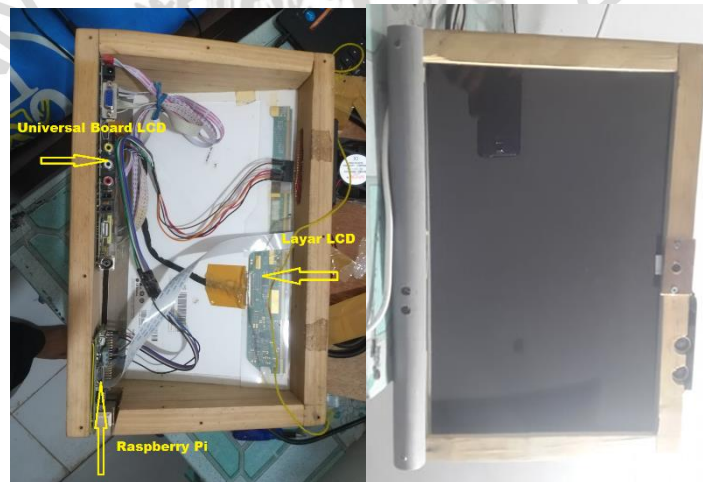
Pada sub bab ini ditunjukkan oleh gambar 5.8 tentang posisi objek dan gambar prototype alat dari samping. Bagian *case* terbuat dari kayu yang ringan



agar dapat mempertahankan posisi kabel serta menjaga LCD dan kamera tetap pada posisi yang diinginkan. Raspberry Pi juga ditanamkan dalam alat agar memudahkan instalasi pemasangan pada saat sistem akan digunakan. Terdapat pelindung pada kamera dan sensor jarak agar aman dari gesekan benda lain di sekitar prototype. Sedangkan tampilan bagian belakang dan depan ditunjukkan oleh gambar 5.9.



Gambar 5.8 Tampilan Prototype Alat Nampak Samping



Gambar 5.9 Tampilan Prototype Alat

5.2.2 Implementasi Perangkat Lunak

Pada sub bab ini dijelaskan tentang implementasi perangkat lunak yang merupakan proses utama dari program berupa *source code* yang dijabarkan tahapan dari tiap baris kode program. Proses pemrograman menggunakan *Python* versi 2.7 dengan *OpenCV* 4.1.1 sebagai *library* penting dalam program sebagai *library image processing* selain *OpenCV* terdapat beberapa *library* lain yang digunakan. Pada tabel 5.2 berikut kode inialisasi variabel juga *library* yang digunakan.

Tabel 5.5 Kode Program Inisialisasi *library* dan *class*

Baris	Kode Program
1	<code>import cv2</code>
2	<code>import numpy as np</code>
3	<code>import time</code>
4	<code>import collections</code>
5	<code>import imutils</code>
6	<code>import pyautogui</code>
7	<code>from PIL import Image</code>
8	<code>from scipy.spatial import distance as dist</code>

Berikut penjelasan dari tabel 5.5 :

Baris 1 : Inisialisasi *cv2 library* sebagai *image processing*.

Baris 2 : Inisialisasi *numpy library* sebagai pengolah matematis.

Baris 3 : Inisialisasi *library time* sebagai pengatur waktu delay dan waktu komputasi.

Baris 4 : Inisialisai *library collections* sebagai penghitung kumulah piksel pada gambar.

Baris 5 : Inisialisasi *library imutils* sebagai utilitas untuk mengatur ukuran frame video.

Baris 6 : Inisialisasi *pyautogui library* sebagai fitur untuk klik navigasi pada menu.

Baris 7 : Impor *class* bernama *Image* dalam *library PIL* untuk utilitas gambar.

Baris 8 : Impor *class distance* dan diberi alias *dist* dari *library PIL* yang berfungsi sebagai penghitung jarak matriks.

Setelah dilakukan inisialisasi *library* langkah selanjutnya adalah proses *capture* gambar yang digunakan sebagai input program berupa video. Dalam tabel 5.6 dijelaskan proses input video hingga mengakhiri proses pengambilan input.

Tabel 5.6 Kode Program Input Video RGB

Baris	Kode Program
1	<code>cap= cv2.VideoCapture(0)</code>
2	<code>while cap.isOpened() :</code>
3	<code> frame= cap.read()</code>
4	<code> frame = imutils.resize(frame, width=480)</code>
5	<code> flip = cv2.flip(frame, 1)</code>
6	<code> main_frame = cv2.GaussianBlur(flip, (5,5), 0)</code>
7	<code> cv2.imshow ('mask',main_frame)</code>
8	<code> cv2.imshow ('frame oval', frame_oval)</code>
9	<code> if cv2.waitKey(1) & 0xFF == ord('q'):</code>
10	<code> break</code>
11	<code>cv2.destroyAllWindows()</code>
12	<code>cap.release()</code>



5.2.2.1 Implementasi Konversi Input RGB menjadi HSV dan YCbCr

Berikutnya pada tabel 5.7 adalah tabel berisi kode program yang berfungsi untuk mengkonversi hasil inputan yang berupa RGB menjadi diubah kedalam HSV dan YCbCr. Pada kode program baris 1 gambar input RGB diubah menjadi YCbCr, baris 2-4 menentukan threshold pada nilai minimum dan maksimum YCbCr, pada baris 5 input RGB diubah menjadi HSV dan pada baris 6-8 menentukan threshold pada nilai minimum dan maksimum HSV, selanjutnya pada baris 9 hasil kedua konversi tersebut digabungkan menjadi satu. Pada baris 10-13 merupakan proses penguatan hasil *masking* yakni melalui proses dilasi dan erosi pada gambar. Baris 14-18 merupakan fungsi dari filter morfologi, dan baris terakhir 19-22 merupakan kode program dari pembuatan area *contour*.

Tabel 5.7 Kode Program Konversi Warna HSV dan YCbCr

Baris	Kode Program
1	<code>ycbcr = cv2.cvtColor(main_frame, cv2.COLOR_BGR2YCrCb)</code>
2	<code>min_ycbcr = np.array ([0,132,76])</code>
3	<code>max_ycbcr = np.array ([255,173,127])</code>
4	<code>ycbcrmask = cv2.inRange(ycbcr, min_ycbcr, max_ycbcr)</code>
5	<code>hsv = cv2.cvtColor(main_frame, cv2.COLOR_BGR2HSV)</code>
6	<code>min_hsv = np.array ([0,40,0])</code>
7	<code>max_hsv = np.array ([25,255,255])</code>
8	<code>hsvmask = cv2.inRange(hsv, min_hsv, max_hsv)</code>
9	<code>mask = cv2.add(ycbcrmask,hsvmask)</code>
10	<code>kernel = np.ones((3,3), np.uint8)</code>
11	<code>thresh_erosi = cv2.erode(mask, kernel, iterations=5)</code>
12	<code>dilasi = cv2.dilate(thresh_erosi, kernel, iterations=6)</code>
13	<code>dilasi_cp = dilasi.copy()</code>
14	<code>hh, ww = dilasi.shape[:2]</code>
15	<code>kern = np.zeros((hh+2, ww+2), np.uint8)</code>
16	<code>cv2.floodFill(dilasi_cp, kern, (0,0), 255)</code>
17	<code>dilasi_cp = cv2.bitwise_not(dilasi_cp)</code>
18	<code>res_dilasi = dilasi dilasi_cp</code>
19	<code>_, contours, _ = cv2.findContours(res_dilasi, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)</code>
20	<code>bitwise = cv2.bitwise_and(main_frame, main_frame, mask=res_dilasi)</code>
21	<code>frame oval = np.zeros((480,480,1), np.uint8)</code>
22	

5.2.2.2 Implementasi Membentuk Bounding Box

Kode program selanjutnya adalah kode program dari implementasi pembentukan *bounding box*. Pada baris 1-3 adalah proses pembentukan area deteksi yang berarti dibatasi minimum 500 titik piksel. Kemudian baris 4-24 merupakan penentuan titik titik koordinat dari x,w,y,h sebagai parameter dari

bounding box. Pada baris 25-29 kode program untuk menggambar *ellipse* pada frame berdasarkan nilai dari kontur. Pada baris 30-35 menentukan koordinat centroid melalui titik koordinat *cX* dan *cY* dan dicetak kedalam frame . Pada baris ke 36 hingga 67 merupakan penentuan dari titik garis A yang terbentuk dari centroid ke *x* dan garis B yang terbentuk dari centroid ke *y* yang kemudian juga dicetak dalam bentuk garis dan titik kedalam frame. Pada baris 68-71 merupakan kode program untuk posisi wilayah dan jumlah piksel yang terdapat pada kuadran 1 dan 2, *areaAB* merupakan kuadran 1 dan *areaCD* merupakan kuadran 2. Pada baris 72 dan 73 merupakan kode rumus perbandingan dari point A ke centroid dan point B ke centroid. Pada baris 74-77 merupakan threshold dari perbandingan nilai *h/w*, baris 78 dan 79 merupakan rumus pembandingan dari *h/w* yang disimpan pada variabel "pkepala" dan panjang garis B dibandingkan dengan panjang garis A yang nilai hasilnya disimpan pada variabel *bpera* serta baris 80 memanggil modul navigasi. Pada baris 81-85 merupakan kode program untuk menampilkan nilai dari variabel "bpera", "pkepala" dan waktu komputasi kedalam frame. Tampilan kode program tersebut berada di tabel 5.8 dibawah ini.

Tabel 5.8 Kode Program Pembentukan Bounding Box

Baris	Kode Program
1	for cnt in contours :
2	area = cv2.contourArea(cnt)
3	if area > 500 :
4	xmin = 360
5	ymin = 360
6	xmax = 0
7	ymax = 0
8	for c in cnt :
9	xx = c[0][0]
10	yy = c[0][1]
11	if (xx < xmin):
12	xmin = xx
13	if (yy < ymin):
14	ymin = yy
15	if (xx > xmax):
16	xmax = xx
17	if (yy > ymax):
18	ymax = yy
19	x = xmin
20	y = ymin
21	w = xmax - xmin
22	h = ymax - ymin
23	x1 = x + w
24	y1 = y + h
25	ellipse = cv2.fitEllipse(cnt)
26	cv2.ellipse(frame_oval, ellipse, [255], -1)
27	cv2.ellipse(main_frame, ellipse, (0, 255, 255), 2)
28	cv2.rectangle(main_frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

Tabel 5.8 (Lanjutan)

```

29 cv2.rectangle(frame_oval, (x,y), (x+w, y+h),
[255], 1)
30 cX = int((x+w)/2)
31 cY = int((y+(y+h))/2)
32 pointCenter = (cX,cY)
33 cv2.circle(main_frame, (cX, cY), 3, (255, 255,
255), -1)
34 cv2.putText(main_frame, 'Center', (cX+5,cY+5),
35 cv2.FONT_HERSHEY_PLAIN, 1, (200,255,155), 1,
cv2.LINE_AA)
36 point1 = (x,y)
37 point2 = (x+w,y)
38 point3 = (x1,y1)
39 point4 = (x,y1)
40 cv2.circle(main_frame, (point1[0], point1[1]), 3,
(255, 255, 255), -1)
41 cv2.circle(main_frame, (point2[0], point2[1]), 3,
(255, 255, 255), -1)
42 cv2.circle(main_frame, (point3[0], point3[1]), 3,
(255, 255, 255), -1)
43 cv2.circle(main_frame, (point4[0], point4[1]), 3,
(255, 255, 255), -1)
44 PointAx = int((point1[0] + point4[0])/2)
45 PointAy = int((point1[1] + point4[1])/2)
46 PointBx = int((point1[0] + point2[0])/2)
47 PointBy = int((point1[1] + point2[1])/2)
48 pointA = (PointAx,PointAy) #kolom, baris
49 pointB = (PointBx,PointBy)
50 pointATextx = int((pointA[0] + cX)/2)
51 pointATexty = int((pointA[1] + cY)/2)
52 pointAText = (pointATextx,pointATexty)
53 pointBTextx = int((pointB[0] + cX)/2)
54 pointBTexty = int((pointB[1] + cY)/2)
55 pointBText = (pointBTextx,pointBTexty)
56 cv2.circle(main_frame, (pointA[0], pointA[1]), 3,
(255, 255, 255), -1)
57 cv2.circle(main_frame, (pointB[0], pointB[1]), 3,
(255, 255, 255), -1)
58 cv2.line(main_frame, (pointA[0],pointA[1]),
(cX,cY), (0,255,0), 2)
59 cv2.line(main_frame, (pointB[0],pointB[1]),
(cX,cY), (0,255,0), 2)
60 cv2.line(frame_oval, (pointA[0],pointA[1]),
(cX,cY), (0,255,0), 2)
61 cv2.line(frame_oval, (pointB[0],pointB[1]),
(cX,cY), (0,255,0), 2)
62 cv2.circle(main_frame, (pointAText[0],
pointAText[1]), 3, (255, 255, 255), -1)
63 cv2.circle(main_frame, (pointBText[0],
pointBText[1]), 3, (255, 255, 255), -1)

```

Tabel 5.8 (Lanjutan)

```

64 cv2.putText(main_frame, 'A', (pointAText[0]-
5, pointAText[1]+18), cv2.FONT_HERSHEY_PLAIN, 1,
(200, 255, 155), 1, cv2.LINE_AA)
65 cv2.putText(main_frame, 'B', (pointBText[0]+5, point
BText[1]+5), cv2.FONT_HERSHEY_PLAIN, 1,
(200, 255, 155), 1, cv2.LINE_AA)
66 areaAB = frame_oval[ y: y+int(h/2), x:
x+int(w/2)]
67 areaCD= frame_oval[ y: y+int(h/2), x+int(w/2):
x+w]
68 total_areaAB=collections.Counter(np.ravel(areaAB)
)[255]
69 total_areaCD=collections.Counter(np.ravel(areaCD)
)[255]
70 LengthAtoCenter=dist.euclidean(pointA, pointCenter
)
71 LengthBtoCenter=dist.euclidean(pointB, pointCenter
)
72 if w == h:
73 DivAandB = LengthBtoCenter / LengthAtoCenter
74 else:
75 pass
76 pkepala = h/w
77 bpera = LengthBtoCenter / LengthAtoCenter
78 navigasi()
79 cv2.putText(main_frame, 'h/w : '+
80 format(pkepala),
81 (10, 340), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(255, 255, 255), 2)
82 cv2.putText(main_frame, 'b/a : '+ format(bpera),
83 (10, 320), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(255, 255, 255), 2)
84 wkomp = millis() - startTime
85 cv2.putText(main_frame, 'w.komputasi : '+
format(wkomp), (10, 360), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (255, 255, 255), 2)

```

5.2.2.3 Implementasi Klasifikasi Pergerakan Kepala

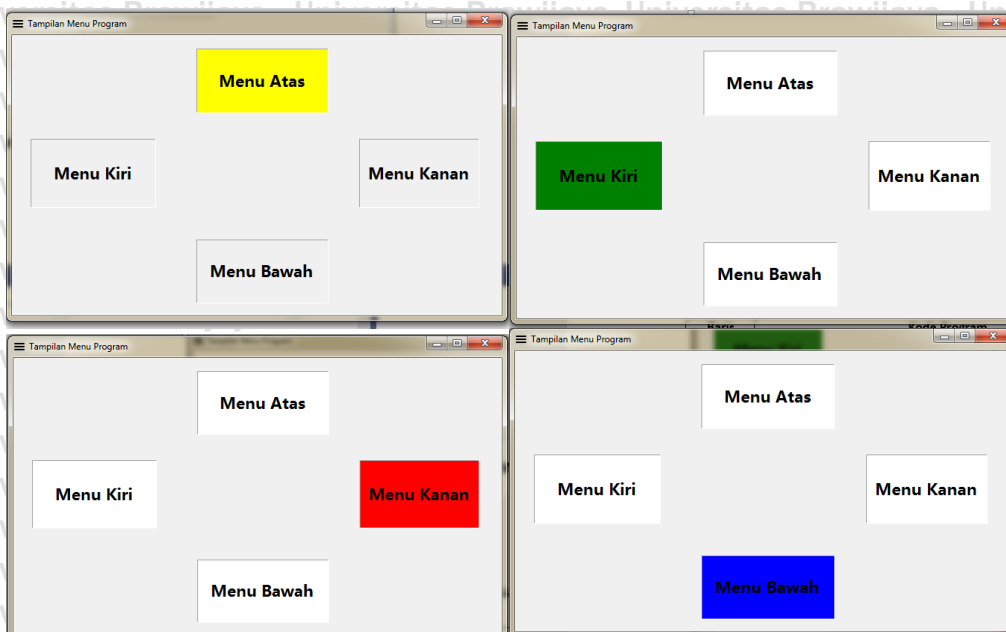
Pada tahap klasifikasi pergerakan tabel 5.9 kepala terdapat 2 metode yang digunakan yakni membuat *class* dan fungsi. Pada baris 1-5 pembuatan *class* dengan nama koordinat untuk koordinat masing masing opsi pada menu, dengan fitur autoclick dari *pyautogui*. Pada baris 6-21 merupakan fungsi bernama navigasi dengan logika jika variabel *pkepala* $\geq 1,07$ maka melihat pada nilai variabel *bpera*, jika *bpera* bernilai $\leq 1,4$ dan jumlah piksel kuadran 1 > kuadran 2 maka hasil pergerakan adalah kiri, jika *bpera* bernilai $\leq 1,4$ dan jumlah piksel kuadran 2 > kuadran 1 maka hasil pergerakan adalah kanan, jika nilai variabel *bpera* > 1,4 maka tergolong gerakan tegak. Jika seluruh kondisi tidak memenuhi maka termasuk pada gerakan menunduk

Tabel 5.9 Kode Program Penentu Hasil Klasifikasi

Baris	Kode Program
1	class kordinat():
2	korkiri = (162,225)
3	korkanan = (748,224)
4	kornunduk = (480,349)
5	kortegak = (477,98)
6	def navigasi() :
7	if pkepala >= 1.07 :
8	if ((bpera<=1.4) and (total_areaAB >
9	total_areaCD)):
10	cv2.putText(main_frame, 'arah gerak : '+ 'kiri',
11	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75,
12	(255,255,255), 2)
13	pyautogui.click(kordinat.korkiri)
14	elif ((bpera<=1.4) and (total_areaCD >
15	total_areaAB)):
16	cv2.putText(main_frame, 'w.komputasi : '+ 'kanan',
17	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75,
18	(255,255,255), 2)
19	pyautogui.click(kordinat.korkanan)
20	else :
21	cv2.putText(main_frame, 'w.komputasi : '+ 'tegak',
22	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75,
23	(255,255,255), 2)
24	pyautogui.click(kordinat.kortegak)
25	else :
26	cv2.putText(main_frame, 'w.komputasi : '+
27	'menunduk', (10,300),cv2.FONT_HERSHEY_SIMPLEX,
28	0.75, (255,255,255), 2)
29	pyautogui.click(kordinat.kornunduk)

5.2.2.4 Implementasi Menu dengan Visual Studio

Pada sub bab ini dijelaskan tentang perancangan dari menu yang berfungsi sebagai tampilan output hasil sistem. Memiliki 4 pilihan yaitu menu atas, menu bawah, menu kiri dan menu kanan. Masing masing memiliki warna tersendiri jika menerima sinyal klik dari kursor. Pada menu atas akan berwarna kuning jika menerima sinyal dari kursor dan berwarna putih jika tidak menerima sinyal kursor, menu kiri akan berwarna hijau jika menerima sinyal dari kursor dan berwarna putih jika tidak menerima sinyal kursor, menu kanan akan berwarna merah jika menerima sinyal dari kursor dan berwarna putih jika tidak menerima sinyal kursor, dan pada menu bawah akan berwarna biru jika menerima sinyal dari kursor dan berwarna putih jika tidak menerima sinyal kursor. Gambar 5.10 menampilkan masing masing opsi ketika mendapat sinyal dari kursor.



Gambar 5.10 Tampilan Menu dengan 4 Opsi

Selanjutnya pada Tabel 5.10 ditampilkan kode program dari menu berbasis Visual Studio. Pada baris 1-8 merupakan kode list dari beberapa fitur yang digunakan dalam aplikasi, Pada baris 9 merupakan inialisasi objek dengan nama AppFinal . Pada baris 10-12 merupakan inialisasi form yang digunakan. Baris 13-20 merupakan inialisasi *event* Pada 4 label yang merupakan cursor *enter* dan *leave* yang berfungsi mengeksekusi label ketika *event* dijalankan. Baris 21-36 merupakan inialisasi warna pada 4 label sesuai tampilan.

Tabel 5.10 Kode Program Menu pada Visual Studio

Baris	Kode Program
1	using System;
2	using System.Collections.Generic;
3	using System.ComponentModel;
4	using System.Data;
5	using System.Drawing;
6	using System.Linq;
7	using System.Text;
8	using System.Windows.Forms;
9	namespace AppFinal{
10	public partial class Form1 : Form{
11	public Form1(){
12	InitializeComponent();
13	label1.MouseEnter += OnMouseEnterLabel1;
14	label1.MouseLeave += OnMouseLeaveLabel1;
15	label2.MouseEnter += OnMouseEnterLabel2;
16	label2.MouseLeave += OnMouseLeaveLabel2;
17	label3.MouseEnter += OnMouseEnterLabel3;
18	label3.MouseLeave += OnMouseLeaveLabel3;
19	label4.MouseEnter += OnMouseEnterLabel4;
20	label4.MouseLeave += OnMouseLeaveLabel4; }

Tabel 5.10 Lanjutan

```
21 private void OnMouseEnterLabel1(object sender,  
22 EventArgs e){  
23     label1.BackColor = Color.Green;}  
24 private void OnMouseLeaveLabel1(object sender,  
25 EventArgs e){  
26     label1.BackColor = Color.White;}  
27 private void OnMouseEnterLabel2(object sender,  
28 EventArgs e){  
29     label2.BackColor = Color.Yellow;}  
30 private void OnMouseLeaveLabel2(object sender,  
31 EventArgs e){  
32     label2.BackColor = Color.White;}  
33 private void OnMouseEnterLabel3(object sender,  
34 EventArgs e){  
35     label3.BackColor = Color.Red;}  
36 private void OnMouseLeaveLabel3(object sender,  
EventArgs e){  
    label3.BackColor = Color.White;}  
private void OnMouseEnterLabel4(object sender,  
EventArgs e){  
    label4.BackColor = Color.Blue;}  
private void OnMouseLeaveLabel4(object sender,  
EventArgs e){  
    label4.BackColor = Color.White;}
```

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis merupakan bab yang menjelaskan hasil uji dari bab 5 yaitu perancangan dan implementasi. Pengujian dilakukan guna mendapatkan jawaban dari rumusan masalah dan mengetahui proses berjalan dari sistem dengan tujuan sebagai kesesuaian dengan hasil yang diinginkan dari penelitian. Penelitian mencakup pengaruh tingkat pencahayaan pada proses deteksi kepala dan pergerakan kepala, pengaruh jarak pada pengimplementasian sistem deteksi kepala dan pergerakan kepala pada jarak dan visualisasi masing masing pergerakan kepala dan deteksi kepala pada sistem

6.1 Pengujian pengaruh tingkat pencahayaan pada deteksi kepala

Pengujian pada tahap ini merupakan tahap pengujian tingkat deteksi objek terhadap tingkat pencahayaan dengan 2 tingkat pencahayaan yakni, pencahayaan pada siang hari dan malam hari dengan bantuan alat penerang. Nilai *threshold* dari 2 konversi warna HSV dan YCbCr merupakan faktor penentu dari pengujian ini.

6.1.1 Strategi Pengujian

Strategi pengujian dilakukan guna sebagai bentuk pengujian terhadap *threshold* dari program terhadap tingkat pencahayaan yang didapatkan oleh subjek. Penelitian dilakukan dengan cara menganalisis secara Kuantitatif, dengan perhitungan dari akurasi deteksi kepala terhadap jarak dan pencahayaan, juga menghitung nilai rata2 akurasi rekognisi masing2 pergerakan kepala pada 4 pola gerakan yaitu ke atas, bawah, kanan dan kiri.

Persentase Akurasi : $(\text{total percobaan benar} / \text{total seluruh percobaan}) * 100\%$

6.1.2 Hasil dan Analisis Pengujian

Pada analisis hasil di peroleh hasil percobaan dengan 2 fase yakni cahaya siang (1500 - 1800 lux) dan cahaya dari lampu malam hari (1200 - 1400 lux)

Tabel 6.1 Hasil Pengujian pada Perbedaan Tingkat Pencahayaan

Tipe Pergerakan	Σ Percobaan Masing-Masing Tingkat Pencahayaan	Tingkat Pencahayaan					
		Siang (1500 - 1800 lux)		Akurasi	Malam (1200 - 1400 lux)		Akurasi
		Σ Benar	Σ Salah		Σ Benar	Σ Salah	
Tegak	40	40	0	100%	36	4	90%
Menunduk		37	3	93%	37	3	93%
Kanan		37	3	93%	37	3	93%
Kiri		39	1	98%	37	3	93%
Total Akurasi		96%			92%		

Hasil menunjukkan bahwa nilai akurasi pada kondisi siang mencapai 96 % dan malam mencapai 92% dari hasil ini nilai akurasi siang lebih tinggi daripada pada saat malam hari, hal ini menunjukkan bahwa kondisi tingkat pencahayaan mempengaruhi tingkat kejelasan objek.

6.2 Pengujian terhadap jarak pada implementasi sistem deteksi kepala

Pada tahap pengujian ini sistem *Bounding box* a diuji tingkat keakurasiannya pada jarak yang bervariasi. Jarak yang dipakai pada penelitian ini antara 40cm, 50cm, 60cm dan 80cm. Pada area kontur terdapat pembatasan piksel sebagai langkah untuk mengurangi noise yang didapatkan saat program berjalan.

6.2.1 Hasil dan Analisis Pengujian

Pada hasil pengujian untuk jarak didapatkan nilai nilai pada tabel 6.2, tabel diisi berdasarkan percobaan dengan jarak objek pada kamera sesuai dengan perumusan masalah yakni 40cm,50cm,60cm dan 80cm. Jarak ini kemudian didapatkan nilai akurasi pada jarak 40cm memiliki jumlah rata-rata akurasi 94%, pada kondisi jarak 50cm sebesar 93%, jarak 60 memiliki akurasi 96% dan pada jarak 80cm rata rata untuk jarak tersebut adalah adalah 95%.

Tabel 6.2 Hasil Pengujian pada Jarak

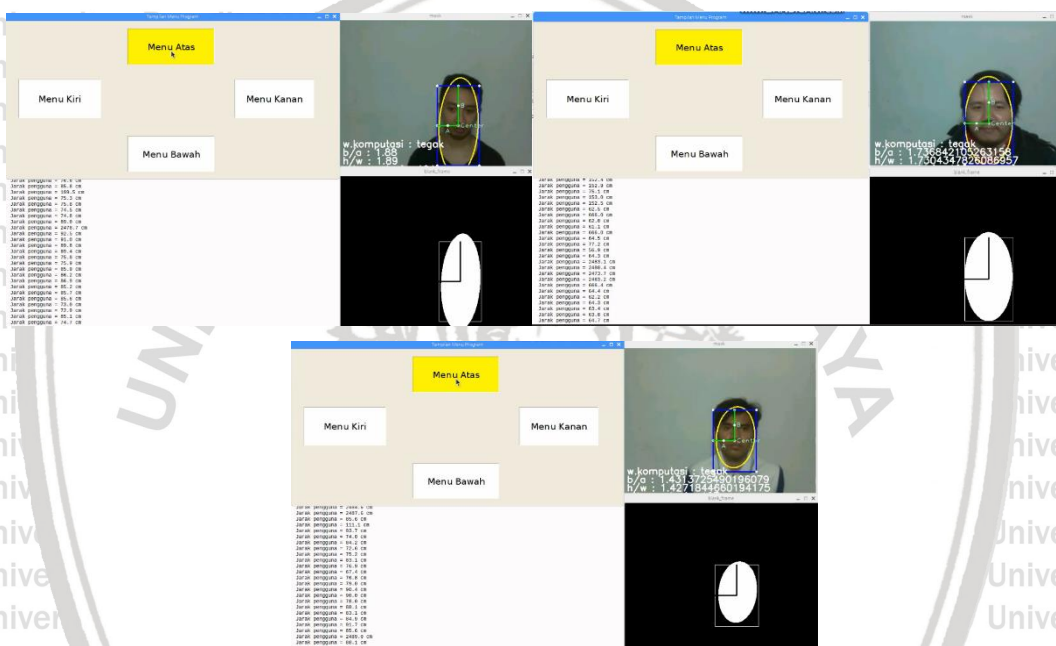
Variasi Jarak	Tipe Pergerakan	Σ Percobaan Tiap Jarak	Hasil Deteksi		Akurasi	Rata-rata Akurasi
			Σ Benar	Σ Salah		
40cm	Tegak	20	19	1	95%	94%
	Menunduk		19	1	95%	
	Kanan		19	1	95%	
	Kiri		18	2	90%	
50cm	Tegak	20	18	2	90%	93%
	Menunduk		17	3	85%	
	Kanan		20	0	100%	
	Kiri		19	1	95%	
60cm	Tegak	20	19	1	95%	96%
	Menunduk		20	0	100%	
	Kanan		18	2	90%	
	Kiri		20	0	100%	
80cm	Tegak	20	20	0	100%	95%
	Menunduk		18	2	90%	
	Kanan		19	1	95%	
	Kiri		19	1	95%	

6.3 Visualisasi masing masing pergerakan kepala dan deteksi kepala pada sistem.

Pada proses pengujian tampilan visualisasi pada sistem navigasi menu diuji tampilan sistem pada saat program dieksekusi pada sistem navigasi dan tampilan pada saat terdeteksi pergerakan pada sistem.

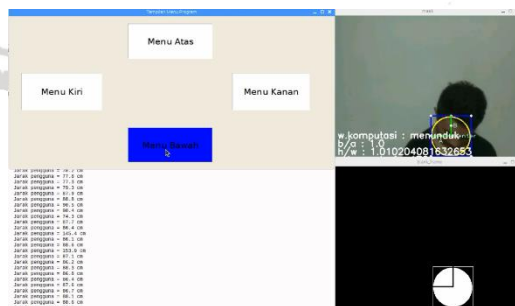
6.3.1 Hasil dan Analisis Pengujian

Hasil yang diperoleh pada pengujian visualisasi gerakan kepala didapatkan hasil yang akurat, diambil contoh masing masing 3 objek pada tiap macam gerakan.

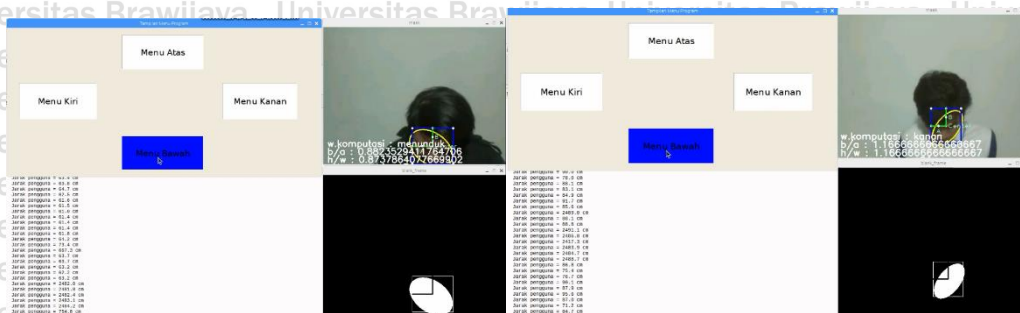


Gambar 6.1 Hasil Visualisasi dari Pergerakan ke Atas

Pada gambar 6.1 dapat dilihat dari hasil visualisasi pergerakan ke atas ketiga sampel memiliki hasil visualisasi yang tergolong baik dan sesuai dengan tujuan penelitian.

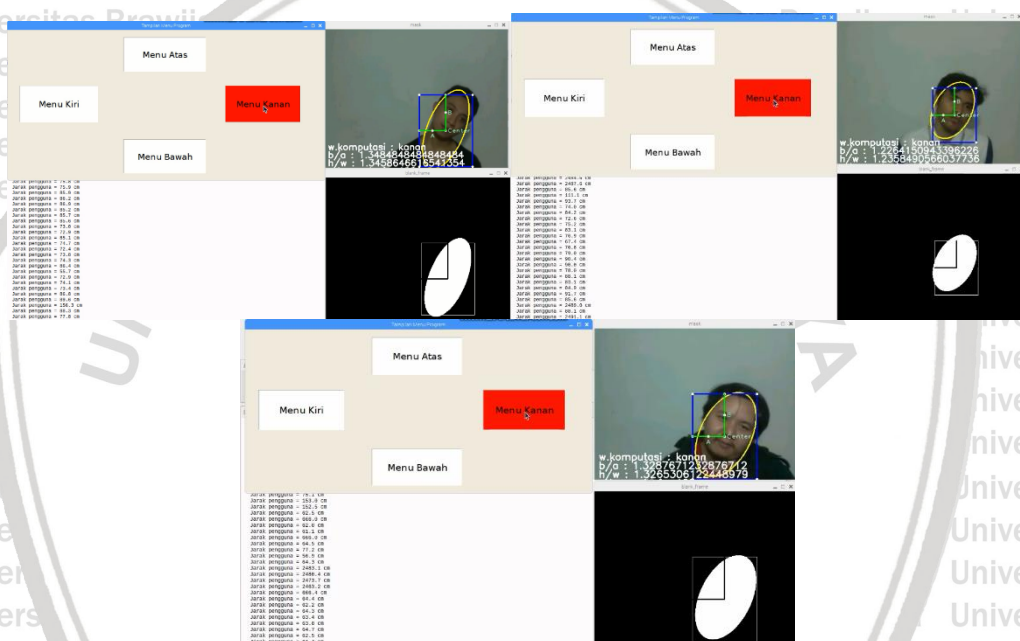


Gambar 6.2 Hasil Visualisasi dari Pergerakan ke Bawah



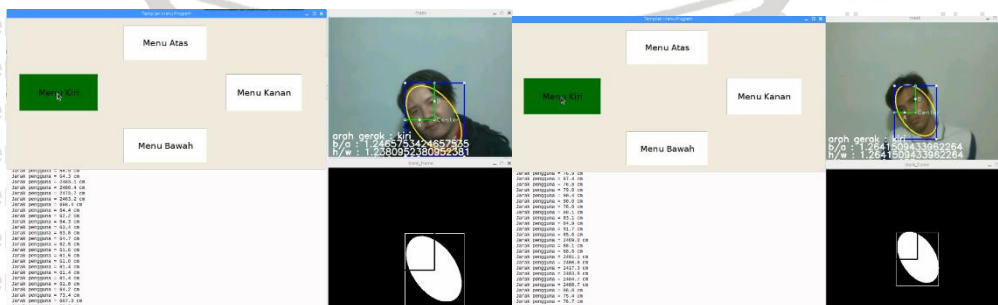
Gambar 6.2 Hasil Visualisasi dari Pergerakan ke Bawah (Lanjutan)

Pada gambar 6.2 juga dapat dilihat dari hasil visualisasi pergerakan ke atas ketiga sampel memiliki hasil visualisasi yang tergolong baik dan sesuai dengan tujuan penelitian.

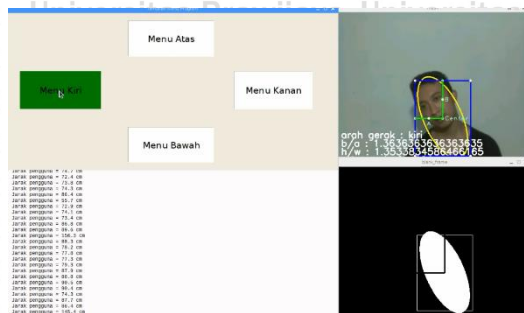


Gambar 6.3 Hasil Visualisasi dari Pergerakan ke Kanan

Pada gambar 6.3 juga masih memiliki hasil visualisasi pergerakan ke atas ketiga sampel memiliki hasil visualisasi yang tergolong baik dan sesuai dengan tujuan penelitian.



Gambar 6.4 Hasil Visualisasi dari Pergerakan ke Kiri



Gambar 6.4 Hasil Visualisasi dari Pergerakan ke Kiri (Lanjutan)

Terakhir pada gambar hasil visualisasi pergerakan ke kiri memiliki hasil yang juga baik sesuai dengan hasil yang diinginkan penulis pada penelitian.



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari rumusan masalah terdapat beberapa poin yang dapat diambil kesimpulan berdasarkan hasil yang telah diperoleh dari pengujian dan analisis yang telah dilakukan melalui penelitian. Beberapa poin kesimpulan yang dapat dijabarkan ialah :

1. Hasil menunjukkan bahwa nilai akurasi pada kondisi siang mencapai 96 % dan malam mencapai 92% dari hasil ini nilai akurasi siang lebih tinggi daripada pada saat malam hari, pengaruh tingkat pencahayaan pada proses deteksi kepala dan pergerakan kepala, untuk pencahayaan perbedaan cahaya akan mempengaruhi jika cahaya yang didapat sangat kurang, namun jika masih dalam kisaran lux normal. Hasil ditunjukkan akan lebih baik, tingkat threshold *HSV* dan *YCbCr* yang merupakan faktor penentu dalam tingkat deteksi pergerakan terhadap cahaya.
2. pengaruh jarak pada pengimplementasian sistem deteksi kepala dan pergerakan kepala pada jarak perbedaan jarak menunjukkan hasil pada jarak 40cm memiliki jumlah rata-rata akurasi 94%, pada kondisi jarak 50cm sebesar 93%, jarak 60 memiliki akurasi 96% dan pada jarak 80cm rata rata untuk jarak tersebut adalah adalah 95%. Bisa diambil kesimpulan bahwa untuk jarak terjauh yakni 60 dan 80 cm dapat memperoleh akurasi tinggi karena pada saat objek dekat tingkat sensitifitas terhadap warna akan lebih besar, fitur wajah seperti alis dan rambut dapat mempengaruhi hasil dari percobaan.
3. Visualisasi gambar sangat baik dapat dilihat pada hasil yang diperoleh dalam percobaan, hasil menunjukkan minimal area *noise* . dan oval terbentuk secara sempurna.

7.2 Saran

Setelah berjalannya penelitian, maka penulis menyarankan beberapa hal yang dapat dikembangkan kembali oleh peneliti selanjutnya. Saran tersebut didapatkan saat penelitian telah berlangsung. Poin saran adalah sebagai berikut ini :

1. Menguji metode deteksi pengolah citra selain yang digunakan dalam penelitian ini guna pembandingan dari segi performa dan tingkat akurasi pada pemrosesan deteksi kepala.
2. Melakukan pengujian dengan fitur lain selain yang digunakan oleh penulis guna pembandingan dalam maksud pengembangan secara maksimal metode pengolahan citra.
3. Mencoba alternatif mikroprocessor lain selain Raspberry Pi guna pembandingan performa deteksi dari segi hardware.
4. Peningkatan waktu komputasi akan sangat berguna dalam pengujian deteksi secara realtime.

DAFTAR REFERENSI

Ashish Kapoor, R. W. (2001). A Real-Time Head Nod and Shake Detector. *Proceedings from the Workshop on Perceptive User Interfaces*.

Ghinmine, S. a. (2017). Comparative Study of RGB, HSV & YcbCr Color Model Saliency. *Map*. 5(6), pp 2320-9801.

H. H. Goldstine, A. G. (1946). The Electronic Numerical Integrator and Computer (ENIAC).

HaolinWei, P. ,. (t.thn.). Real-Time Head Nod And Shake Detection For Continuous Human Affect Recognition. *CentreforSensor WebTechnologies, DublinCityUniversity, Ireland* .

Iswindarty, P. (2013). *Pengolahan Citra Digital Semester 1* . Malang : Kementerian Pendidikan & Kebudayaan .

Jose, A. (2014, Mei 27). Perjalanan Panjang Terciptanya Teknologi Layar Sentuh. Diambil kembali dari Okezone Techno: <https://techno.okezone.com/read/2014/05/27/363/990571/perjalanan-panjang-terciptanya-teknologi-layar-sentuh>

Konrad Baumann, B. T. (2001). User interface design for electronic appliances.

Prasetyo, M. E. (2011). Teori Dasar Hidden Markov Model. *Makalah I/2092 Probabilitas dan Statistik* .

Prof. P Y Kumbhar, M. A. (2017). Real Time Face Detection and Tracking Using OpenCV . *International Journal For Research In Emerging Science And Technology, Volume-4*.

Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012). Realtime Computer Vision with OpenCV. 40:40–40:56.

Rafael C.Gonzales, R. E. (2014). Digital Image Processing. Tennessee.

S Kolkur, D. K. (2017). Human Skin Detection Using RGB, HSV and YCbCr Color Models. *arXiv preprint arXiv:1708.02694*.

Thad Starner, A. P. (1995). Real-Time American Sign Language Visual Recognition From Video Using Hidden Markov Models. *Master's Thesis, MIT, Feb 1995, Program in Media Arts*.

Vignesh. S PG Scholar, M. (2014). Real Time Hand Gesture Recognition for Human Machine Communication Using ARM Cortex A-8 . *IOSR Journal of Computer Engineering (IOSR-JCE)* , 43-48.

LAMPIRAN

Lampiran Kode Program

Baris	Kode Program
1	import cv2
2	import numpy as np
3	import time
4	import collections
5	import imutils
6	import pyautogui
7	from PIL import Image
8	from scipy.spatial import distance as dist
9	class kordinat():
10	korkiri = (162,225)
11	korkanan = (748,224)
12	kornunduk = (480,349)
13	kortegak = (477,98)
14	def navigasi() :
15	if pkepala >= 1.07 :
16	if ((bpera<=1.4) and (total_areaAB > total_areaCD)):
17	cv2.putText(main_frame, 'arah gerak : '+ 'kiri',
18	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
19	2)
20	pyautogui.click(kordinat.korkiri)
21	elif ((bpera<=1.4) and (total_areaCD > total_areaAB)):
22	cv2.putText(main_frame, 'w.komputasi : '+ 'kanan',
23	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
24	2)
25	pyautogui.click(kordinat.korkanan)
26	else :
27	cv2.putText(main_frame, 'w.komputasi : '+ 'tegak',
28	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
29	2)
30	pyautogui.click(kordinat.kortegak)
31	else :
32	cv2.putText(main_frame, 'w.komputasi : '+ 'menunduk',
33	(10,300),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
34	2)
35	pyautogui.click(kordinat.kornunduk)
36	cap= cv2.VideoCapture(0)
37	while cap.isOpened():
38	frame= cap.read()
39	frame = imutils.resize(frame, width=480)
40	flip = cv2.flip(frame, 1)
41	main_frame = cv2.GaussianBlur(flip,(5,5),0)
42	ycbcr = cv2.cvtColor(main_frame, cv2.COLOR_BGR2YCrCb)
43	min ycbcr = np.array ([0,132,76])



```
36 max_ybcbcr = np.array ([255,173,127])
37 ybcbcrmask = cv2.inRange(ybcbcr, min_ybcbcr, max_ybcbcr)
38 hsv = cv2.cvtColor(main_frame, cv2.COLOR_BGR2HSV)
39 min_hsv = np.array ([0,40,0])
40 max_hsv = np.array ([25,255,255])
41 hsvmask = cv2.inRange(hsv, min_hsv, max_hsv)
42 mask = cv2.add(ybcbcrmask,hsvmask)
43 kernel = np.ones((3,3), np.uint8)
44 thresh_erosi = cv2.erode(mask, kernel, iterations=5)
45 dilasi = cv2.dilate(thresh_erosi, kernel, iterations=6)
46 dilasi_cp = dilasi.copy()
47 hh, ww = dilasi.shape[:2]
48 kern = np.zeros((hh+2, ww+2), np.uint8)
49 dilasi_cp = cv2.bitwise_not(dilasi_cp)
50 res_dilasi = dilasi | dilasi_cp
51 _, contours, _ = cv2.findContours(res_dilasi,
52 cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
53 bitwise = cv2.bitwise_and(main_frame, main_frame,
54 mask=res_dilasi)
55 frame_oval = np.zeros((480,480,1), np.uint8)
56 for cnt in contours :
57 area = cv2.contourArea(cnt)
58 if area > 500 :
59 xmin = 360
60 ymin = 360
61 xmax = 0
62 ymax = 0
63 for c in cnt :
64 xx = c[0][0]
65 yy = c[0][1]
66 if (xx < xmin):
67 xmin = xx
68 if (yy < ymin):
69 ymin = yy
70 if (xx > xmax):
71 xmax = xx
72 if (yy > ymax):
73 ymax = yy
74 x = xmin
75 y = ymin
76 w = xmax - xmin
77 h = ymax - ymin
78 x1 = x + w
79 y1 = y + h
80 ellipse = cv2.fitEllipse(cnt)
91 cv2.ellipse(frame_oval,ellipse,[255],-1)
92 cv2.ellipse(main_frame,ellipse,(0,255,255), 2)
93 cv2.rectangle(main_frame,(x,y),(x+w,y+h),
94 (255,0,0),2)
95 cv2.rectangle(frame_oval,(x,y),(x+w,y+h), [255],1)
96 cX = int((x+w/2))
97 cY = int((y+(y+h))/2)
98 pointCenter = (cX,cY)
```

```

99 cv2.circle(main_frame, (cx, cy), 3, (255, 255, 255), -
100 1)
101 cv2.putText(main_frame, 'Center', (cx+5, cy+5),
102 cv2.FONT_HERSHEY_PLAIN, 1, (200, 255, 155), 1,
103 cv2.LINE_AA)
104 point1 = (x, y)
105 point2 = (x+w, y)
106 point3 = (x1, y1)
107 point4 = (x, y1)
108 cv2.circle(main_frame, (point1[0], point1[1]), 3, (255,
109 255, 255), -1)
110 cv2.circle(main_frame, (point2[0], point2[1]), 3, (255,
111 255, 255), -1)
112 cv2.circle(main_frame, (point3[0], point3[1]), 3, (255,
113 255, 255), -1)
114 cv2.circle(main_frame, (point4[0], point4[1]), 3, (255,
115 255, 255), -1)
116 PointAx = int((point1[0] + point4[0])/2)
117 PointAy = int((point1[1] + point4[1])/2)
118 PointBx = int((point1[0] + point2[0])/2)
119 PointBy = int((point1[1] + point2[1])/2)
120 pointA = (PointAx, PointAy) #kolom, baris
121 pointB = (PointBx, PointBy)
122 pointATextx = int((pointA[0] + cx)/2)
123 pointATexty = int((pointA[1] + cy)/2)
124 pointAText = (pointATextx, pointATexty)
125 pointBTextx = int((pointB[0] + cx)/2)
126 pointBTexty = int((pointB[1] + cy)/2)
127 pointBText = (pointBTextx, pointBTexty)
128 cv2.circle(main_frame, (pointA[0], pointA[1]), 3, (255,
129 255, 255), -1)
130 cv2.circle(main_frame, (pointB[0], pointB[1]), 3,
131 (255, 255, 255), -1)
132 cv2.line(main_frame, (pointA[0], pointA[1]), (cx, cy),
133 (0, 255, 0), 2)
134 cv2.line(main_frame, (pointB[0], pointB[1]), (cx, cy),
135 (0, 255, 0), 2)
136 cv2.line(frame_oval, (pointA[0], pointA[1]), (cx, cy),
137 (0, 255, 0), 2)
138 cv2.line(frame_oval, (pointB[0], pointB[1]), (cx, cy),
139 (0, 255, 0), 2)
140 cv2.circle(main_frame, (pointAText[0], pointAText[1]),
141 3, (255, 255, 255), -1)
142 cv2.circle(main_frame, (pointBText[0], pointBText[1]),
143 3, (255, 255, 255), -1)
144 cv2.putText(main_frame, 'A', (pointAText[0]-
145 5, pointAText[1]+18), cv2.FONT_HERSHEY_PLAIN, 1,
146 (200, 255, 155), 1, cv2.LINE_AA)
147 cv2.putText(main_frame, 'B', (pointBText[0]+5, pointBText[1]+
148 5), cv2.FONT_HERSHEY_PLAIN, 1, (200, 255, 155), 1,
149 cv2.LINE_AA)
150 areaAB = frame_oval[ y: y+int(h/2), x: x+int(w/2)]
151 areaCD = frame_oval[ y: y+int(h/2), x+int(w/2): x+w]

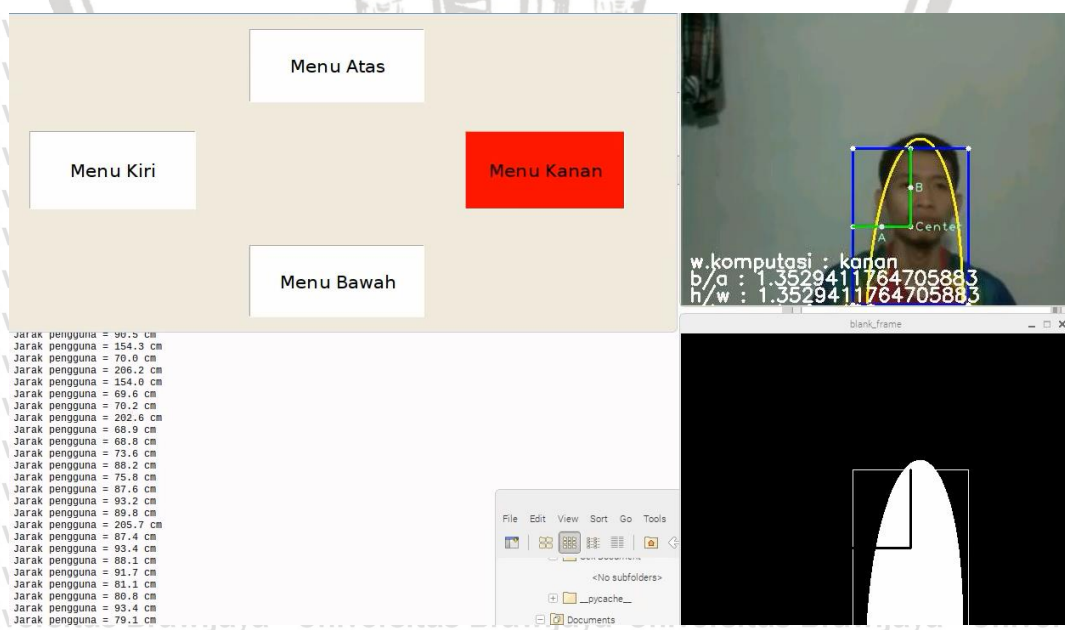
```

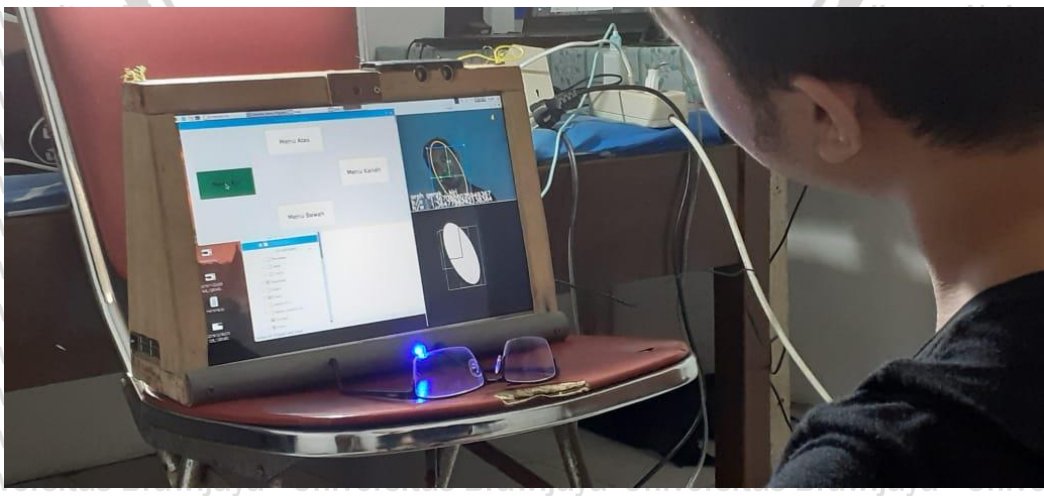
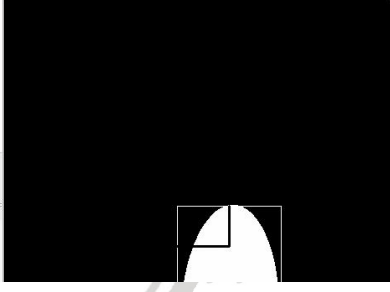
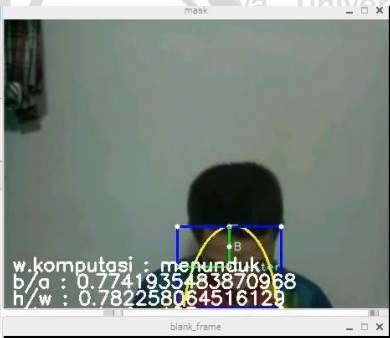
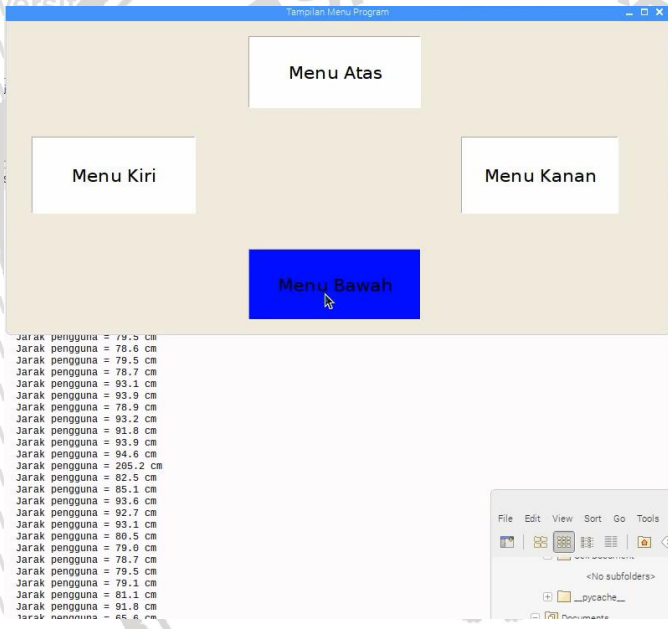
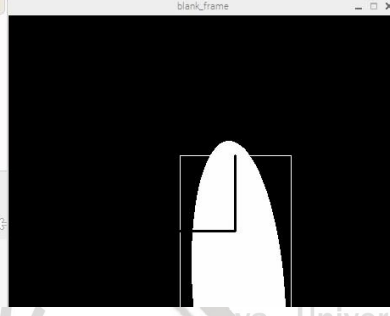
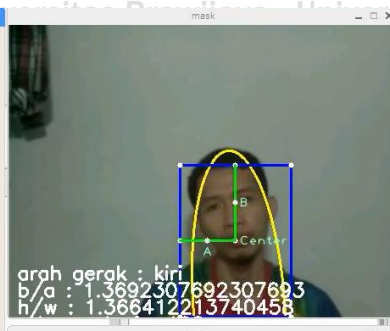
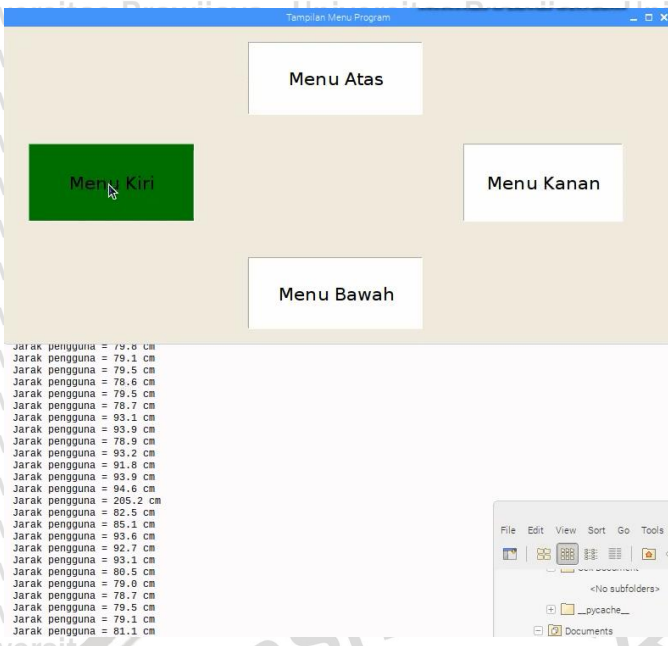
```

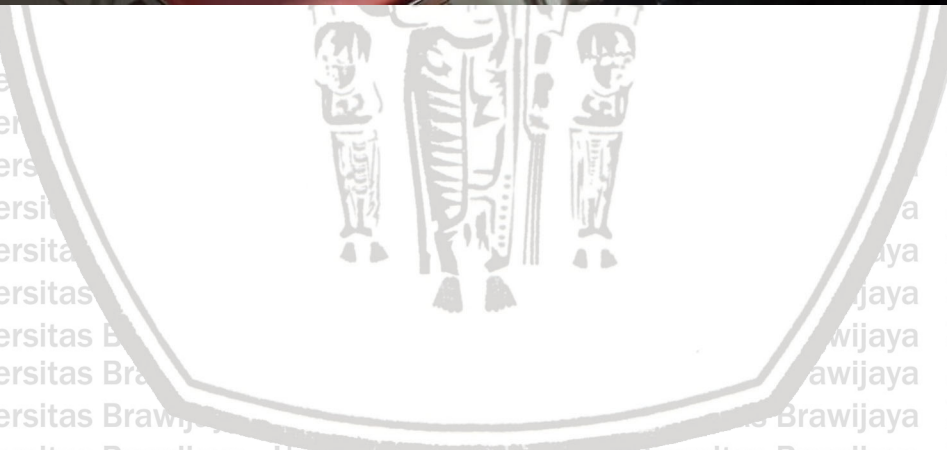
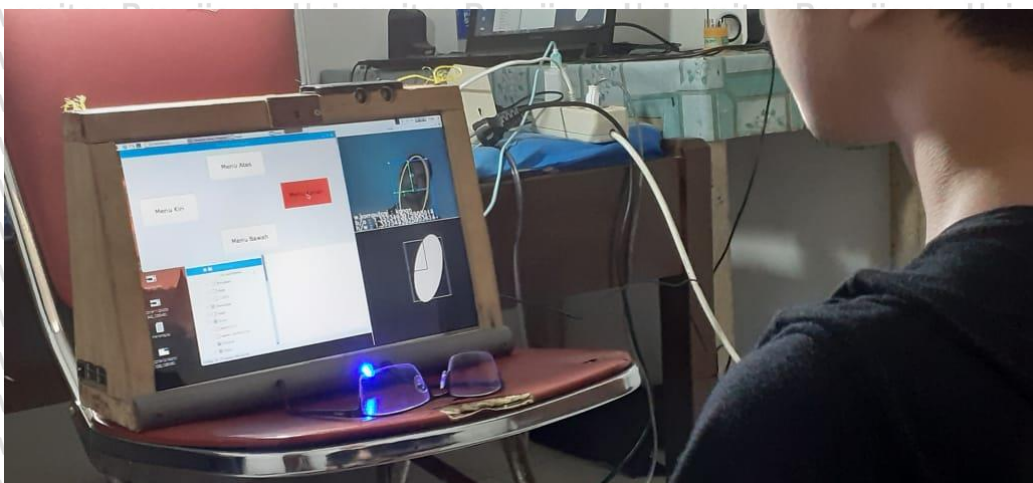
136 total_areaAB=collections.Counter(np.ravel(areaAB)) [255]
137 total_areaCD=collections.Counter(np.ravel(areaCD)) [255]
138 LengthAtoCenter=dist.euclidean(pointA,pointCenter)
139 LengthBtoCenter=dist.euclidean(pointB,pointCenter)
140 if w == h:
141     DivAandB = LengthBtoCenter / LengthAtoCenter
142 else:
143     pass
144     pkepala = h/w
145     bpera = LengthBtoCenter / LengthAtoCenter
146     navigasi()
147     cv2.putText(main_frame, 'h/w : '+ format(pkepala),
148     (10,340),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
149     2)
149     cv2.putText(main_frame, 'b/a : '+ format(bpera),
150     (10,320),cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255,255,255),
151     2)
150     wkomp = millis() - startTime
151     cv2.putText(main_frame, 'w.komputasi : '+
152     format(wkomp), (10,360),cv2.FONT_HERSHEY_SIMPLEX, 0.75,
153     (255,255,255), 2)
152     cv2.floodFill(dilasi_cp, kern, (0,0), 255)
153     cv2.imshow ('mask',main_frame)
154     cv2.imshow ('frame_oval', frame_oval)
155     if cv2.waitKey(1) & 0xFF == ord('q'):
156         break
157     cv2.destroyAllWindows()
158     cap.release()

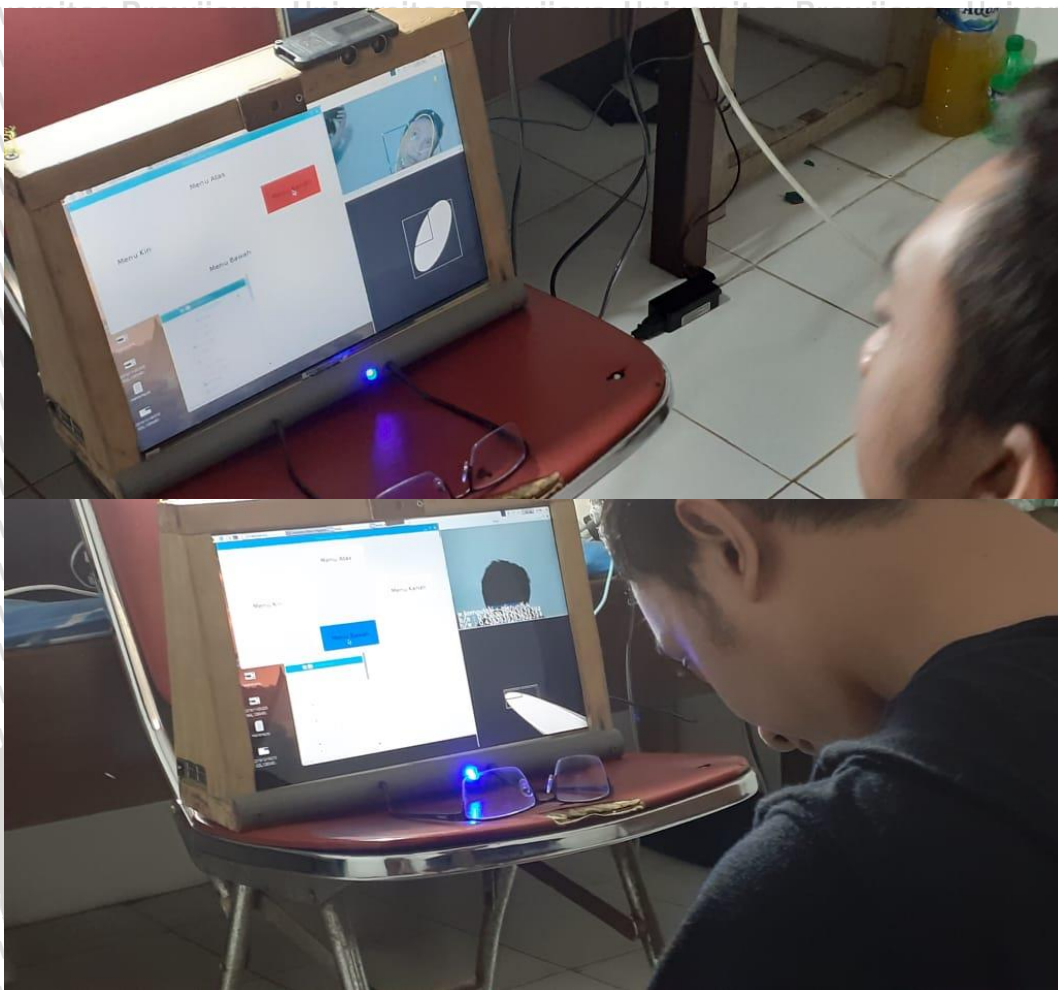
```

Lampiran Foto Pengujian









Lampiran Tabel Hasil Percobaan
Malam Hari

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	40	Kanan	Kanan	Benar	100	Faris	40	Atas	Tegak	Benar	90
Fau			Kanan	Benar		Fau			Tegak	Benar	
Budi			Kanan	Benar		Budi			Tegak	Benar	
Jo			Kanan	Benar		Jo			Tegak	Benar	
Giri			Kanan	Benar		Giri			Bawah	Salah	
Nikma			Kanan	Benar		Nikma			Tegak	Benar	
Ayu			Kanan	Benar		Ayu			Tegak	Benar	
Tiqo			Kanan	Benar		Tiqo			Tegak	Benar	
Radea			Kanan	Benar		Radea			Tegak	Benar	
Verro			Kanan	Benar		Verro			Tegak	Benar	
User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	40	Kiri	Kiri	Benar	80	Faris	40	Bawah	Bawah	Benar	90
Fau			Kiri	Benar		Fau			Bawah	Benar	
Budi			Kiri	Benar		Budi			Bawah	Benar	
Jo			Kiri	Benar		Jo			Bawah	Benar	
Giri			Bawah	Salah		Giri			Bawah	Benar	
Nikma			Kiri	Benar		Nikma			Bawah	Benar	
Ayu			kanan	Salah		Ayu			Bawah	Benar	
Tiqo			Kiri	Benar		Tiqo			Kanan	Salah	
Radea			Kiri	Benar		Radea			Bawah	Benar	
Verro			Kiri	Benar		Verro			Bawah	Benar	



User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	50	Kanan	Kanan	Benar	100	Faris	50	Atas	Tegak	Benar	80
Fau			Kanan	Benar		Fau			Tegak	Benar	
Budi			Kanan	Benar		Budi			Kanan	Salah	
Jo			Kanan	Benar		Jo			Tegak	Benar	
Giri			Kanan	Benar		Giri			Kanan	Salah	
Nikma			Kanan	Benar		Nikma			Tegak	Benar	
Ayu			Kanan	Benar		Ayu			Tegak	Benar	
Tiqo			Kanan	Benar		Tiqo			Tegak	Benar	
Radea	Kanan	Benar	Radea	Tegak	Benar						
Verro	Kanan	Benar	Verro	Tegak	Benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	50	Kiri	Kiri	Benar	90	Faris	50	Bawah	Kanan	Salah	80
Fau			Kiri	Benar		Fau			Bawah	Benar	
Budi			Kiri	Benar		Budi			Kanan	Salah	
Jo			Kiri	Benar		Jo			Bawah	Benar	
Giri			Bawah	Salah		Giri			Bawah	Benar	
Nikma			Kiri	Benar		Nikma			Bawah	Benar	
Ayu			Kiri	Benar		Ayu			Bawah	Benar	
Tiqo			Kiri	Benar		Tiqo			Kanan	Salah	
Radea	Kiri	Benar	Radea	Bawah	Benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	60	Kanan	Kanan	Benar	90	Faris	60	Atas	Tegak	Benar	90
Fau			Kanan	Benar		Fau			Tegak	Benar	
Budi			Kanan	Benar		Budi			Tegak	Benar	
Jo			Kanan	Benar		Jo			Tegak	Benar	
Giri			Bawah	Salah		Giri			Kanan	Salah	
Nikma			Kanan	Benar		Nikma			Tegak	Benar	
Ayu			Kanan	Benar		Ayu			Tegak	Benar	
Tiqo			Kanan	Benar		Tiqo			Tegak	Benar	
Radea	Kanan	Benar	Radea	Tegak	Benar						
Verro	Kanan	Benar	Verro	Tegak	Benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	60	Kiri	Kiri	Benar	100	Faris	60	Bawah	Bawah	Benar	100
Fau			Kiri	Benar		Fau			Bawah	Benar	
Budi			Kiri	Benar		Budi			Bawah	Benar	
Jo			Kiri	Benar		Jo			Bawah	Benar	
Giri			Kiri	Benar		Giri			Bawah	Benar	
Nikma			Kiri	Benar		Nikma			Bawah	Benar	
Ayu			Kiri	Benar		Ayu			Bawah	Benar	
Tiqo			Kiri	Benar		Tiqo			Bawah	Benar	
Radea	Kiri	Benar	Radea	Bawah	Benar						
Verro	Kiri	Benar	Verro	Bawah	Benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	80	Kanan	Kanan	Benar	100	Faris	80	Atas	Tegak	Benar	100
Fau			Kanan	Benar		Fau			Tegak	Benar	
Budi			Kanan	Benar		Budi			Tegak	Benar	
Jo			Kanan	Benar		Jo			Tegak	Benar	
Giri			Kanan	Benar		Giri			Kanan	Benar	
Nikma			Kanan	Benar		Nikma			Tegak	Benar	
Ayu			Kanan	Benar		Ayu			Tegak	Benar	
Tiqo			Kanan	Benar		Tiqo			Tegak	Benar	
Radea	Kanan	Benar	Radea	Tegak	Benar						
Verro	Kanan	Benar	Verro	Tegak	Benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	80	Kiri	Kiri	Benar	100	Faris	80	Bawah	Bawah	Benar	100
Fau			Kiri	Benar		Fau			Bawah	Benar	
Budi			Kiri	Benar		Budi			Bawah	Benar	
Jo			Kiri	Benar		Jo			Bawah	Benar	
Giri			Kiri	Benar		Giri			Bawah	Benar	
Nikma			Kiri	Benar		Nikma			Bawah	Benar	
Ayu			Kiri	Benar		Ayu			Bawah	Benar	
Tiqo			Kiri	Benar		Tiqo			Bawah	Benar	
Radea	Kiri	Benar	Radea	Bawah	Benar						
Verro	Kiri	Benar	Verro	Bawah	Benar						



Siang Hari

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	40	Kanan	kanan	benar	90	Faris	40	Atas	tegak	benar	100
Fau			kanan	benar		Fau			tegak	benar	
Budi			kanan	benar		Budi			tegak	benar	
Jo			kanan	benar		Jo			tegak	benar	
Giri			kanan	benar		Giri			tegak	benar	
Nikma			menunduk	salah		Nikma			tegak	benar	
Ayu			kanan	benar		Ayu			tegak	benar	
Tiqo			kanan	benar		Tiqo			tegak	benar	
Radea	kanan	benar	Radea	tegak	benar						
Verro	kanan	benar	Verro	tegak	benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	40	Kiri	kiri	benar	100	Faris	40	Bawah	menunduk	benar	100
Fau			kiri	benar		Fau			menunduk	benar	
Budi			kiri	benar		Budi			menunduk	benar	
Jo			kiri	benar		Jo			menunduk	benar	
Giri			kiri	benar		Giri			menunduk	benar	
Nikma			kiri	benar		Nikma			menunduk	benar	
Ayu			kiri	benar		Ayu			menunduk	benar	
Tiqo			kiri	benar		Tiqo			menunduk	benar	
Radea	kiri	benar	Radea	menunduk	benar						
Verro	kiri	benar	Verro	menunduk	benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	50	Kanan	kanan	benar	100	Faris	50	Atas	tegak	benar	100
Fau			kanan	benar		Fau			tegak	benar	
Budi			kanan	benar		Budi			tegak	benar	
Jo			kanan	benar		Jo			tegak	benar	
Giri			kanan	benar		Giri			tegak	benar	
Nikma			kanan	benar		Nikma			tegak	benar	
Ayu			kanan	benar		Ayu			tegak	benar	
Tiqo			kanan	benar		Tiqo			tegak	benar	
Radea	kanan	benar	Radea	tegak	benar						
Verro	kanan	benar	Verro	tegak	benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	50	Kiri	kiri	benar	100	Faris	50	Bawah	menunduk	benar	90
Fau			kiri	benar		Fau			menunduk	benar	
Budi			kiri	benar		Budi			menunduk	benar	
Jo			kiri	benar		Jo			menunduk	benar	
Giri			kiri	benar		Giri			kanan	salah	
Nikma			kiri	benar		Nikma			menunduk	benar	
Ayu			kiri	benar		Ayu			menunduk	benar	
Tiqo			kiri	benar		Tiqo			menunduk	benar	
Radea	kiri	benar	Radea	menunduk	benar						
Verro	kiri	benar	Verro	menunduk	benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	60	Kanan	kanan	benar	90	Faris	60	Atas	tegak	benar	100
Fau			kanan	benar		Fau			tegak	benar	
Budi			kanan	benar		Budi			tegak	benar	
Jo			kanan	benar		Jo			tegak	benar	
Giri			kanan	benar		Giri			tegak	benar	
Nikma			kanan	benar		Nikma			tegak	benar	
Ayu			tegak	salah		Ayu			tegak	benar	
Tiqo			kanan	benar		Tiqo			tegak	benar	
Radea	kanan	benar	Radea	tegak	benar						
Verro	kanan	benar	Verro	tegak	benar						

User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	60	Kiri	kiri	benar	100	Faris	60	Bawah	menunduk	benar	100
Fau			kiri	benar		Fau			menunduk	benar	
Budi			kiri	benar		Budi			menunduk	benar	
Jo			kiri	benar		Jo			menunduk	benar	
Giri			kiri	benar		Giri			menunduk	benar	
Nikma			kiri	benar		Nikma			menunduk	benar	
Ayu			tegak	salah		Ayu			menunduk	benar	
Tiqo			kiri	benar		Tiqo			menunduk	benar	
Radea	kiri	benar	Radea	menunduk	benar						
Verro	kiri	benar	Verro	menunduk	benar						



User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	80	Kanan	tegak	salah	90	Faris	80	Atas	tegak	benar	100
Fau			kanan	benar		Fau			tegak	benar	
Budi			kanan	benar		Budi			tegak	benar	
Jo			kanan	benar		Jo			tegak	benar	
Giri			kanan	benar		Giri			tegak	benar	
Nikma			kanan	benar		Nikma			tegak	benar	
Ayu			kanan	benar		Ayu			tegak	benar	
Tiqo			kanan	benar		Tiqo			tegak	benar	
Radea			kanan	benar		Radea			tegak	benar	
Verro			kanan	benar		Verro			tegak	benar	
User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %	User	Jarak (cm)	Kondisi Aktual	Hasil Sistem	Akurasi Sistem	Akurasi Pergerakan %
Faris	80	Kiri	kiri	benar	90	Faris	80	Bawah	kanan	salah	80
Fau			kiri	benar		Fau			menunduk	benar	
Budi			kiri	benar		Budi			menunduk	benar	
Jo			kiri	benar		Jo			menunduk	benar	
Giri			kiri	benar		Giri			menunduk	benar	
Nikma			kiri	benar		Nikma			menunduk	benar	
Ayu			kiri	benar		Ayu			menunduk	benar	
Tiqo			kiri	benar		Tiqo			menunduk	benar	
Radea			kiri	benar		Radea			kirri	salah	
Verro			tegak	salah		Verro			menunduk	benar	

