DETEKSI PERGERAKAN BOLA MATA DAN KEDIPAN MATA UNTUK PENGGANTI POINTER YANG DITERAPKAN SEBAGAI KONTROL MENU LAYAR NAVIGASI PADA SMART WHEELCHAIR

TESIS

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Magister Komputer

Disusun oleh:
Gusti Pangestu
NIM: 176150100111034

PROGRAM STUDI MAGISTER ILMU KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

PENGESAHAN

DETEKSI PERGERAKAN BOLA MATA DAN KEDIPAN MATA UNTUK PENGGANTI POINTER YANG DITERAPKAN SEBAGAI KONTROL MENU LAYAR NAVIGASI PADA SMART WHEELCHAIR

TESIS

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun Oleh:
Gusti Pangestu
NIM:176150100111034

Tesis ini telah diuji dan dinyatakan lulus pada
27 Mei 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Eng. Fitri Utaminingrum, S.T., M.T.	Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng.
NIP: 19820710 200812 2	NIK:2012018406281001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

DAFTAR ISI

BAB I **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Tujuan 5

1.4 Manfaat Penelitian 5

1.5 Batasan Masalah 5

BAB 2 ..... **6**

2.1 Populasi disabilitas di Indonesia 6

2.2 Penelitian Terkait 7

2.3 Mata 8

2.3.1. Penggunaan mata dalam teknologi 9

2.4 Deteksi area mata 11

2.4.1. Deteksi bola mata 11

2.4.2. Face Landmark 13

2.5. Pengolahan Citra 13

2.5.1. Morfologi Citra 14

2.5.2. Konversi Citra Berwarna 16

2.5.3. Resize 17

2.6. Naïve Bayes 18

2.7. Pixel Value 19

BAB 3 ..... **21**

Tahapan Penelitian 21

3.1. Studi Literatur 22

3.2. Skenario Perancangan Program 22

3.2.1. Deteksi Area Mata dengan Face Landmark 23

3.2.2. Morfologi Citra 26

3.2.3. Eye Sector Area (ESA) 31

3.2.4. Deteksi gerakan mata dengan metode *Naïve Bayes Eye Sector Area* (NBESA) 36

3.2.5. Penentuan Arah Gerak Bola Mata Berdasarkan Data Registrasi 38

3.3. Skenario Pengujian 40

3.4. Implementasi 43

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

3.4.1. Face Landmark 44
3.4.2. Morfologi Citra 45
3.4.3. Eye Sector Area (ESA) 46
3.4.4. Naïve Bayes Eye Sector Area (NBESA) 47
3.4.4.1. Naïve Bayes 47
3.4.4.2. Pengujian Naïve Bayes 48
BAB 4 52
4.1. Spesifikasi Pengujian 52
4.2. Deteksi Mata 52
4.2.1. Pengaruh Jarak Terhadap Hasil Deteksi Mata 53
4.2.2. Pengaruh ketersediaan cahaya, jarak dan nilai threshold pada hasil deteksi arah tengah 65
4.3. Pengujian akurasi metode *Naïve Bayes Eye Sector Area* (NBESA) untuk mendeteksi arah gerakan mata dan kedipan mata dengan parameter yang sudah didapat 70
4.3.1. Precision dan Recall Metode Naïve Bayes Eye Sector Area (NBESA) 72
4.3.2. Perbandingan dengan metode lain 74
4.4. Integrasi metode NBESA pada tampilan layar berisi menu untuk deteksi gerakan bola mata dan kedipan mata 76
BAB 5 81
5.1Kesimpulan 81
5.2. Saran 82
DAFTAR PUSTAKA 83

BAB I
PENDAHULUAN

1.1 Latar Belakang

Terdapat 21.930.529 orang atau sekitar 8% dari seluruh penduduk Indonesia yang mengalami disabilitas (Kemenkes RI, 2017). Disabilitas disini didefinisikan sebagai tidak mampunya seseorang dalam melakukan kegiatan normal seperti pada umumnya. Penyandang disabilitas membutuhkan tambahan dan bantuan ekstra dari orang sekelilingnya ataupun dari peralatan yang ia miliki untuk membantu setiap aktivitasnya. Dari total penyandang disabilitas terdapat 7.808.377 jiwa penyandang disabilitas yang mengalami gannggguan berjalan, dan terdapat tidak kurang dari 2.718.290 jiwa peyandang disabilitas yang mengalami gangguan dalam menggerakan tangan atau jari mereka. Dari beberapa paparan data tersebut, mengingat jumlah penyandang disabilitas yang tidak sedikit, maka perlu adanya perhatian khusus terhadap para penyandang disabilitas agar mereka juga memiliki kesempatan dan hak yang sama untuk melakukan aktivitas mereka layaknya orang normal lainnya.

Masalah navigasi memang menjadi fokus yang dikembangkan oleh beberapa peneliti untuk membantu para penyandang disabilitas. Banyak penelitian yang bertujuan untuk membantu para penyandang disabilitas, seperti diciptakannya kursi roda ataupun juga kursi roda pintar. Kursi roda tersebut memiliki kemampuan untuk bernavigasi sesuai dengan kehendak user yang dapat dikontrol dengan menggunakan berbagai macam cara seperti menggunakan gerakan jari, tangan dan bahkan menggunakan gelombang suara. Namun, bagi penyandang disabilitas ganda yang tdiak mampu untuk menggerakan tangan dan kaki mereka, hal tersebut menjadi sulit dikarenakan keterbatasan mereka dalam menggunakan tangan atau jari mereka. Perlu adanya solusi lain yang mampu menghubungkan antara penyandang disabilitas yang tidak mampu menggunakan tangan dan kaki mereka dengan sistem komputer yang nantinya diharapkan dapat membantu para penyandang disabilitas.

Solusi yang diajukan adalah dengan menggunakan mata sebagai pengganti alat kontrol sistem komputer yang digunakan untuk memilih menu-menu navigasi yang ada pada layar *smart wheelchair*. Penggunaan mata dipilih dikarenakan beberapa alasan, diantaranya adalah letak posisi mata yang berada di wajah sehingga mudah untuk ditangkap kamera, juga fungsi mata yang sangat vital dalam kehidupan sehari-hari. Pergerakan mata yang fleksibel juga menjadi pertimbangan, kemampuan bola mata untuk bergerak seperti ke atas, bawah, kanan dan kiri menjadikan bola mata semakin memiliki kriteria untuk dipilih menjadi salah satu alat kontrol untuk penggerak elektronik yang dapat juga diaplikasikan pada alat bantu navigasi seperti kursi roda dan beberapa teknologi lainnya.

Banyak penelitian yang berfokus pada mata sebagai alat kontrol penggerak, penelitian-penelitian tersebut menggunakan media kamera untuk menangkap pergerakan mata yang kemudian di hubungkan dengan sebuah komputer dan menggunakan beberapa algoritma untuk menampilkan hasil sesuai dengan gerakan mata yang telah disepakati. Ada beberapa penelitian yang mengangkat subjek tentang mata, salah satunya adalah penelitian tentang pendeteksian pergerakan bola mata dengan menggunakan *Gaze detection* untuk digunakan seabagai verifikator (Siripitakchai & Phimoltares, 2017). Penelitian lain tentang pergerakan mata juga dilakukan dengan menggunakan pendekatan *Hough Circle* untuk kontrol navigasi kursi roda pintar (Utaminingrum, Fauzi, & Sari, n.d.), namun metode ini terbatas hanya pada arah kiri, kanan dan depan saja. Penelitian lain tentang deteksi pergerakan bola mata juga dilakukan dengan menggunakan kamera infra merah yang terpasang pada *dashboard* mobil untuk mendeteksi adanya kelelahan pada pengendara mobil (Catalbas, Cegovnik, Sodnik, & Gulten, n.d.), namun metode atau langkah ini membutuhkan kamera infra merah yang harus terpasang, sedangkan kamera infra merah memiliki harga yang cukup tinggi. Ada pula penelitian yang menggunakan *electroenchepaloghrapic (EEG)* untuk mendeteksi adanya gerakan pada mata (Sotelo et al., 2018), namun penggunaan EEG tergolong cukup rumit dikarenakan selain harga yang cukup mahal, juga tidak praktisnya penggunaan dikarenakan harus memasang bebarapa elektroda di area kepala. Maka dari itu, pada penelitian ini akan dibahas bagaimana langkah atau metode untuk mendeteksi dan membaca gerakan pada mata untuk diaplikasikan pada sebuah sistem tampilan menu pada sebuah layar LCD dengan menggunakan kamera *webcam* yang berbiaya produksi lebih rendah dibandingkan dengan menggunakan EEG dan kamera infra merah.

Deteksi pergerakan bola mata pada prinsipnya diawali dengan mendeteksi area wajah terlebih dahulu, hal ini penting dilakukan agar area seleksi yang harus diolah oleh komputer menjadi lebih kecil dan terfokus. Setelah area pada wajah dapat dideteksi, maka langkah selanjutnya adalah dengan mendeteksi area mata, baru setelah area mata terdeteksi maka program dapat fokus untuk mencari lingkaran pupil mata dan menentukan titik tengahnya. Pada prinsipnya pendeteksian area wajah, mata dan pupil mata telah banyak disepakati dan memiliki metode-metode yang sering dan akurat untuk digunakan. Metode yang sering dan dapat digunakan untuk mendeteksi area wajah dan mata adalah dengan menggunakan *Haar Cascade Classifier. Haar Cascade Classifier* sendiri adalah sebuah metode yang terbukti handal untuk membedakan antara objek yang ingin dikenali dan objek yang tidak ingin dikenali (Viola & Jones, 2004). Haar memiliki fitur warna hitam dan putih untuk mendeteksi objek secara terperinci, Haar sendiri telah banyak dimanfaatkan dan diterapkan untuk pengendali pergerakan suatu perangkat navigasi dengan menggunaan gerakan mata (Patel & Prakash, 2015). Haar juga dapat digunakan untuk mendeteksi berbagai objek lainnya seperti plat nomor kendaraan (Amin & Wahono, 2015).

Berbeda dengan pendeteksian area wajah dan mata yang telah banyak disepakati dengan menggunakan Haar dan metode lainnya yang sudah terbukti akurat, metode untuk mendeteksi gerakan pada mata masih dalam tahap pencarian dan belum memiliki suatu metode yang paling akurat dan diterapkan oleh banyak peneliti. Banyak sekali cara yang digunakan oleh para peneliti untuk mendeteksi gerakan bola mata seperti penelitian tentang deteksi pergerakan bola mata untuk menggerakan kursi roda pintar (Utaminingrum et al., n.d.), namun deteksi hanya bisa dilakukan untuk arah kanan, tengah dan kiri saja. Penelitian juga dilakukan dengan memanfaatkan sudut yang terbentuk antara bola mata, hidung dan suatu titik pada kotak imajiner area deteksi wajah (Prasetya & Utaminingrum, 2017), namun dengan menggunakan metode *Triangle Similarity* ini keakuratan sistem untuk mendeteksi gerakan mata ke arah bawah masih kurang maksimal.

Seperti deteksi gerakan bola mata, deteksi kedipan mata juga masih belum memiliki sebuah metode yang umum digunakan seperti pendeteksian area wajah dan mata. Banyak peneliti yang menawarkan beberapa solusinya untuk mendeteksi adanya kedipan mata, seperti penggunaan mekanisme proyeksi atau penjumlahan nilai pixel pada suatu baris atau kolom untuk mendapatkan nilai vektor yang bisa digunakan untuk membedakan antara mata terbuka dan mata tertutup (Haq & Hasan, 2017) dan ada juga yang menggunakan *Face Landmark* untuk mendeteksi adanya kedipan dengan menggunakan persamaan *Eye Aspect Ratio (EAR)* yang memiliki tingkat akurasi dan kecepatan yang cukup baik.

Dari beberapa penelitian yang ada, di penelitian ini diusulkan penyempurnaan metode deteksi pergerakan bola mata dengan menggabungkan antara metode *Naïve Bayes* dan *Pixel Value* berbasis sector dengan beberapa perubahan dan tambahan pada pengolahan citra-nya untuk dikombinasikan dengan deteksi kedipan mata menggunakan EAR berbasis *Face Landmark* yang diterapkan secara real time sehingga dapat dihasilkan sebuah skema baru untuk mendeteksi gerakan bola mata dan kedipan mata yang dapat diaplikasikan sebagai pengganti kontrol pointer konvensional yang menggunakan tangan, dimana mekanisme kontrol tersebut dapat diterapkan pada perangkat *digital* atau komputer. Pada penelitian sebelumnya telah dilakukan penggunaan gabungan metode yang sama untuk mendeteksi arah gerakan bola mata namun terbatas hanya pada *still image* atau Gambar diam berekstensi JPEG saja. Maka dari itu pada penelitian ini, untuk membuktikan mekanisme penggabungan metode tersebut dan digunakan secara real time, maka pada penelitian ini diterapkan untuk memilih menu-menu yang ada pada sebuah layar LCD yang terpasang di sebuah *Smart Wheelchair.* Pemilihan metod *Naïve Bayes* di penelitian ini dikarenakan kemampuan *Naïve Bayes* yang tidak terpengaruh terhadap banyaknya data yang digunakan dalam proses pelatihan, dikarenakan penelitian ini bertujuan untuk menerapkan metode usulan pada sebuah sistem real time maka diperlukan adanya kalibrasi antara sistem dengan user pengguna. Kalibrasi

dilakukan untuk mendapatkan kondisi mata pengguna atau user dalam keadaan normal / idle, melihat ke atas, bawah, kanan dan kiri. Dengan menggunakan data hasil kalibrasi lalu dilakukan proses pelatihan dengan menggnakan *Naïve Bayes* sebagai metode pelatihannya. Data kalibrasi yang didapat tentu tidak memiliki jumlah yang banyak karena kalibrasi hanya dilakukan secukupnya saja, maka dari itu, sedikitnya jumlah data latih hasil dari kalibrasi ini menjadi alasan yang cukup kuat mengapa *Naïve Bayes* digunakan alam penelitian ini dikarenakan kemampuannya yang tidak terpengaruh oleh banyaknya data. Selain adanya kalibrasi posisi mata, penentuan kondisi pencahayaan juga menjadi pengaruh dalam penelitian ini, kondisi pencahayaan yang pas akan menghasilkan hasil yang maksimal. Penentuan kondisi pencahayaan dilakukan dengan menggunakan parameter *lux* yang diujikan pada beberapa kondisi seperti gelap (100 lux), cukup terang (300 lux) dan sangat terang (700 lux). Dengan memanfaatkan parameter lux tersebut maka akan ditentukan pada kondisi mana sistem yang dibangun ini dapat berjalan dengan menghasilkan hasil yang maksimal. Adanya kondisi cahaya tersebut menuntut adanya nilai *threshold* yang digunakan agar sistem mampu untuk mendeteksi arah gerakan mata dengan akurasi yang baik sehingga perlu dilakukannya beberapa percobaan untuk mewujudkan hal tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang dapat dijabarkan adalah sebagai berikut :

1. Bagaimana hasil akurasi deteksi mata tanpa kacamata dan mata dengan kacamata dengan pengaruh jarak sebesar 10, 20, 30, 40, 50 dan 60 cm dari kamera menuju mata dan dengan kondisi cahaya stabil ?
2. Berapa nilai parameter jarak (cm), pencahayaan (lux) dan nilai *threshold* untuk mendapatkan nilai akurasi yang baik dalam mendeteksi pergerakan bola mata ?
3. Bagaimana hasil akurasi deteksi gerakan bola mata terhadap arah gerakan ke kiri, kanan, atas, bawah dan kedipan (mata tertutup) dengan menggunakan parameter terbaik yang telah didapatkan ?
4. Bagaimana hasil perhitungan *Precision & Recall* untuk mendeteksi gerakan ke Atas, Bawah, Kanan, Kiri, Tengah dan Kedipan ?
5. Bagaimana hasil performa akurasi metode usulan jika dibandingkan dengan metode-metode lain ?
6. Bagaimana hasil akurasi sistem dalam mengintegrasikan antara deteksi gerakan bola mata dan deteksi gerakan menutup mata (kedipan) yang diterapkan pada sebuah layar yang berisi menu pilihan mekanisme kontrol kursi roda ?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Menerapkan metode *Face Landmark, Eye Sector Area* dan *Naïve Bayes* untuk mendeteksi pergerakan bola mata dan kedipan secara *real time* dan bersamaan ?
2. Mengetahui pengaruh kondisi lingkungan (cahaya dan jarak) terhadap kemampuan sistem untuk mendeteksi bola mata secara tepat
3. Mengetahui besaran nilai parameter yang digunakan oleh sistem agar dapat mendeteksi arah gerakan mata dan kedipan dengan tepat dan akurat
4. Mengetahui tingkat akurasi yang dicapai dalam mendeteksi pergerakan bola mata terhadap arah gerakan ke kiri, kanan, atas dan bawah pada dengan parameter yang telah ditentukan
5. Mengetahui tingkat akurasi yang diperoleh terhadap deteksi gerakan menutup mata (kedipan) dengan parameter-parameter yang ditentukan.
6. Mengetahui mekanisme integrasi dan akurasi sistem antara mendeteksi gerakan bola mata dan gerakan menutup mata serta implementasinya berupa menggerakan sebuah GUI dengan tampilan yang interaktif

1.4 Manfaat Penelitian

Manfaat dalam penelitian ini adalah diharapkan sistem yang dibangun memiliki sebuah solusi aternatif bagi para penyandang disabilitas untuk menggunakan mata mereka sebagai pengganti alat kontrol atau pointer yang dapat digunakan dan diaplikasikan ke berbagai media elektronik, komputer ataupun alat bantu navigasi seperti kursi roda pintar dan sebagainya. Diharapkan, dengan menggunakan deteksi pergerakan bola mata dan gerakan menutup mata (kedipan), biaya produksi sebuah alat bantu tidak melonjak tinggi dikarenakan hanya membutuhkan sebuah kamera digital yang terpasang untuk menangkap gerakan pada mata.

1.5 Batasan Masalah

1. Pada penelitian ini tidak akan dibahas mengenai pengujian usability
2. Kamera yang digunakan adalah kamera digital dengan resolusi 1280 x 720
3. Pencahayaan diujikan dengan kondisi cahaya redup (100 lux), Sedang (300 lux) dan sangat terang (700 lux).
4. Pencahayaan dilakukan secara statis dengan cara memanfaatkan cahaya LED yang terpasang dibelakang kamera untuk menjaga cahaya yang ada tetap stabil
5. Bahasa pemrograman menggunakan Python dengan versi 2.7
6. Jarak pengujian yang dilakukan dimulai dari 10 cm hingga 60 cm dari kamera, nantinya akan diketahui jarak yang paling ideal untuk digunakan oleh sistem ini

BAB 2
LANDASAN TEORI

2.1 Populasi disabilitas di Indonesia

Buletin Data dan Informasi Kesehatan oleh Kemenkes RI menyebutkan bahwa pengertian Disabilitas adalah suatu kelompok masyarakat yang memiliki keterbatasan yang dapat menghambat partisipasi dan peran serta mereka dalam kehidupan sehari-hari dan kehidupan bermasyarakat. Konvensi yang menyangkut tentang Hak-hak penyandang Disabilitas yang telah ikut ditandatangani oleh Indonesia mengamanahkan Negara untuk mengambil kebijakan yang diperlukan untuk menjamin akses bagi para penyandang disabilitas terhadap pelayanan kesehatan (Kementrian Kesehatan RI, 2014). Di Indonesia, terdapat sekitar 21.930.529 jiwa penyandang disabilitas, dimana angka itu merepresentasikan bahwa 8% penduduk di Indonesia memiliki kekurangan atau menyandang disabilitas (Kemenkes RI, 2017). Penyandang disabilitas itu memiliki berbagai jenis kekurangan dan disfungsionalitas pada anggota tubuhnya dikarenakan cacat yang mereka sandang. Dalam kehidupan sehari-hari aktivitas mobilitas tentu menjadi hal yang wajib dan lumrah untuk dilakukan seseorang, namun hal tersebut cukup sulit untuk dilakukan oleh para penyandang disabilitas dikarenakan anggota tubuh yang menjadi aktor utama untuk bermobilitas tidak dapat berfungsi dengan sempurna seperti cacat pada kaki maupun juga cacat pada tangan. Memang telah ada salah satu solusi yang dapat menjadi alternative bagi penyandang disabilitas seperti penggunaan kursi roda dengan menggunakan tangan sebagai penggerak roda. Namun, kesulitan datang ketika penyandang disabilitas memiliki lebih dari satu kekurangan atau biasa disebut dengan disabilitas ganda seperti mereka yang memiliki kecacatan pada tangan dan kaki sekaligus sehingga mereka akan susah untuk mengoperasikan kursi roda yang dioperasikan dengan menggunakan tangan. Padahal, jumlah orang dengan penyandang disabilitas lebih dari satu atau seseorang dengan disabilitas tidak dapat mengoperasikan tangan ataupun jari mereka tergolong cukup banyak seperti ditampikan pada Tabel 2.1.

Pemenuhan hak-hak untuk para disabilitas sendiri harus dipenuhi sesuai dengan UU. RI No. 8 Tahun 2016 tentang kesehatan BAB VII Bagian ke-tiga pasal 139, bahwa pemerintah memiliki kewajiban untuk upaya pemeliharaan penyandang disabilitas di Indonesia. Maka dari itu, pentingnya fasilitas agar para penyandang disabilitas dapat melakukan kegiatan sehari-hari diharapkan mampu mengurangi beban yang dimiliki oleh para penyandang disabilitas.

Table 2.1 Disabilitas dengan kesulitan menggerakan tangan dan kaki

Jenis Disabilitas	Jumlah Total
Kesulitan berjalan	
Sepenuhnya butuh bantuan orang lain	672.757
Memakai alat bantu dan bantuan orang	329.659
Memakai alat bantu	1.092.448
Tidak memakai alat bantu	5.713.513
Kesulitan menggunakan atau menggerakan tangan / jari	
Sama sekali tidak bias	129.959
Banyak	526.048
Sedikit	2.062.283

Sumber : BPS (2015)

2.2 Penelitian Terkait

Terdapat beberapa penelitian yang telah dilakukan sebelumnya dengan memanfaatkan bagian anggota tubuh serta mata sebagai media kontrol dan penggerak sebuah alat elektronik (device).

1. Menggerakan pointer komputer dengan gestur tangan (Manchanda & Bing, 2010)

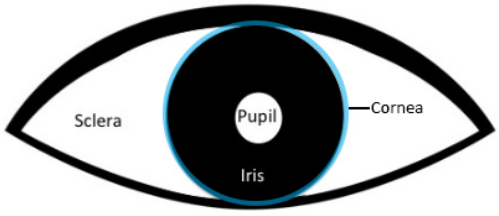
   Dalam penelitian ini, pointer pada komputer dikontrol dengan menggunakan gerakan gestur tangan. Alat yang digunakan sendiri adalah dengan menggunakan kamera. Metode yang digunakan adalah dengan mendeteksi warna kullit pengguna. Setelah warna kulit dapat terdeteksi maka dilakukan proses berikutnya yaitu Erosi dan Dilasi sehingga Gambar area tangan dapat terdeteksi dengan jelas. Logika komputer dalam mengetahui gerak gestur tangan adalah dengan menggunakan *Global Motion Vector.* Metode *Global Motion Vector* sendiri memanfaatkan jumlah perpindahan objek berdsarkan sumbu x dan sumbu y yang kemudian dirata-rata, sehingga dapat diketahui arah dan banyaknya jarak perubahan gestur tangan. Dengan menggunakan metode ini, gerakan tangan masih tetap bisa terbaca walaupun user berada pada jarak yang cukup jauh, maksimal jarak yang bisa dideteksi adalah 3 meter dari kamera. Dengan menggunakan gestur tangan ini komputer dapat terkontrol dengan cukup baik. Namun, metode ini memiliki kelemahan yaitu harus menggunakan tangan sebagai media kontrol, sehingga tidak cocok digunakan bagi para penyandang disabilitas ganda yang memiliki cacat pada tangan dan kakinya.

2. Menggunakan gerakan kaki untuk menggantikan mouse konvensional (Ortiz, Arciaga, Escusa, & Ubales, 2017)
Penelitian ini mengusung ide untuk menggunakan kaki sebagai pengganti fungsi mouse. Dengan menggunakan alat bernama sensor pad maka dapat dideteksi adanya tekanan pada suatu bagian tertentu yang menandakan adanya perintah yang diberikan. Walaupun ide yang diusulkan cukup unik namun metode ini masih kurang pas digunakan bagi penyandang disabilitas, terlebih para disabilitas tentunya kurang mampu untuk menggerakan kaki ataupun tangan mereka dengan baik. Metode ini memiliki tingkat keakurasian sebanyak 73.375%.

3. Menggunakan gerakan mata sebagai alat penggerak pointer (Kim, Kang, Jung, & Kim, 2005)
Pada penelitian ini, mata digunakan sebagai alternative dalam mengontrol gerakan *pointer* atau *mouse*. Dalam penelitian ini digunakan metode MLP (*Multi Layer Perceptron*) untuk mendeteksi bola mata, dimana sebelumnya area wajah dideteksi terlebih dahulu dengan memanfaatkan warna kulit. Setelah area wajah terdeteksi maka langkah selanjutnya adalah mendeteksi area mata dengan menggunakan MLP dengan mekanisme perbandingan maka akan didapat area yang masuk wilayah mata dan area yang bukan termasuk mata. Mekanisme *tracking* gerakan bola mata sendiri adalah dengan menggunakan nilai koordinat bola mata yang dicatat pada setiap framenya. Dari hasi koordinat tersebut dapat diperoleh nilai perubahan letak posisi mata yang dikorelasikan dengan perubahan letak posisi pointer. Untuk mode *click* pada penelitian ini menggunakan sebuah kondisi dimana jika tidak ada gerakan atau aktifitas pada suatu titik koordinat maka dianggap *user* telah memilih titik itu untuk di *click* atau dipilih. Pendekatan yang diusulnya oleh penelitian ini cukup menarik namun masalah ada pada metode untuk pemilihan menu atau *click*, dimana ketika tidak ada aktifitas dalam jangka waktu tertentu maka *user* dianggap memilih menu tersebut. Hal tersebut menjadi kurang pas atau cocok ketika diterapkan pada orang awam yang baru pertama kali menggunakan alat tersebut.

2.3 Mata

Mata adalah salah satu komponen wajah yang sangat menonjol, melalui mata pula dapat diketahui ekspresi yang dikeluarkan seseorang seperti marah, senang, sedih, bahagia ataupun yang lainnya. Manusia dapat dengan mudah menebak sebuah emosi yang timbul pada lawan bicaranya hanya dengan memperhatikan mata saja (Robson, 1967). Secara global terdapat beberaa bagian-bagian pada mata yaitu pupil, kornea, sclera, kelopak mata dan lainnya (Iveson-Iveson, 1983). Gambar bagian-bagian mata ditunjukan pada Gambar 2.1.



Gambar 2.1 Anatomi mata sederhana

Sumber : Utaminingrum (2016)

Pada Gambar 2.1 terlihat bagian pupil berada di tengah lingkaran hitam atau iris. Penelitian ini akan fokus untuk mendeteksi letak pupil berada dengan menggunakan sebuah kamera. Pada penelitian ini pupil dapat diartikan menjadi titik tengah bola mata, dengan begitu kemanapun bola mata bergerak maka program dapat mendeteksinya dengan cara melacak pergerakan pupil atau titik tengah ini. Selain pada pupil, penelitian ini juga memanfaatkan kelopak mata atau garis melengkung bagian atas dan bawah mata untuk mendeteksi adanya aktifitas buka tutup mata atau kedipan. Fungsi dari pendeteksian kedipan ini adalah untuk menggantikan proses *click* pada menu layar navigasi yang digunakan untuk eksekusi pemilihan menu

2.3.1. Penggunaan mata dalam teknologi

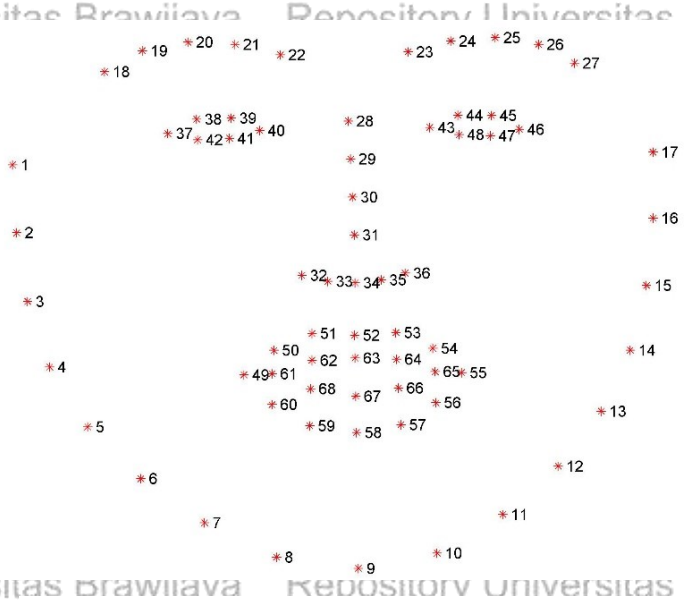
Mata sebagai sebuah organ yang paling penting bagi manusia memiliki banyak peran dalam dunia keilmuan dan teknologi. Banyak dari para peneliti menjadikan mata sebagai objek penelitian mereka dikarenakan sifat mata yang vital juga letak posisi mata yang mudah untuk diakses, selain itu gerakan pada mata juga fleksibel dan mudah bagi kebanyakan orang. Ada dua jenis penelitian yang menggunakan mata sebagai objek yaitu :

a. Penggunaan mata untuk teknologi yang bersifat interaktif
   Mata dapat dimanfaatkan kegunaannya untuk media pengganti alat kontrol seperti tangan. Dalam hal ini mata dapat digunakan untuk media pengganti channel televisi, ataupun sebagai media penggerak media lain.
b. Penggunaan mata untuk teknologi yang bersifat diagnosis
   Dalam hal ini mata digunakan sebagai media analisis atensi pengguna misalnya, biasa diterapkan dalam dunia pendidikan ataupun dunia *e-commerce*. Mata juga dapat dimanfaatkan sebagai *assistant driver* untuk mendeteksi kantuk dan kelelahan pada seorang driver mobil, ataupun mata dapat digunakan juga dalam bidang kesehatan seperti mendeteksi adanya gejala *vertigo.*

2.3.1.1. Face Landmark Dengan Menggunakan Pohon Regresi

Didalam berbagai penelitian yang berfokus pada *face landmark* atau pemetaan wajah, banyak dijumpai berbagai metode untuk memetakan titik anatomi wajah atau *face landmark*. Salah satu yang paling popular dan paling banyak digunakan adalah dengan menggunakan Pohon Regresi (Kazemi & Sullivan, 2017). Penggunaan pohon regresi pada *face landmark* menunjukan hasil yang cukup memuaskan dan akurasi yang tinggi. Prinsip kerja dari *face landmark* sendiri adalah memanfaatkan output dari algoritma pengenalan daerah wajah seperti *Haar Cascade*, setelah output didapat maka *Face Landmark* akan memetakan titik-titik anatomi yang ada pada wajah dengan bantuan data latih yang telah dilabeli secara manual.

Pada prinsipnya, didalam pohon regresi pasti akan terjadi split atau percabangan, maka dari itu ketika menemukan percabangan maka prediksi akan didasarkan pada thresholding nilai intensitas. Thresholding nilai intensitas digunakan karena kemampuannya dalam beradaptasi dengan cahaya, maka dari itu selama cahaya tidak berubah secara total maka masih bisa untuk di tolerir. *Exponensial prior* juga digunakan sebagai bantuan dalam menentukan cabang mana yang akan dipilih, dikarenakan hasil kembalian dari penggunaan threholding adalah jumlah kemungkinan percabangan itu sendiri. Pada metode ini, jika terjadi kemungkinan tidak tercatatnya label pada data latih maka dapat digunakannya



Gambar 2.2 visualisasi 68 titik pada wajah dengan menggunakan *facial landmark* pohon regresi

Gradient Boosting dengan cara mencocokan pohon regresi dan bobot rata-rata pada target. Ilustrasi hasil dari *face landmark* ditunjukan pada Gambar 2.2.

2.4 Deteksi area mata

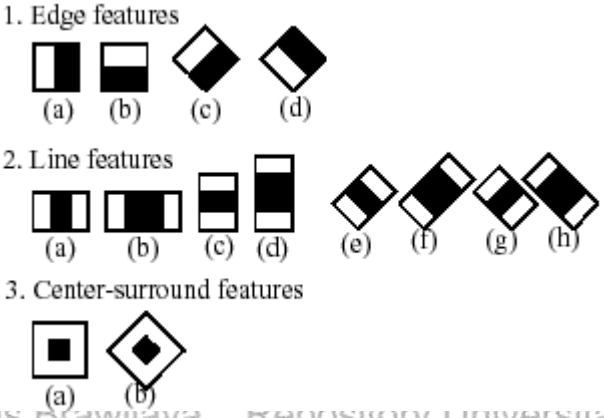
Pada proses pendeteksian area mata, terbagi menjadi dua bagian. Bagian pertama adalah untuk mendeteksi pupil mata dan bagian kedua adalah untuk mendeteksi letak kelopak mata.

2.4.1. Deteksi bola mata

Tujuan dari pendeteksian bola mata ini adalah untuk dapat mengetahui arah gerakan yang dilakukan oleh mata. Pada proses deteksi ini, program akan fokus untuk mencari letak dari pupil mata yang sekaligus menjadi titik tengah dari bola mata. Pada proses pendeteksian bola mata ini akan dibagi menjadi beberapa bagian mulai dari deteksi wajah sampai dengan deteksi titik tengah bola mata. Pada ranah pendeteksian area mata telah banyak dikembangkan metode-metode untuk mendeteksinya, salah satu yang paling umum dan paling banyak digunakan adalah metode *Haar Cascade Classifier,* dimana metode ini terbukti memiliki tingkat akurasi yang cukup tinggi dan memang telah banyak digunakan oleh para peneliti sebelumnya. Pada sub bab ini akan dijelaskan bagaimana metode *Haar Cascade* dapat mendeteksi area mata dengan berbagai metode pendukung yang dipakai pada penelitian ini.

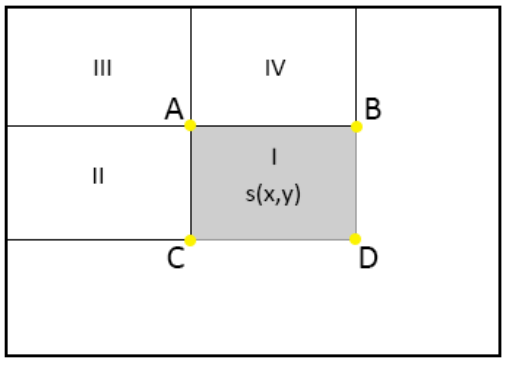
2.4.1.1. Haar Cascade Classifier

Metode *Haar Cascade* adalah sebuah metode pengenalan objek yang handal dengan akurasi yang cukup tinggi dan dapat diandalkan, dikarenakan metode ini menggabungkan metode supervised dan unsupervised learning secara bersamaan(Viola & Jones, 2001). Ciri khas dari metode *Haar Cascade* ini adalah memiliki data training yang cukup banyak, sehingga metode ini membutuhkan waktu yang cukup lama untuk proses pelatihan data (Gupta & Sharma, 2014). Namun, sejalan dengan waktu pelatihannya, hasil akurasi yang dihasilkan oleh metode ini cukup tinggi. Ada beberapa penelitian yang menggunakan metode *Haar Cascade* dalam karyanya, diantaranya adalah sebuah sistem yang dapat menrekognisi kuda secara real time (Uddin & Akhi, 2016). Algoritma *Haar Cascade* menggunakan template fitur hitam putih yang dimiliki oleh Haar dalam pengaplikasiannya, adapun template hitam putih tersebut ditunjukan pada Gambar 2.3.



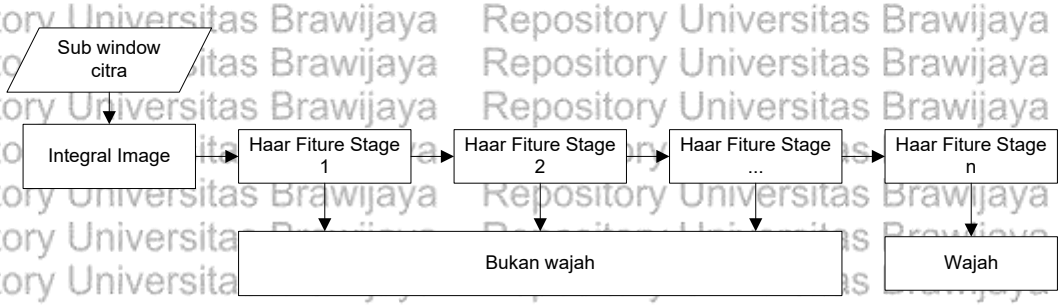
Gambar 2.3 Beberapa macam template fitur yang dimiliki Haar

Perhitungan nilai fitur Haar menggunakan sebuah konsep penghitungan nilai pixel yang bernama *Integral Image*. Fungsi dan kegunaan dari *Integral Image* sendiri adalah untuk menyingkat waktu proses yang dilakukan oleh Haar dengan cara mengelompokan pixel-pixel kecil menjadi satu nilai kesatuan yang merepresentasi nilai-nilai pixel disekitarnya. Skema perhitungan *Haar* ditunjukan pada Gambar 2.4.



Gambar 2.4 skema mekanisme perhitungan Inegral Image

Pada Gambar 2.3 ditunjukan bahwa nilai yang ada pada pixel I adalah representasi dari ketiga pixel disekitarnya (bagian kiri dan atas saja) yaitu (I,III, II, dan IV). Proses selanjutnya adalah mencocokan nilai yang ada pada kotak pixel dengan nilai ambang yang dimiliki oleh Haar. Apabila nilai yang ada pada kotak tidak memenuhi nilai ambang yang menjadi batasan dari Haar maka disimpulkan bahwa objek atau Gambar tidak dapat dikenali. Namun apabila nilai dari kotak pixel cocok dengan nilai ambang yang dimiliki oleh Haar maka proses akan dilanjutkan ke tingkatan berikutnya hingga tingkat yang paling akhir sampai pada ahirnya dapat ditentukan bahwa terdapat objek yang dikenali oleh *Haar* dalam Gambar tersebut (*Cascade*). Dalam penelitian ini digunakan sebanyak 5 Stage dengan masing-masing stage menggunakan 50 Fitur Haar yang sudah ada pada *Haar Cascade* library dengan data latih lebih dari 6000 data yang berupa xml. Adapun ilustrasi tentang mekanisme stage ditunjukan pada Gambar 2.5.



Gambar 2.5. Mekanisme *Cascade* Haar untuk pencarian area wajah

Apabila area wajah telah ditemukan dengan menggunakan mekanisme Haar Cascade ini, maka langkah selanjutnya adalah memetakan area wajah tersebut untuk difokuskan pada area mata saja, yang akan dijelaskan pada subbab berikutnya.

2.4.2. Face Landmark

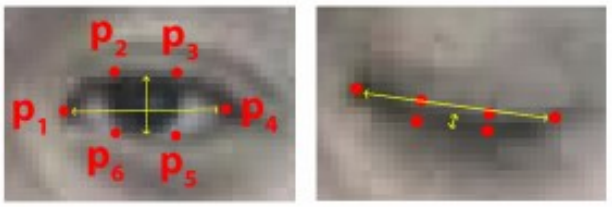
Face Landmark adalah sebuah proses memetakan ruang-ruang atau titik-titik yang ada pada wajah. Pemetaan titik-titik tersebut dapat dilakukan dengan memanfaatkan landmark. Cukup banyak penelitian yang mengangkat tema tentang *face landmark* ini, dikarenakan *Face Landmark* sendiri cukup berperan penting didalam dunia komputer visi terlebih sejak era tahun 1990 (Jiang, Zhang, Deng, Guo, & Liu, 2017).

2.4.2.1. EAR berdasarkan Face Landmark

EAR atau *Eye Aspect Ratio* adalah sebuah nilai yang didapatkan dari proses perhitungan memanfaatkan titik-titik area mata yang didapat menggunakan proses *Face landmark*. Persamaan EAR ditunjukan pada Persamaan 2.1.

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||} \quad (2.1)$$

Titik-titik $P_1 - P_6$ didapat dari hasil proses Face Landmark, adapun titik-titik tersebut ditunjukan pada Gambar 2.6.



Gambar 2.6. Titik koordinat area mata

2.5. Pengolahan Citra

Pengolahan citra digita merupakan sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra secara digital yang dipresentasikan secara

numeric dengan tujuan memperbaiki kualitas citra, dan mengekstrasi informasi ciri atau fitur pada sebuah citra.

2.5.1. Morfologi Citra

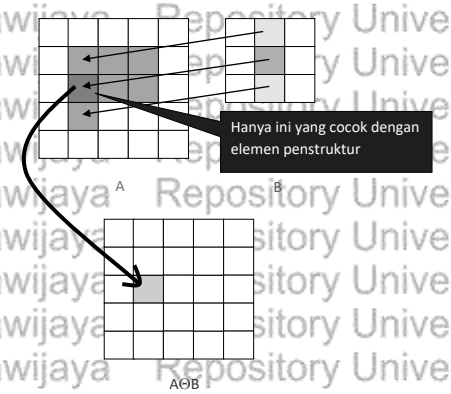
Proses morfologi memiliki tujuan untuk mengubah struktur bentuk objek pada citra. Objek bisa dirubah menjadi lebih besar, kecil, tipis atau bahkan tebal sesuai dengan operasi yang dilakukan. Banyak hasil yang dapat diperoleh dengan menggunakan operasi morfologi, diantaranya adalah untuk memperoleh filter spasial, memperoleh kerangka objek, menentukan letak objek didalam sebuah citra sampai dengan memperoleh bentuk dari suatu objek (Kadir & Susanto, 2013). Inti dari operasi morfologi sendiri adalah melibatkan dua larik pixel, larik pertama adalah citra yang akan dilakukan operasi morfologi dan larik kedua adalah kernel (Shih, 2017). Pada penelitian ini, akan melibatkan beberapa operasi morfologi yaitu *opening* dan juga *closing* yang fungsinya nanti adalah untuk pre-proses citra sehingga memiliki nilai yang jelas dan dapat diolah dengan baik juga meminimalisasi terjadinya error.

Operasi *Opening* sendiri adalah sebuah operasi citra yang dilakukan dengan melibatkan dua operasi sederhana yaitu operasi *erosi* yang kemudian diikuti oleh operasi *dilasi* dengan menggunakan elemen dan atribut nilai yang sama persis. Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan seluruh *pixel* di area yang terlalu kecil untuk ditempati oleh elemen penstruktur. Dengan kata lain, semua struktur latardepan yang berukuran lebih kecil daripada elemen penstruktur akan tereleminasi oleh *erosi* dan kemudian dihaluskan dengan menggunakan *dilasi*. Berbanding terbalik dengan operasi *Opening*, maka operasi *Closing* adalah menerapkan proses *dilasi* kemudian baru diikuti dengan operasi *erosi*. Tujuan dari operasi *Closing* ini adalah untuk menghaluskan kontur seperti menghilangkan lubang-lubang kecil pada citra.

Didalam proses operasi *Opening* maupun *Closing* pasti melibatkan operasi citra lain yaitu operasi *erosi* dan *dilasi*. *Erosi* sendiri adalah sebuah operasi yang memiliki tujuan untuk memperkecil ukuran citra. Operasi ini dirumuskan pada Persamaan 2.2.

$$A \ominus B = \{p \in Z^2 \mid (a + b) \in I, \ untuk\ setiap\ b \in B\} \quad (2.2)$$

Dimana posisi p terdapat pada $A \ominus B$ jika seluruh nilai 1 di B terkandung di posisi p tersebut. Illustrasi dari proses erosi sendiri ditunjukan pada Gambar 2.7.



Gambar 2.7 Visualisasi proses Erosi

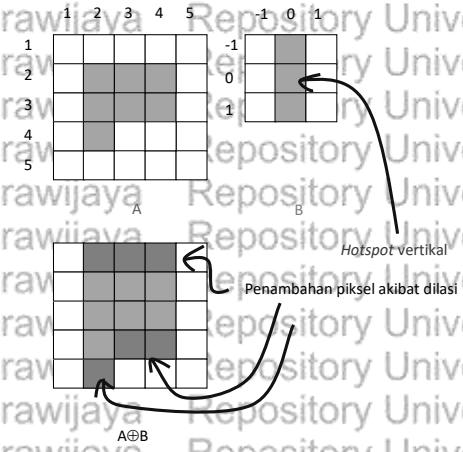
Jika proses *Erosi* menghasilkan hasil akhir berupa Gambar dengan objek yang lebih kecil, maka sebaliknya porses dilasi akan menghasilkan hasil keluaran berupa Gambar yang diperbesar ukurannya. Persamaan matematika untuk proses *dilasi* ditunjukan pada Persamaan 2.3.

$$A \oplus B = \left\{ z \middle| \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A \right\} \quad (2.3)$$

Dalam hal ini,

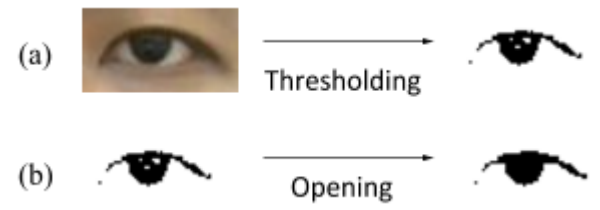
a) $\hat{B} = \{w | w = -b, untuk\ b \in B\}$
b) $(B)_z = \{c | c = a + z, untuk\ a \in A\}$
c) z=(z1, z2)

Hasil dari proses *dilasi* sendiri adalah berupa penjumlahan seluruh pasangan koordinat. Ilustrasi proses dilasi ditunjukan pada Gambar 2.8.



Gambar 2.8 Ilustrasi dilasi dengan kernel vertikal

Operasi *dilasi* dan *erosi* pada penelitian ini diterapkan pada sesi pendeteksian bola mata, dengan tujuan agar program mampu mendeteksi bola mata secara tepat dan akurat. Maka dari itu perlu dilakukannya proses pengolahan citra sehingga memiliki nilai yang jelas, dengan begitu maka akan mengurangi kemungkinan terjadinya kesalahan deteksi untuk letak bola mata. Prsoes *opening* dipilih untuk memperbaiki citra bola mata yang agak sedikit rusak ketika melalui proses segmentasi warna. Rusak disini diakibatkan proses segmentasi hanya mengandalkan ambang nilai pixel saja, sehingga bagian mata yang warnanya lebih muda dikarenakan faktor pantulan cahaya misalnya akan dianggap sebagai warna putih dan akan dihilangkan. Dari sini, proses opening yang berguna untuk memperjelas objek perlu dilakukan untuk menutupi lubang-lubang *pixel* hasil dari proses sebelumnya yaitu proses *tresholding*. Ilustrasi proses *opening* pada citra mata ditampilkan pada Gambar 2.9.



Gambar 2.9 Ilustrasi hasil proses opening yang dilakukan pada citra area mata

2.5.2. Konversi Citra Berwarna

Dalam pemrosesan citra digital, proses awal yang sering dilakukan adalah merubah citra berwarna atau citra *Red*, *Green*, *Blue* (RGB) menjadi citra yang memiliki derajat keabuan atau disebut citra *grayscale*. Dalam proses deteksi mata, metode *Haar Cascade* juga menggunakan citra *grayscale*. Representasi nilai *grayscale* dari nilai RGB pada suatu citra, dapat dihitung menggunakan Persamaan 2.4.

$$I = \frac{R+G+B}{3} \quad (2.4)$$

Pada Persamaan 2.4 dapat diketahui jika *I* adalah citra berskala keabuan, *R* adalah nilai merah pada sebuah pixel, *G* adalah nilai hijau pada satu pixel dan *B* adalah nilai biru pada satu pixel. Perhitungan rumus tersebut dilakukan sebanyak jumlah *n* pixel.

Dari proses grayscale, bisa dijabarkan lebih lanjut menuju porses *Biner,* dimana citra biner adalah sebuah citra yang hanya memiliki 2 nilai warna saja yaitu 1 dan 0 atau hitam dan putih saja. Pada proses biner kita akan mengenal suatu nilai yang dikenal dengan nilai ambang (*treshhold).* nilai tersebut digunakan untuk

menentukan suatu batas dimana pixel akan dirubah menjadi nilai 1 atau dirubah menjadi nilai 0. Secara matematis, konversi dinyatakan dengan Persamaan 2.5.

$$b(i) = \begin{cases} 0, & i \geq a \\ 1, & i < a \end{cases} \quad (2.5)$$

Dimana, $b(i)$ adalah pixel hasil binerisasi, i adalah nilai pixel saat ini dan a adalah nilai batas ambang atau nilai *threshold*.

2.5.3. Resize

Resize merupakan proses merubah ukuran citra digital melalui sebuah operasi geometri yang memberikan pengaruh untuk memperbesar (*zoom-in*) atau memperkecil (*zoom-out*) ukuran citra input berupa *width* dan *height* sesuai dengan nilai variabel penskalaan yang ditentukan. Persamaan yang digunakan dalam proses *resize* adalah :

$$w' = SW * w \quad (2.6)$$

$$h' = SH * h \quad (2.7)$$

Keterangan :

SW = faktor skala lebar

SH = faktor skala tinggi

w = lebar awal

h = tinggi awal

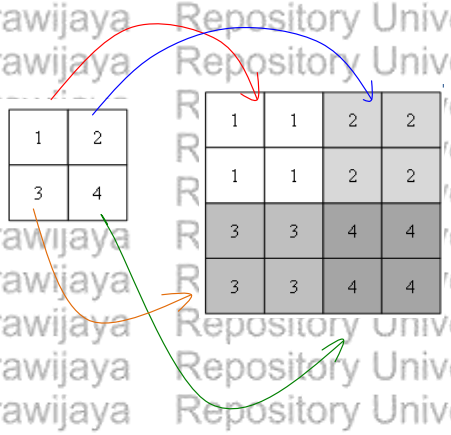
w' = lebar hasil skala

h' = tinggi hasil skala

dengan *w'* dan *h'* maka dapat diketahui citra baru hasil resize menggunakan persamaan 2.8.

$$resize_{(x_1,y_1)} = \sum_{i=1}^{h'} y_1 = (y-1)/SH \sum_{j=1}^{w'} x_1 = (x-1)/SW \quad (2.8)$$

ilustrasi perbesaran citra dengan menggunakan pendekatan *replication* sendiri ditunjukan dengan Gambar 2.10 berikut.



Gambar 2.10. Ilustrasi pembesaran citra / resize

2.6. *Naïve Bayes*

Penggolongan *Naïve Bayes* adalah keluarga penggolong probabilistik sederhana berdasarkan penerapan teorema Bayes. Penelitian ini menerapkan metode *Naïve Bayes* untuk melatih dan mendeteksi arah gerakan bola mata. Fitur *Naïve Bayes* disini menggunakan nilai EAR dan jumlah pixel untuk mengklasifikasikan. Langkah pertama *Naïve Bayes* adalah menentukan nilai rata-rata dari data yang ditunjukkan dalam Persamaan (2.9), ada sekitar 50 data Gambar yang akan diklasifikasikan menggunakan *Naïve Bayes*. Setiap data Gambar akan menentukan nilai EAR dan *Corner* dengan menggunakan beberapa metode seperti yang disebutkan sebelumnya.

$$\mu = \frac{1}{N} . \sum_{i=1}^{N} x_i \quad (2.9)$$

Misalnya, *N* adalah total data dan x_i adalah nilai dari data. Dengan menggunakan nilai μ, kita dapat menentukan varians sampel yang tidak bias (δ^2) seperti yang ditunjukkan dalam Persamaan (2.10).

$$\delta^2 = \frac{1}{N-1} . \sum_{i=1}^{N} (x_i - \mu)^2 \quad (2.10)$$

Dengan menggunakan μ dan δ^2, mean dan varians dari data pelatihan diperoleh. Ada 4 kelas yang diklasifikasikan oleh *Naïve Bayes* yang menghasilkan total 16 persamaan termasuk sarana dan varian di setiap kelas.

μ dan δ^2 diperlukan untuk menentukan kepadatan probabilitas. Kepadatan probabilitas dapat diukur menggunakan persamaan (2.11) dengan mengganti nilai μ dan δ^2.

$$f(x;\mu,\delta^2) = \frac{1}{\sqrt{2\,.\,\pi\,.\,\delta^2}}\,.3^{-\frac{(x-\mu)^2}{2\,.\,\delta^2}} \quad (2.11)$$

Misalnya, *f* adalah kepadatan probabilitas kelas dan fiturnya, μ adalah sarana yang telah diperoleh sebelumnya oleh persamaan (2.9) dan δ^2 adalah varian yang diperoleh menggunakan persamaan (2.10). Dengan menggunakan persamaan (2.11), akan membentuk beberapa nilai yang akan menjadi nilai input dari rumus selanjutnya seperti yang ditunjukkan dalam Persamaan (2.12).

$$Evidence = \sum_{i=1}^{N} P\,.\,f(x;\mu)\,.\,f(x;\delta^2) \quad (2.12)$$

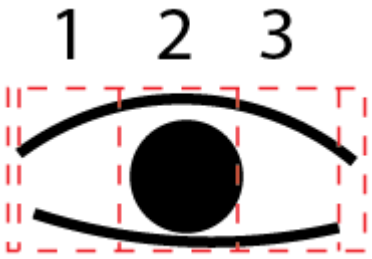
Nilai *P* adalah probabilitas kelas dalam data, *f (x: μ)* adalah sarana yang dihitung dengan persamaan (2.11) dan *f (x* δ^2*)* adalah varians yang dihitung dengan persamaan (2.11) juga.

Bukti digunakan untuk pembagi pada persamaan untuk menemukan nilai-nilai *Posterior* seperti yang ditunjukkan dalam Persamaan (2.13) di bawah ini:

$$Posterior(x) = \frac{\sum_{i=1}^{N} P\,.\,f(x;\mu)\,.\,f(x;\delta^2)}{Evidence} \quad (2.13)$$

2.7. Pixel Value

Pada permodelan ini, area mata dibagi menjadi tiga area besar, yaitu sector 1, 2 dan sector 3 dimana masing-masing sector mewakili 3 arah yang berbeda yaitu 1 sebagai kanan, 2 sebagai tengah dan 3 sebagai arah kiri. Ilustrasi pembagian sector ini ditunjukan pada Gambar 2.11.



Gambar 2.11. Pembagian sector mata

Pada Gambar 2.11 terlihat bahwa area keseluruhan mata dibagi menjadi 3 bagian besar, dimana masing-masing bagian memiliki luas area yang hampir sama. Tujuan dari dibaginya area mata ini adalah untuk menentukan kearah mana mata melihat yang dihitung berdasarkan banyaknya jumlah pixel berwarna hitam yang ada pada suatu sector area. Dalam persamaan matematika, kondisi pembagian dan penentuan arah gerakan bola mata dirumuskan pada persamaan 2.14 berikut:

$$sektor = \sum_{x=1}^{N}\sum_{y=1}^{M} I(b_k, k) \quad (2.14)$$

Pada tiap masing-masing sector memiliki nilai yang didapat dengan cara menjumlahkan seluruh pixel 1 atau hitam yang ada pada kolom area sector tersebut. Dimana *b* adalah baris dan *k* adalah kolom yang ada pada tiap sector area.

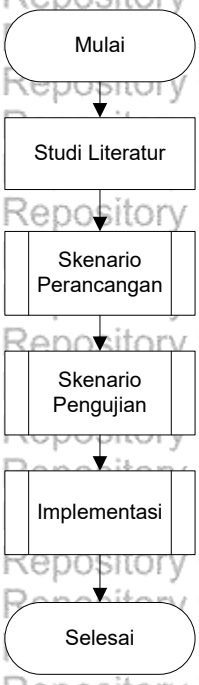
Terdapat 2 bagian dalam pendeteksian arah gerakan mata ini, bagian pertama adalah dengan cara membagi bagian mata menjadi 3 sektor seperti pada Gambar 4.4 dimana tujuannya adalah untuk menentukan arah gerakan mata ke kanan, kiri ataupun tengah. Pada bagian pertama ini, focus sistem hanya pada pendeteksian arah gerakan mata pada 3 arah saja, yaitu dengan cara menggunakan Persamaan 2.13 pada tiap sector lalu kemudian dibandingkan dengan cara memilih bagian sector mana yang memiliki nilai hasil penjumlahan pixel yang paling tinggi. Rumus perbandingan ditunjukan pada Persamaan 2.15 berikut :

$$arah = \begin{cases} kanan, if\ sektor_2 < sektor_1 > sektor_3 \\ tengah, if\ sektor_1 < sektor_2 > sektor_3 \\ kiri, if\ sektor_1 < sektor_3 > sektor_2 \end{cases} \quad (2.15)$$

BAB 3
METODOLOGI

Tahapan Penelitian

Tahapan penelitian ini disusun agar proses penelitian dapat dilakukan dengan sesuai skema dan berurutan dan sistematis juga terencana. Langkah-langkah yang akan dilakukan dalam penelitian ini diGambarkan pada Gambar 3.1.



Gambar 3.1 Diagram alur penelitian

Data yang digunakan dalam penelitian ini terbagi menjadi 2 bagian, yaitu data latih dan data uji. Adapaun lebih jelasnya adalah :

1. Dataset Gambar wajah dan mata yang berekstensi xml yang diperoleh dari library openCV yang merupakan sebuah data digital hasil training deteksi wajah dan mata. Data ini antinya akan digunakan sebagai data untuk mendeteksi region area wajah dengan menggunakan *Haar Cascade.*
2. Data uji pengguna yang diperoleh dengan melibatkan pengguna laki-laki dan perempuan dengan berbagai kondisi bentuk wajah dan mata (Gambar & video).
3. Data uji terdiri dari perorangan dengan ketentuan :
    1. Menggunakan kacamata dan tidak menggunakan kacamata untuk mengetahui kemampuan program dalam mendeteksi wajah yang berkacamata

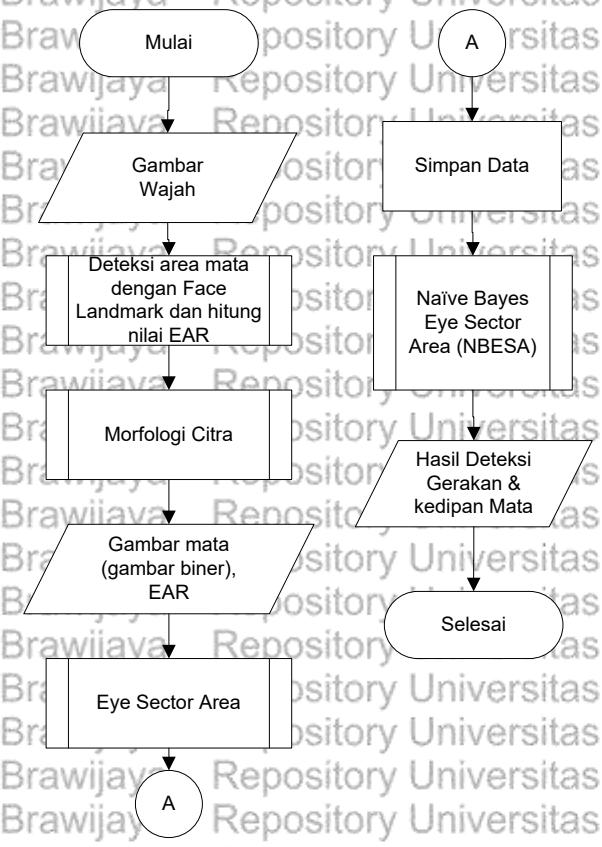
2. Gambar perorangan yang menggerakan bola matanya dan kelopak matanya (berkedip)

3.1. Studi Literatur

Tahapan penelitian dimulai dengan studi literatur terhadap berbagai referensi dari sumber-sumber seperti jurnal ilmiah nasional maupun internasional, artikel internet, ataupun buku, sebagai dasar teori yang dapat menunjang penelitian tesis ini. Adapun berdasarkan studi literatur yang telah digali, penelitian disusun berdasarkan:

1. Metode pendeteksian area wajah dan mata dengan menggunakan *Haar Cascade*
2. Metode pendeteksian gerakan bola mata dengan menggunakan *Naïve Bayes Eye Sector Area (NBESA)*
3. Metode pendeteksian gerakan kelopak mata atau kedipan dengan menggunakan EAR dan berbasis *Face Landmark*
4. Metode dan teori atau pendekatan lain yang mendukung dengan penelitian ini, digunakan sebagai pembanding ataupun digunakan dan diterapkan dalam penelitian ini.

3.2. Skenario Perancangan Program



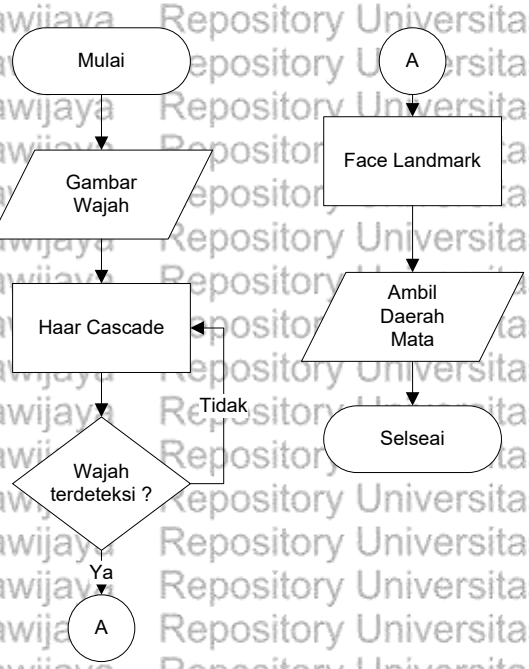
Gambar 3.2. Flowchart Perancangan Sistem

Pada Gambar 3.2 diatas, terdapat beberapa tahapan dalam sistem yang dibangun, penjelasannya adalah sebagai berikut :

1. Dimulai dengan masuknya Gambar input berupa Gambar wajah user atau pengguna
2. Gambar kemudian diolah kembali pada proses selanjutnya yaitu proses untuk mendapatkan area mata dengan menggunakan *Face Landmark* yang dijelaskan pada subbab 3.2.1.
3. Dilanjutkan dengan proses Morfologi Citra yang berfungsi untuk mengolah Gambar input (Gambar wajah), adapun proses pengolahan tersebut dijelaskan dalam subbab 3.2.2.
4. Hasil dari proses *Face Landmark* tersebut akan menghasilkan data berupa Gambar mata saja tanpa kelopak mata juga nilai *Eye Aspect Ratio* (EAR). Lalu Gambar mata tersebut diolah dengan menggunakan metode *Eye Sector Area* (ESA) yang dijelaskan pada subbab 3.2.3.
5. Nilai EAR dan hasil dari perhitungan metode ESA tersebut kemudian disimpan dan digunakan sebagai data referensi atau data latih. Yang kemudian dilakukan proses pendeteksian gerakan mata dengan menggunakan metode *Naïve Bayes Eye Sector Area* (NBESA) yang dijelaskan pada subbab 3.2.4.

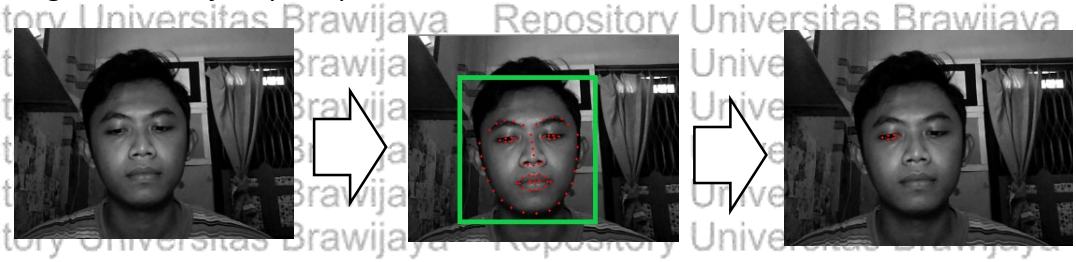
3.2.1. Deteksi Area Mata dengan Face Landmark

Sebelum mendeteksi area mata, terlebih dahulu dilakukan scanning dengan menggunakan *Haar Cascade* untuk mendeteksi wajah. Hal ini agar focus pencarian area mata lebih terarah dan terbatas sehingga dapat mengurangi kesalahan deteksi. Setalah itu pada area wajah dilakukan proses labeling dengan menggunakan *Face Landmark*. Adapun mekanisme dari proses ini ditunjukan pada Gambar 3.3.



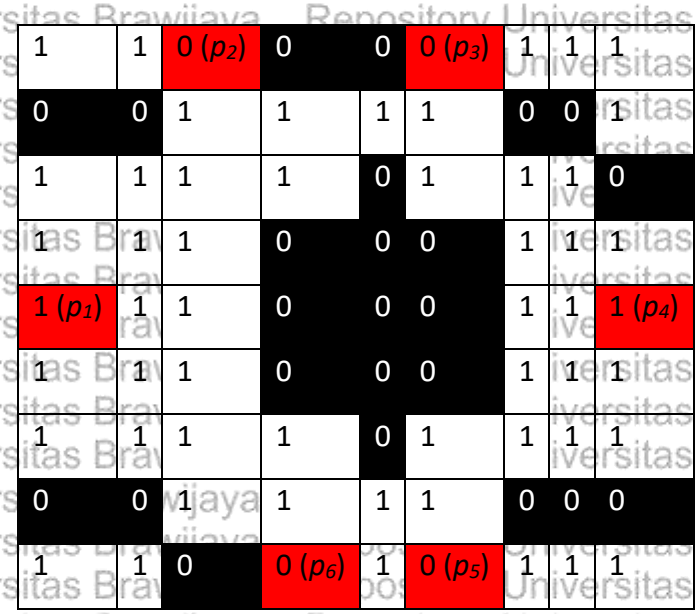
Gambar 3.3. Flowchart deteksi area mata

Gambar yang mengandung wajah sebagai input pertama kali akan dilakukan proses pendeteksian wajah dengan *Haar Cascade,* setelah area wajah dapat ditemukan maka selanjutnya adalah menempatkan titik-titik face landmark pada wajah tersebut, penempatan tersebut lalu dikoreksi dengan melakukan proses *regression tree* yang sudah ada pada *Face Landmark DLIB library.* Hasil dari penempatan titik face landmark tersebut lalu diseleksi dengan hanya mengambil bagian mata saja seperti pada Gambar 3.4.



Gambar 3.4. Hasil dari deteksi Face Landmark area mata

Dari Gambar 3.4, dengan memanfaatkan titik-titik yang ada pada mata maka dapat diberlakukan proses perhitungan untuk mencari nilai EAR dengan menggunakan persamaan (2.1) yang ada pada Bab 2. Adapun diberikan sebuah contoh pada Gambar 3.5.



Gambar 3.5. Ilustrasi mata terbuka dengan kotak merah sebagai kotak koordinat eyelandmark

Pada Gambar 3.5 terdapat beberapa kotak berwarna merah yang merepresentasikan titik koordinat area *eyelandmark* dengan menggunakan persamaan untuk menghitung EAR, maka dapat dihitung :

Koordinat p_1 = (0,4) Koordinat p_2 = (2,0)

Koordinat p_3 = (5,0) Koordinat p_4 = (8,4)

Koordinat p_5 = (5,8) Koordinat p_6 = (3,8)

$$EAR = \frac{||(2,0)-(3,8)||+||(5,0)-(5,8)||}{2||(0,4)-(8,4)||} = \frac{7}{4} = 1.75$$

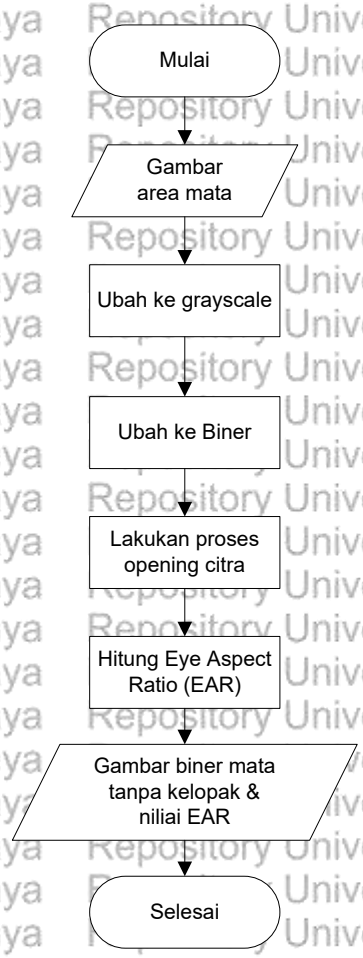
Dari hasil perhitngan tersebut, didapatkan nilai EAR adalah 1.75 yang berarti mata terbuka. Nilai EAR akan mengecil apabila kelopak mata semakin rapat atau semakin tertutup dan apabila mata tertutup maka nilai EAR adalah 0 atau mendekati 0.

Langkah selanjutnya adalah proses *cropping* area mata saja yang dibatasi oleh titik-titik merah. Dengan begitu, maka hasil *cropping* akan terbatas hanya pada area mata saja tanpa adanya kelopak mata. Daerah mata disini adalah Gambar mata tanpa kelopak mata, dimana hasil dari proses ini nantinya akan dimasukan kedalam proses *preprocessing*.

3.2.2. Morfologi Citra

Dalam tahap *Morfologi citra* ini terdapat beberapa langkah yang harus dilakukan, yang paling utama adalah memperbaiki hasil citra sehingga proses yang dilakukan sesudahnya akan lebih mudah. Selain itu dalam proses *preprocessing* ini juga dilakukan beberapa metode untuk pengambilan nilai fitur seperti nilai EAR yang berfungsi untuk mengetahui kerapatan antara kelopak mata bagian atas dan kelopak mata bagian bawah. Untuk Gambar mata yang sudah di *cropping* pada proses sebelumnya, dilakukan proses pengolahan seperti *opening* yang memiliki tujuan agar Gambar hasil binerisasi tidak rusak sehingga menyebabkan tidak tersampainya informasi yang dimiliki oleh Gambar tersebut.

Secara garis besar, langkah-langkah dalam mengolah Gambar input (Gambar area mata hasil proses *Face Landmark*) adalah sebagai berikut, ditunjukan dengan Gambar 3.6.



Gambar 3.6. Flowchart proses Morfologi Citra

Tujuan utama dari proses ini adalah untuk memperjelas Gambar pupil mata yang berwarna hitam, sehingga perlu adanya sebuah deklarasi 2 tipe warna

saja yaitu hitam dan putih (Gambar biner), sedangkan dalam merubah RGB menjadi biner dibutuhkan nilai *threshold* yang berfungsi sebagai pembatas antara kelompok pixel yang bernilai 0 dan kelompok pixel yang bernilai 1. Proses binerisasi ini sendiri diawali dengan proses *Grayscaling* terlebih dahulu untuk merubah Gambar RGB menjadi bernilai keabuan yang tersusun dari unsur warna putih hingga hitam saja. Misalkan terdapat sebuah Gambar area mata hasil proses *Face Landmark* sebagai berikut (Gambar 3.7) :

121-100-134	120-101-112	130-177-90	121-100-134	120-101-112	130-177-90	121-100-134	120-101-112	130-177-90
102-177-90	100-100-134	122-100-134	102-177-90	100-100-134	122-100-134	102-177-90	100-100-134	122-100-134
121-101-112	103-101-112	107-101-112	121-101-112	42-45-52	107-101-112	121-101-112	103-101-112	107-101-112
121-100-134	120-101-112	130-177-90	45-45-52	121-101-112	55-45-52	121-100-134	120-101-112	130-177-90
102-177-90	100-100-134	122-100-134	37-45-52	42-45-52	41-45-52	102-177-90	100-100-134	122-100-134
121-101-112	103-101-112	107-101-112	55-45-52	52-45-52	54-45-52	121-101-112	103-101-112	107-101-112
121-100-134	120-101-112	130-177-90	121-100-134	55-45-52	130-177-90	121-100-134	120-101-112	130-177-90
102-177-90	100-100-134	122-100-134	102-177-90	100-100-134	122-100-134	102-177-90	100-100-134	122-100-134
121-101-112	103-101-112	107-101-112	121-101-112	103-101-112	107-101-112	121-101-112	103-101-112	107-101-112

Gambar 3.7. Ilustrasi Gambar pixel mata, area warna biru merepresentasikan pupil mata yang berwarna hitam

Pada Gambar 3.6, diilustrasikan bahwa terdapat Gambar matriks dengan ukuran 9x9 dimana terdapat 11 kotak berwarna biru yang mengGambarkan pupil mata yang berwarna hitam. Sesuai dengan kaidah perancangan yang telah dijelaskan sebelumnya maka langkah yang pertama adalah merubah Gambar tersebut menjadi format *binary* yang memiliki format nilai 0 dan 1.

3.2.2.1. *Grayscaling* Gambar

Binerisasi Gambar membutuhkan Gambar dengan format *grayscale* atau Gambar keabu-abuan. Perubahan ke *grayscale* tersebut dapat dilakukan dengan rumus rata-rata, misalnya :

Pada pixel (0,0) yaitu dengan nilai RGB 121-100-134, maka :

121+100+134/3 = 118

Maka nilai *grayscale* pada pixel koordinat (0,0) adalah 118. Hasil dari perubahan ke grayscale ditunjukan pada Gambar 3.8.

118	111	132	118	111	132	118	111	132
123	111	118	123	111	118	123	111	118
111	105	106	111	46	106	111	105	106
118	111	132	47	111	50	118	111	132
123	111	118	45	46	46	123	111	118
111	105	106	50	49	50	111	105	106
118	111	132	118	50	132	118	111	132
123	111	118	123	111	118	123	111	118
111	105	106	111	105	106	111	105	106

Gambar 3.8. Hasil Grayscale Gambar RGB

3.2.2.2. Binerisasi Gambar

Setelah proses *grayscale,* proses selanjutnya adalah merubah Gambar *grayscale* menjadi biner dengan menggunakan *Threshold* atau batas nilai. Misalkan dalam penelitian ini nilai variabel ambang atau *threshold* adalah 60, maka nilai-nilai yang melebihi 60 akan dianggap sebagai warna 1 atau putih dan nilai-nilai yang ada dibawah batas 60 dianggap sebagai nilai hitam atau 0. Hasil dari proses binerisasi ditunjukan pada Gambar 3.9.

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	0	1	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Gambar 3.9. Hasil proses binerisasi Gambar Grayscale

Untuk lebih memperjelas Gambar maka dilakukan proses morfologi citra seperti dilasi ataupun erosi, pada penelitian ini proses dilasi dan erosi dikombinasi sedemikian rupa sehingga membentuk proses yang bernama *opening* dengan tujuan untuk memperjelas area warna hitam pada Gambar hasil proses dilasi, terlihat pada pixel dengan koordinat (4,3) terdapat pixel dengan nilai 1 yang berada diantara pixel dengan nilai 0. Proses Opening sendiri diawali dengan proses erosi yang diikuti dengan proses dilasi.

Dengan menggunakan kernel berukuran 3x3, maka dilakukan proses erosi pada Gambar 3.10 hasil dari proses binerisasi,

1	1	1
1	1	1
1	1	1

Gambar 3.10. Kernel erosi

Contoh penerapan perhitungan kernel pada Gambar biner ditunjukan pada Gambar 3.11. Dimana jendela kernel akan bergerak terus sampai keseluruhan Gambar terlewati atau dengan istilah lain *sliding window*.

Dimulai dengan citra koordinat (0,0) sampai dengan (2,2), maka :

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	0	1	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Gambar 3.11. Ilustrasi area kernel pada Gambar biner

Warna merah adalah area yang dilalui oleh kernel 3x3, dengan ketentuan apabila nilai-nilai pixel yang ada pada area merah tersebut cocok atau sesuai dengan kernel maka nilai pada koordinat tengah akan dirubah menjadi 1. Namun apabila ada satu saja nilai yang tidak sesuai dengan kernel, maka nilai tengah akan diberikan nilai 0. Hasil dari proses erosi ditunjukan pada Gambar 3.12 berikut :

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0
0	1	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0

Gambar 3.12. hasil proses erosi

Setelah proses erosi, maka dilanjutkan dengan proses dilasi terhadap Gambar hasil dari proses dengan menggunakan kernel 3x3 yang sama pada Gambar 4.9. Prinsip pada proses Dilasi adalah nilai tengah akan ditubah menjadi 1 apabila ada minimal

1 nilai pixel yang sama engan pixel pada kernel, dan nilai tengah bernilai 0 apabila tidak ada sama sekali pixel yang memiliki nilai sama dengan nilai pixel pada Gambar. Hasil dari proses Dilasi ditunjukan pada Gambar 3.13.

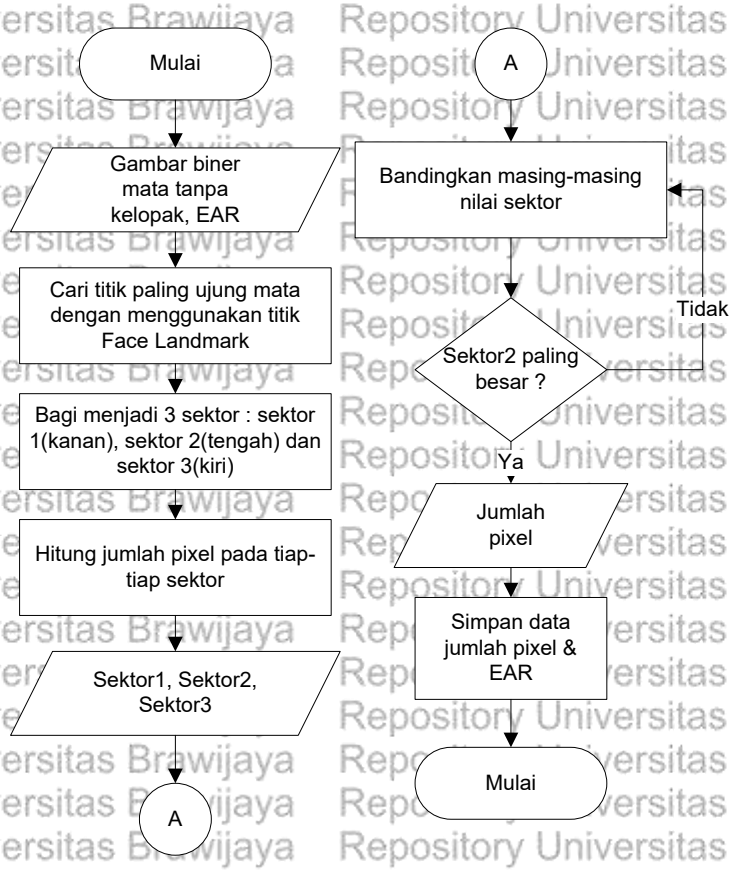
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Gambar 3.13. Hasil proses Dilasi

Gambar 3.13. menunjukan hasil yang diperoleh setelah melakukan proses *Opening* yang bertujuan untuk memperbaiki kualitas citra dengan harapan agar tidak ada informasi yang tidak tersampaikan pada Gambar tersebut. Dari proses ini, maka hasil yang diperoleh dapat diteruskan untuk diproses pada proses selanjutnya yaitu proses *Eye Sector Area* (ESA) untuk mengetahui arah gerakan mata.

3.2.3. Eye Sector Area (ESA)

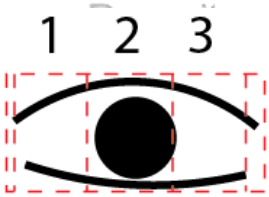
Eye Sector Area (ESA) adalah sebuah proses yang dilakukan dengan cara menghitung banyaknya nilai pixel yang terkandung dalam sebuah bagian atau sector pada mata, ditunjukan oleh Flowchart berikut (Gambar 3.14)



Gambar 3.14. Flowchart Eye Sector Area (ESA)

Setelah Gambar melalui proses *preprocessing,* maka langkah selanjutnya adalah proses *Eye Sector Area* (ESA). Dimana proses ini dilakukan dengan membagi area mata menjadi 3 sektor utama dengan patokan titik terluar bagian kiri dan titik terluar bagian kanan.

Pada permodelan ini, area mata dibagi menjadi tiga area besar, yaitu sector 1, 2 dan sector 3 dimana masing-masing sector mewakili 3 arah yang berbeda yaitu 1 sebagai kanan, 2 sebagai tengah dan 3 sebagai arah kiri. Ilustrasi pembagian sector ini ditunjukan pada Gambar 3.15.



Gambar 3.15.. Pembagian sector mata

Pada Gambar 3.15 terlihat bahwa area keseluruhan mata dibagi menjadi 3 bagian besar, dimana masing-masing bagian memiliki luas area yang hampir sama. Tujuan dari dibaginya area mata ini adalah untuk menentukan kearah mana mata melihat yang dihitung berdasarkan banyaknya jumlah pixel berwarna hitam yang ada pada suatu sector area. Dalam persamaan matematika, kondisi pembagian dan penentuan arah gerakan bola mata dirumuskan pada persamaan 3.1 berikut :

$$sektor = \sum_{x=1}^{N} \sum_{y=1}^{M} I(b_k, k) \quad (3.1)$$

Pada tiap masing-masing sector memiliki nilai yang didapat dengan cara menjumlahkan seluruh pixel 1 atau hitam yang ada pada kolom area sector tersebut. Dimana *b* adalah baris dan *k* adalah kolom yang ada pada tiap sector area.

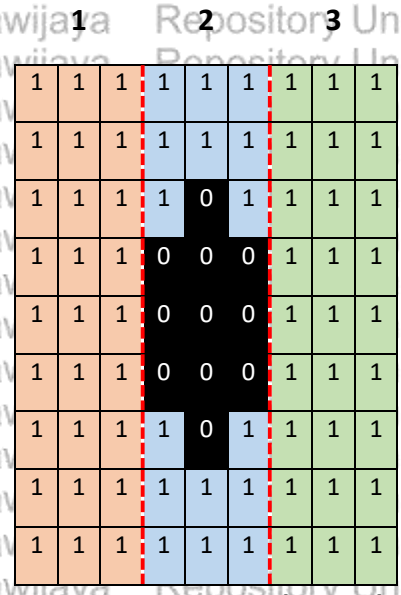
Terdapat 2 bagian dalam pendeteksian arah gerakan mata ini, bagian pertama adalah dengan cara membagi bagian mata menjadi 3 sektor seperti pada Gambar 4.4 dimana tujuannya adalah untuk menentukan arah gerakan mata ke kanan, kiri ataupun tengah. Pada bagian pertama ini, fokus sistem hanya pada pendeteksian arah gerakan mata pada 3 arah saja, yaitu dengan cara menggunakan Persamaan 4.1 pada tiap sector lalu kemudian dibandingkan dengan cara memilih bagian sector mana yang memiliki nilai hasil penjumlahan pixel yang paling tinggi. Rumus perbandingan ditunjukan pada Persamaan 3.2 berikut :

$$arah = \begin{cases} kanan, if\ sektor_2 < sektor_1 > sektor_3 \\ tengah, if\ sektor_1 < sektor_2 > sektor_3 \\ kiri, if\ sektor_1 < sektor_3 > sektor_2 \end{cases} \quad (3.2)$$

Dari Persamaan 4.2, dapat diketahui beberapa kondisi yang menentukan arah gerakan mata, adalah :

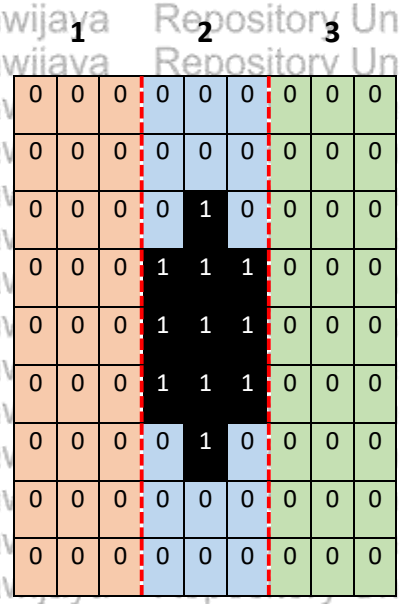
1. Arah kanan apabila nilai pada sector 1 paling tinggi jika dibandingkan dengan sector 2 dan sector 3
2. Arah tengah apabila nilai pada sector 2 paling tinggi jika dibandingkan dengan sector 1 dan sector 3
3. Arah kiri apabila nilai pada sector 3 paling tinggi jika dibandingkan dengan sector 1 dan sector 2.

Adapun contoh dalam perhitungan *Eye Sector Area* (ESA) adalah dicontohkan pada sebuah Gambar 3.16 hasil proses *preprocessing*:



Gambar 3.16. Ilustrasi pemagian sector pada Gambar mata hasil binerisasi

Sebelum proses perhitungan nilai pixel masing-masing area dilakukan, hal yang perlu diperhatikan adalah komponen nilai yang ada pada Gambar, dimana pixel hitam (pupil) diwakili oleh nilai 0 dan pixel putih diwakili oleh nilai 1. Maka dari itu perlu adanya proses *invers* atau pembalikan warna dimana warna 0 menjadi 1 dan warna 1 menjadi 0. Hasil dari proses invers ditunjukan pada Gambar 3.17.



Gambar 3.17. Hasil proses invers

Proses selanjutnya adalah dengan menghitung jumlah pixel pada masing-masing sector, yaitu dilakukan dengan cara menjumlah keseluruhan pixel satu persatu pada tiap sector, misalkan:

Pada sector 1 (kiri), maka :

0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0 = 0

Lalu pada sector 2 (tengah) :

0+0+0+0+0+0+0+1+0+1+1+1+1+1+1+1+1+1+0+1+0+0+0+0+0+0+0+0 = 11

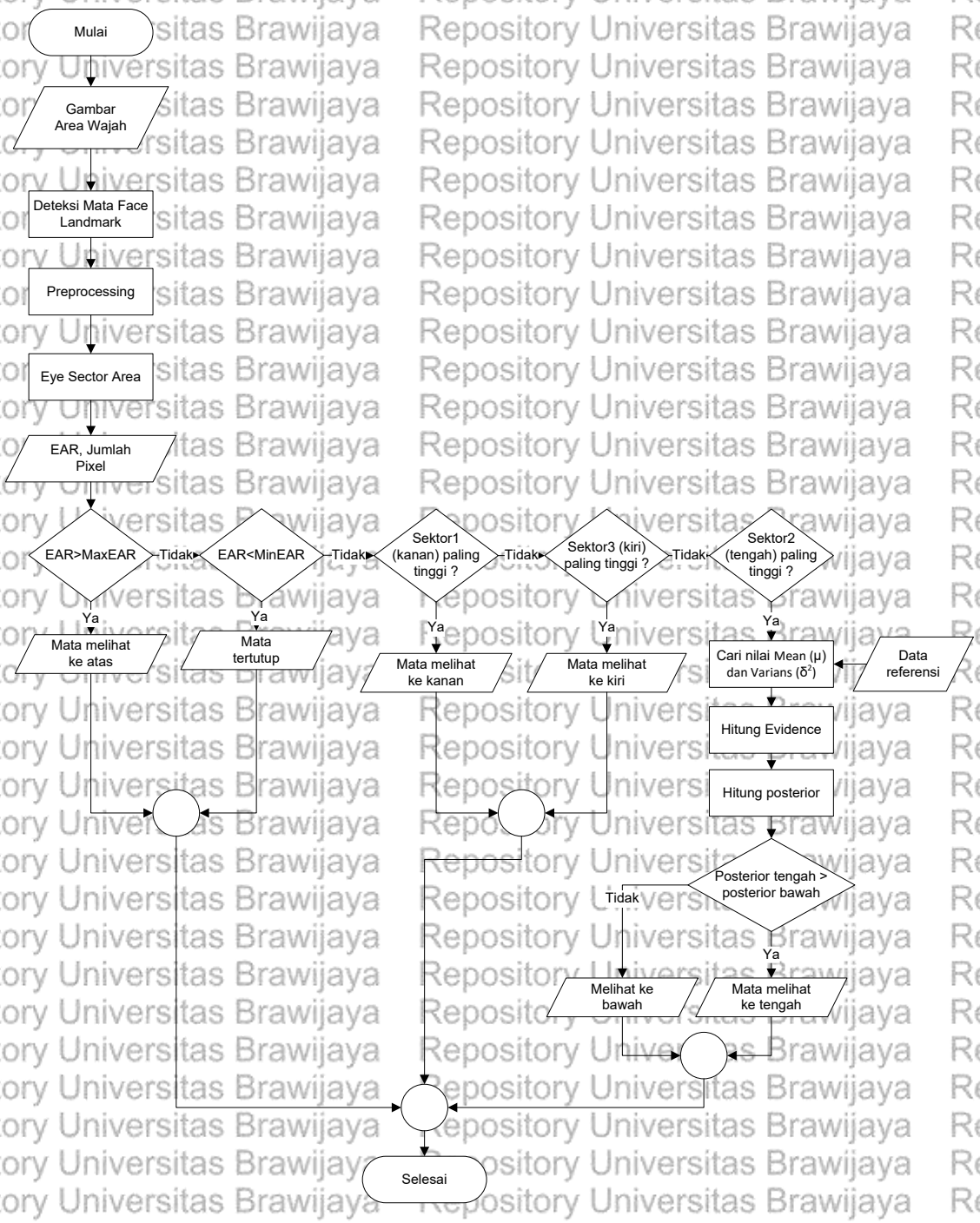
Dan pada sector 3 (kanan) adalah :

0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0 = 0

Dari perhitungan sector tersebut, maka didaptkan jika sector tengah memiliki nilai paling tinggi yaitu 11. Dengan begitu maka pergerakan bola mata dideteksi melihat ke arah tengah. Begitupun juga apabila sector lain yang memiliki nilai paling tinggi diantara sector-sektor lainnya, maka sector tersebut yang merepresentasikan arah gerak mata. Dengan menggunakan metode ini, sistem sudah dapat menentukan arah gerakan mata, namun hanya dapat mendeteksi 3 arah saja yaitu kanan, tengah dan kiri.

Merujuk pada tujuan dari penelitian ini, bahwa penelitian ini ingin memgembangkan pendekatan dan metode yang memiliki kemampuan dalam mendeteksi arah gerakan mata ke kanan, kiri, atas, bawah, tengah dan juga sekaligus dapat mendeteksi adanya kedipan maka perlu adanya tambahan metode ataupun kolaborasi metode atau pendetekatan yang memungkinkan sistem dapat memwujudkan hal tersebut.

3.2.4. Deteksi gerakan mata dengan metode *Naïve Bayes Eye Sector Area* (NBESA)



Gambar 3.18. Flowchart deteksi gerakan mata dengan metode usulan (NBESA)

Dengan hanya memanfaatkan kemampuan dari *Eye Sector Area (ESA)* saja, maka sistem hanya dapat mendeteksi arah gerakan mata pada tiga bagian saja

yaitu kanan, tengah dan kiri. Maka dari itu, pendekatan ini perlu dimodifikasi sehingga mampu meningkatkan kemampuan deteksi arah gerakan mata. Penambahan metode yang dilakukan adalah dengan cara menggunakan nilai variable dari hasil proses EAR atau *Eye Aspect Ratio.* Nilai dari EAR ini berupa sebuah nilai dari range 0 hingga ke atas, dimana nilai EAR ini merepresentasikan seberapa besar jarak yang ada antara kelopak mata bagian atas dengan kelopak mata bagian bawah.

Penggunaan metode EAR ini tentunya digabungkan dengan menggunakan metode *Naïve Bayes* sebagai metode klasifikasinya, tujuannya adalah untuk mengetahui arah gerakan bola mata secara lebih lengkap meliputi arah atas, bawah, kanan, kiri, bawah dan kedipan. Penggunaan pendeketan metode ini akan aktif apabila bola mata terdeteksi sedang berada di sector tengah, pada saat bola mata berada di sector tengah maka sistem akan menghitung nilai EAR dan juga jumlah pixel yang kemudian diklasifikasi dengan menggunakan *Naïve Bayes* sebagai metode klasifikasinya. Penggunaan EAR dan jumlah pixel pada saat bola mata berada ditengah adalah untuk memperjelas posisi bola mata mengingat perbedaan antara mata melihat ke depan, atas dan juga bawah tidaklah begitu signifikan, maka perlu adanya metode bantuan yang dianggap mampu untuk menangani hal tersebut. Perbedaan akan semakin kecil ketika posisi mata berada di tengah dan juga bawah, terlebih ketika mata melihat kebawah maka secara otomatis kelopak mata bagian atas akan menutupi area mata seiring dengan pergerakan kebawah yang dilakukan oleh bola mata. Kondisi tersebut ditunjukan pada Gambar 3.19.

Gambar 3.19. Ilustrasi mata melihat kebawah, warna merah adalah kelopak mata yang ikut menutupi area bola mata mengikuti pergerakan bola mata

Kelopak mata akan terus mengikuti pergerakan bola mata, pada saat mata melihat kebawah maka kelopak mata bagian atas akan menutupi area bola mata bagian atas, dan apabila bola mata melihat ke arah atas maka kelopak bagian atas juga akan ikut bergerak ke atas, maka dari itu untuk menentukan arah gerakan bola mata bisa dilakukan salah satunya dengan mempertimbangkan besaran nilai EAR yang dihasilkan dari pergerakan kelopak mata bagian atas, yang akan semakin membentuk jarak apabila bola mata semakin bergerak ke atas, dan juga sebaliknya apabila bola mata bergerak ke bawah maka jarak dan juga rasio antara kelopak mata bagian atas dan kelopak mata bagian bawah akan semakin mengecil.

Pada penggunaan metode Eye Sector Area (ESA) hal yang paling susah untuk ditentukan adalah menentukan perbedaan antara mata melihat ke arah bawah dengan mata tertutup, dikarenakan mata yang melihat kebawah hampir mirip dengan mata tertutup. Maka dari itu *Naïve Bayes* digunakan untuk

mengklasifikasi arah bola mata ketika bola mata berada di sector tengah (atas, tengah dan bawah). Untuk kondisi atas sendiri dapat dideteksi dengan memanfaatkan perubahan nilai EAR, namun untuk arah bawah tidak bisa hanya dengan memanfaatkan nilai EAR, perlu adanya variable lain yang digunakan untuk mendeteksi secara akurat pada arah bawah ini.

3.2.5. Penentuan Arah Gerak Bola Mata Berdasarkan Data Registrasi

Setiap user yang menggunakan sistem ini, pergerakan mata mereka haruslah disimpan terlebih dahulu, maksud dan tujuannya adalah untuk mengetahui nilai standar yang dimiliki oleh mata pada saat mata melihat tengah atau idle juga ketika mata melihat kea rah bawah. Seperti yang sudah dijelaskan sebelumnya, kondisi paling susah untuk sistem dalam mendeteksi arah gerakan adalah ketika mata melihat ke arah bawah dan ketika mata melihat ke tengah, terlebih pada saat mata melihat kebawah, akan sangat mirip sekali dengan saat mata menutup. Maka dari itu perlu dilakukan proses regristasi yang tujuannya adalah untuk membentuk suatu nilai yang berguna untuk data acuan atau data training dengan menggunakan *Naïve Bayes.*

Dikarenakan kemampuan yang dimiliki oleh metode *Naïve Bayes* dalam klasifikasi tidak terpengaruh dengan banyaknya data yang diolah, maka pada saat proses registrasi data yang disimpan tidak dalam jumlah yang banyak, tujuannya adalah untuk membantu mempersingkat proses komputasi yang dilakukan. Data yang disimpan dalam proses registrasi adalah data jumlah pixel dan besaran nilai EAR, data tersebut disimpan didalam file sementara untuk kemudian diakses sebagai data latih atau data training yang digunakan untuk perhitungan dengan metode *Naïve Bayes.*

Data uji yang digunakan pada tahap registrasi meliputi 2 arah yaitu arah tengah dan arah bawah. Arah tengah pada sistem dianggap sebagai kondisi normal mata (melihat ke depan), dimana ketika mata pada posisi ini maka sistem akan menyimpan nilai jumlah pixel dan nilai EAR. Selain ketika mata pada kondisi melihat ke depan, data latih juga membutuhkan kondisi mata ketika melihat ke bawah. Perbedaan pixel antara mata melihat ke tengah dan mata melihat ke bawah ditunjukan pada Gambar 3.20.

Gambar 3.20 (a) kondisi pixel ketika mata melihat ke depan (tengah), (b) kondisi pixel ketika mata melihat ke bawah

Pada Gambar 3.20. dapat dilihat ketika mata melihat ke arah tengah, maka jumlah pixel berwarna putih akan lebih banyak jika dibandingkan dengan ketika mata melihat kebawah, kondisi tersebut yang dijadikan patokan dengan cara meregistrasi nilai-nilai variable jumlah pixel disertai dengan nilai EAR yang didapat dari besarnya rasio pembukaan kelopak mata. Adapun contoh perhitungan metode *Naïve Bayes Eye Sector Area* (NBESA) adalah dicontohkan sebagai berikut :

Tabel 3.1. Ilustrasi data latih Naïve Bayes

EAR	Jumlah Pixel	Arah
1.75	11	Tengah
0.55	4	Bawah
1.50	13	Tengah
0.7	6	Bawah

Dari data ini, maka dapat dicari Mean (μ) dan Varians (δ^2) yang dimiliki, dengan cara :

$$\mu(\text{EAR}, \text{Tengah}) = \frac{(1.75 + 1.50)}{2} = 1.625$$

$$\delta^2(EAR, Tengah) = \frac{(1.75 - 1.625)^2 + (1.50 - 1.625)^2}{1} = 0.015625$$

$$\mu(\text{EAR}, \text{Bawah}) = \frac{(0.55 + 0.7)}{2} = 0.625$$

$$\delta^2(EAR, Bawah) = \frac{(0.6 - 0.625)^2 + (0.7 - 0.625)^2}{1} = 0{,}00625$$

$$\mu(\text{JP}, \text{Tengah}) = \frac{(11 + 13)}{2} = 12$$

$$\delta^2(JP, Tengah) = \frac{(11 - 12)^2 + (12 - 12)^2}{1} = 1$$

$$\mu(\text{JP}, \text{Bawah}) = \frac{(4 + 6)}{2} = 5$$

$$\delta^2(JP, Bawah) = \frac{(4 - 5)^2 + (6 - 5)^2}{1} = 2$$

Tabel 3.2. Ilustrasi data uji Naïve Bayes

EAR	Jumlah Pixel	Arah
1.5	11	kanan / kiri / atas / bawah / tengah / kedipan ?

Lalu hitung prosterior masing-masing kelas :

$$p(EAR|Tengah) = \frac{1}{\sqrt{2.\pi.0.015625}}.e^{-\frac{(1.5-1.625)}{2.0.015625}} = 4.3$$

$$p(JP|Tengah) = \frac{1}{\sqrt{2.\pi.1}}.e^{-\frac{(11-12)}{2.1}} = 0.542$$

$$p(EAR|Bawah) = \frac{1}{\sqrt{2.\pi.0.00625}}.e^{-\frac{(1.5-0.625)}{2.0.00625}} = 0.000026$$

$$p(JP|Tengah) = \frac{1}{\sqrt{2.\pi.2}}.e^{-\frac{(11-5)}{2.2}} = 0.06$$

Evidence = 0.5*4.3*0.542 + 0.5*0.000026*0.06 = 1.1653 + 0.00000078 = 1.1653

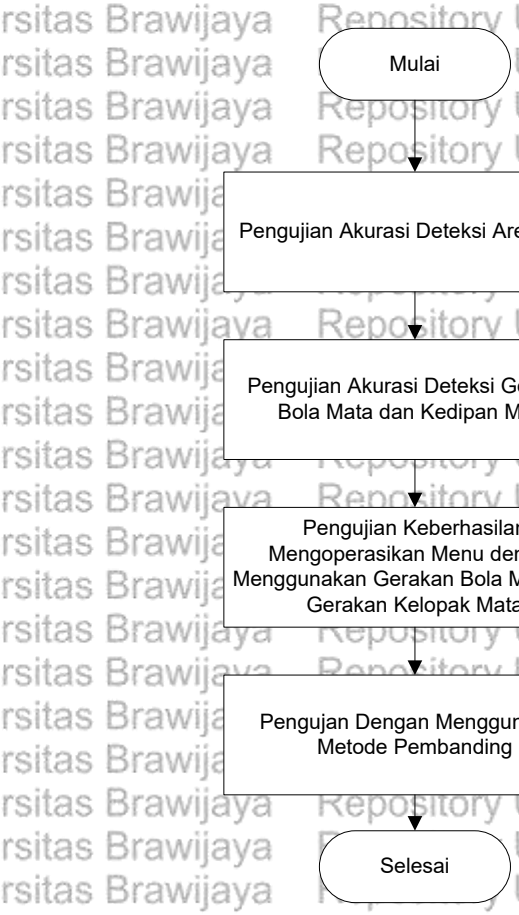
Posterior(Tengah) = $\frac{0.5*4.3*0.542}{1.1653} = 1$

Posterior(Bawah) = $\frac{0.5*0.000026*0.06}{1.1653} = 0.000000069$

Karena Posterior Tengah lebih tinggi maka data dengan nilai EAR = 1.5 dan Jumlah Pixel = 11 terdeteksi melihat kearah ***Tengah.***

3.3. Skenario Pengujian

Pengujian dirancang untuk mengetahui tingkat kehandalan deteksi pergerakan bola mata juga deteksi pergerakan kelopak mata atau kedipan. Pengujian pendeteksian gerakan bola mata yang dilakukan adalah menguji kevalidan sistem dalam mendeteksi arah gerakan bola mata. Arah yang dimaksud adalah gerakan ke Depan, Kanan, Kiri, Atas dan Bawah. Adapun pengujian deteksi kedipan adalah kesesuaian komputer dalam mendeteksi adanya gerakan penutupan kelopak mata dalam satu satuan waktu tertentu. Diagram skenario pengujian ditunjukan melalui Gambar 3.21.



Gambar 3.21 Diagram Sekenario Pengujian

4. Pengujian pengaruh jarak terhadap akurasi *Haar Cascade* dan *Face Landmark* dalam mendeteksi area mata. Pengujian ini dilakukan dengan cara mengukur ketepatan metode *Haar Cascade* dan *Face Landmark* untuk mendeteksi area mata pada sebuah citra masukan dengan jarak tertentu.

Tabel 3.3 Pengujian metode Haar dalam mendeteksi area mata

Data uji	Sampel Gambar	Status Deteksi	Jarak
1			
2			
…			
Data uji ke n			

5. Pengujian pengaruh cahaya, jarak dan nilai *threshold* terhadap akurasi deteksi arah pandangan mata sebagai kondisi *idle* atau normal (melihat kedepan) dilakukan untuk mengetahui kemampuan sistem dalam mendeteksi arah gerakan bola mata secara real time dalam melihat ke arah depan yang

dibangun dengan menggunakan metode *Naïve Bayes Eye Sector Area.* Pengujian dilakukan dengan cara membandingkan hasil yang diperoleh melalui komputer dengan kondisi sebenarnya. Pengujian ini berfungsi untuk mencari nilai-nilai parameter yang paling memiliki akurasi tinggi.

Tabel 3.4 Pengujian Deteksi arah pandangan mata sebagai kondisi *idle* atau normal dengan kondisi cahaya, jarak dan nilai *threshold* yang berbeda

Data	Threshold	lux	Jarak	Hasil Deteksi
Data 1				
Data 2				
Data …				
Data N				

Dari hasil perbandingan tersebut lalu dihitung tingkat akurasinya dengan menggunakan Persamaan 3.1.

$$Accuracy(\%) = \frac{Jumlah\ Benar}{Jumlah\ Data\ uji}\ x\ 100\% \qquad (3.1)$$

6. Pengujian untuk mengetahui kemampuan dan akurasi metode usulan dalam mendeteksi arah gerakan mata ke kanan, atas, bawah, kiri, tengah dan juga kedipan dengan menggunakan parameter yang telah ditentukan melalui hasil proses pengujian sebelumnya. Untuk mengetahui tingkat akurasi yang dimiliki adalah dengan menggunakan peritungan pada Persamaan 3.1.

Tabel 3.5 Pengujian deteksi pergerakan kelopak mata / gerakan menutup mata

No	Data	Kondisi gerakan mata					
		tengah	atas	bawah	kanan	kiri	tertutup
1	Data 1	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
2	Data 2	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
..	Data …	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi
N	Data N	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi	Hasil Deteksi

7. Pengujian Keberhasilan Mengoperasikan Menu dengan Gerakan Bola Mata dan Gerakan Kelopak Mata dilakukan untuk mengetahui tingkat keberhasilan user dalam mengoperasikan menu dengan menggunakan mata sebagai alat kontrol pengganti tombol dan pointer. Pengujian dilakukan pada kondisi pencahayaan terbaik dan pengujian dilakukan dengan menggunakan *precision* dan *recall,* perhitungan *precision* dan *recall* dapat dijelaskan seperti berikut :

$$precision\ (\%) = \frac{TP}{TP+FP} x100\% \quad (3.2)$$

$$recall\ (\%) = \frac{TP}{TP+FN} x100\% \quad (3.3)$$

Keterangan :

TP : True Positive bernilai 1, jika *actual class* bernilai benar dan *predicted class* bernilai benar

FP : False Positive bernilai 1, jika *actual class* bernilai benar dan *predicted class* bernilai salah, dan melibatkan seluruh tipe kelas data.

FN : False Negative bernilai 1, jika *actual class* bernilai benar dan *predicted class* bernilai salah, dan melibatkan satu tipe kelas data.

TN : total data yang tidak terjadi pada kelas data *multiclass* label.

Pengambilan nilai *TP* (*True Positive*), *TN* (*True Negative*), *FP* (*False Positive*) dan *FN* (*False Negative*) adalah dengan menggunakan table rujukan *Confusion Matrix* yang ditunjukan pada Tabel 3.3.

Tabel 3.6 Mekanisme penilaian *Confusion Matrix*

actual	**tengah**	TP	FP/FN	FP/FN	FP/FN	FP/FN	FP/FN	FP/FN
	kanan	FP/FN	TP	FP/FN	FP/FN	FP/FN	FP/FN	FP/FN
	Kiri	FP/FN	FP/FN	TP	FP/FN	FP/FN	FP/FN	FP/FN
	atas	FP/FN	FP/FN	FP/FN	TP	FP/FN	FP/FN	FP/FN
	bawah	FP/FN	FP/FN	FP/FN	FP/FN	TP	FP/FN	FP/FN
	tertutup	FP/FN	FP/FN	FP/FN	FP/FN	FP/FN	TP	FP/FN
		tengah	**kanan**	**kiri**	**atas**	**bawah**	**tertutup**	**Tidak Terdeteksi**
		prediction						

3.4. Implementasi

Pada subbab ini dijelaskan bagaimana rancangan sistem diterapkan pada computer dengan menggunakan kode-kode program. Adapun kode-kode program tersebut dibagi menjadi beberapa proses, diantaranya :

3.4.1. Face Landmark

Hal yang pertama dilakukan pada proses Face Landmark adalah mengambil input Gambar untuk kemudian diproses untuk mencari wajah, kode untuk mengambil Gambar video adalah sebagai berikut :

```
while(True):
    # Capture frame-by-frame
    #cap.set(1,50)
    ret, frame = cap.read()
    arahGerak, inverse = detek(frame,arah,titik)
```

variabel *frame* akan di tangkap dan dimasukan kedalam fungsi *landmrk* untuk mencari dan menentukan titik-titik wajah dengan Face Landmark, kodenya :

```
def landmrk(frame):
    image = frame
    rects = detector(image, 1)
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]


    global kanan
    global kiri
    global mulut
    global pinggirMulut
    global ear
    global kotak


    for (i,rect) in enumerate(rects):
        shape = predictor(image, rect)
        shape = face_utils.shape_to_np(shape)
        rightEye = shape[rStart:rEnd]
        kanan = earMata(rightEye)
        ear = (kanan)/2
        a,b = shape[17]
        c,d = shape[21]
        x,y = shape[28]
        r,t = shape[36]
        l,u = shape[39]
        kotakKiri = image[b:5,a:r]
        jarakKiri = np.size(kotakKiri,1)
        kotakKanan=image[u:5,l:c]
        jarakKanan = np.size(kotakKanan,1)
        kotak = image[b+5:y,a:c]
        lebar = np.size(kotak,1)
        break

        for (x, y) in rightEye:
            cv2.circle(image, (x, y), 1, (0, 0, 0), 3)


    return kotak,ear, image, jarakKiri,jarakKanan
```

Penjelasan program :

1. Baris 1-2 : mendapatkan inputan Gambar yang masuk kedalam fungsi

2. Baris 4 : menentukan titik-titik untuk bagian mata sebelah kanan
3. Baris 7-12 : deklarasi variable
4. Baris 15-35 : menentukan titik-titik yang ada pada mata dengan menggunakan variable "rightEye" yang telah terdefinisikan sebelumnya
5. Baris 19 : mengirimkan area mata untuk dihitung nilai EAR nya
6. Baris 28 : mengembalikan nilai berupa Gambar, nilai EAR dan jarak antara titik terluar sebelah kiri dan kanan pupil mata (tengah) dengan bagian kiri dan kanan mata sebelah kanan

3.4.2. Morfologi Citra

```
4.  def olahCitra(Gambar):
5.      frame = Gambar
6.      frame = imutils.resize(frame, width=300)
7.      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
8.      gray = np.dstack([gray,gray,gray])
9.      tampil,ear,image, batasKiri,batasKanan = lm.landmrk(gray)
10.     height = np.size(tampil,0)
11.     width = np.size(tampil,1)
12.     #print "lebar",width,batasKiri,batasKanan
13.     batasKiriKecil = 100*batasKiri/width
14.     batasKananKecil = 100*batasKanan/width
15.     kotakBesar = imutils.resize(tampil,width=100)
16.     kotakBlur = cv2.cvtColor(kotakBesar, cv2.COLOR_BGR2GRAY)
17.     kotakBesar = cv2.medianBlur(kotakBesar,5)
18.     ret,kotakBesar = cv2.threshold(kotakBesar,90,255,cv2.THRESH_BINARY)

19.     #kotakBesar = cv2.adaptiveThreshold(kotakBesar,255,cv2.ADAPTIVE_THRE
    SH_GAUSSIAN_C,cv2.THRESH_BINARY,21,2)
20.     kernel = np.ones((5,5),np.uint8)
21.     kotakBesar = cv2.morphologyEx(kotakBesar,cv2.MORPH_CLOSE, kernel)
22.
23.
24.
25.     return kotakBesar,ear,image, batasKiriKecil, batasKananKecil, width
```

Penjelasan program :

1. Baris 2 : mendapatkan input Gambar
2. Baris 3 : merubah ukuran citra
3. Baris 4-5 : merubah ke gray
4. Baris 6 : mengirim ke proses landmark untuk mendapatkan nilai jarak kiri dna jarak kanan (bataskanan & bataskiri)
5. Baris 10-11 : menentukan nilai variable batas kanan dan batas kiri mata yang berguna untuk proses *Eye Sector Area* (ESA)
6. Baris 15 : merubah ke biner dengan nilai threshold yang nantinya dicari yang paling efektif pada saat pengujian
7. Baris 18 : melakukan proses Closing untuk memperbaiki kualitas citra / menutup lubang-lubang yang ada pada pupil setelah proses binerisasi

3.4.3. Eye Sector Area (ESA)

```
1.  while(True):
2.      ret, frame = cap.read()
3.      kotakBesar,ear, tampil, batasKiri,batasKanan,lebarKecil = citra.olah
    Citra(frame)
4.      print batasKanan,batasKiri
5.      #=============CARI JUMLAH PIXEL TIAP REGION=======================
    =====#
6.      Gambar = kotakBesar
7.      inverse = (255-Gambar)
8.      panjangTengah=100-float(batasKanan+batasKiri)
9.      region = int(panjangTengah/3)
10.     height = np.size(inverse,0)
11.     width = np.size(inverse,1)
12.     a=0
13.     b=[]
14.     for x in range(0,width):
15.         for y in range(0,height):
16.             a =a+inverse[y][x][0]
17.         b.append(a)
18.         a=0
19.     kanan = np.sum(b[batasKiri:batasKiri+region])
20.     tengah = np.sum(b[batasKiri+region:batasKiri+region+region])
21.     kiri = np.sum(b[batasKiri+region+region:batasKiri+region+region+regi
    on])
22.     isi = [kanan,tengah,kiri]
23.     jumlah = sum(isi)
24.     tinggi = isi.index(max(isi))
25.     if(jumlah==0):
26.         print "mata tertutup"
27.     else:
28.         if(tinggi==1):
29.             print "tengah"
30.         elif(tinggi==2):
31.             print "kiri"
32.         elif(tinggi==0):
33.             print "kanan"
```

Penjelasan program :

1. Baris 2 : input Gambar
2. Baris 3 : mendapatkan area mata dan nilai-nilai koordinat untuk menghitung jumlah pixel berdasarkan letak titik mata. Perlu diketahui bahwa nilai-nilai ini dilakukan pencarian setiap saat dikarenakan sistem ini adalah real time maka sistem harus mampu beradaptasi ketika user merubah kepalanya seperti miring ke kiri atau ke kanan. Namun masih tetap dalam batas yang wajar
3. Baris 7 : inverse Gambar biner (0 menjadi 1 dan 1 menjadi 0)
4. Baris 18-21 : menghitung jumlah yang ada pada masing-masing region

Baris 22-33 : membandingkan nilai pada tiap-tiap region

3.4.4. Naïve Bayes Eye Sector Area (NBESA)

3.4.4.1. Naïve Bayes

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 19 23:09:01 2018

@author: gusti
"""
from io import StringIO
import numpy as np
import math


data = np.loadtxt('test.out', skiprows=0)
print data
pixelMax = max(data[0][0],data[1][0],data[2][0],data[3][0])
pixelMin = min(data[0][0],data[1][0],data[2][0],data[3][0])
earMax = max(data[0][1],data[1][1],data[2][1],data[3][1])
earMin = min(data[0][1],data[1][1],data[2][1],data[3][1])
tengah1Pixel = data[0][0]-pixelMin/pixelMax-pixelMin+0.1
tengah1Ear = data[0][1]-earMin/earMax-earMin+0.1
tengah1Kelas = int(data[0][2])
tengah2Pixel = data[1][0]-pixelMin/pixelMax-pixelMin+0.1
tengah2Ear = data[1][1]-earMin/earMax-earMin+0.1
tengah2Kelas = int(data[1][2])
bawah1Pixel = data[2][0]-pixelMin/pixelMax-pixelMin+0.1
bawah1Ear = data[2][1]-earMin/earMax-earMin+0.1
bawah1Kelas = int(data[2][2])
bawah2Pixel = data[3][0]-pixelMin/pixelMax-pixelMin+0.1
bawah2Ear = data[3][1]-earMin/earMax-earMin+0.1
bawah2Kelas = int(data[3][2])
dataTengahPixel = [tengah1Pixel, tengah2Pixel]
dataTengahEar = [tengah1Ear, tengah2Ear]
dataBawahPixel = [bawah1Pixel, bawah2Pixel]
dataBawahEar = [bawah1Ear, bawah2Ear]
def miu(data):
    sumData = np.sum(data)
    banyakData = len(data)
    mean = sumData / banyakData
    return mean

def teta(data, mean):
    atas = 0
    for i in data:
        atas = atas + math.pow((i-mean),2)
    theta = atas / (len(data)-1)
    return theta

def prior(u,t,nilai):
    bawah = math.sqrt(2*math.pi*t)
    #print bawah
    pangkatAtas = math.pow(nilai-u,2)
    pangkatBawah = 2*t
```

```
52.     hasilPangkat = -(pangkatAtas/pangkatBawah)
53.     #print "hasil pangkat:", hasilPangkat
54.     kiri = 1/bawah
55.     kanan = math.pow(2.72,hasilPangkat)
56.     hasil = kiri*kanan
57.     #print hasil
58.     return hasil
59.
60. def bayes(ear,pixel):
61.
62.     miuEarTengah = miu(dataTengahEar)+0.1
63.     miuPixelTengah = miu(dataTengahPixel)+0.1
64.     miuEarBawah = miu(dataBawahEar)+0.1
65.     miuPixelBawah = miu(dataBawahPixel)+0.1
66.     tetaEarTengah = teta(dataTengahEar, miuEarTengah)+0.1
67.     tetaPixelTengah = teta(dataTengahPixel, miuPixelTengah)+0.1
68.     tetaEarBawah = teta(dataBawahEar, miuEarBawah)+0.1
69.     tetaPixelBawah = teta(dataBawahPixel, miuPixelBawah)+0.1
70.     probTengah = float(2.0/6.0)
71.     probBawah = float(2.0/6.0)
72.     priorEarTengah = prior(miuEarTengah,tetaEarTengah,ear)
73.     priorEarBawah = prior(miuEarBawah, tetaEarBawah, ear)
74.     priorPixelTengah = prior(miuPixelTengah, tetaPixelTengah, pixel)
75.     priorPixelBawah = prior(miuPixelBawah, tetaPixelBawah, pixel)
76.     evidence = (probTengah*priorEarTengah*priorPixelTengah)+(probBawah*p
    riorEarBawah*priorPixelBawah)
77.     posteriorTengah = (probTengah*priorEarTengah*priorPixelTengah)/evide
    nce
78.     posteriorBawah = (probBawah*priorEarBawah*priorPixelBawah)/evidence
79.     return posteriorTengah, posteriorBawah, earMin, earMax
80.
81.
```

Penjelasan program :

1. Baris 12 : membaca data yang telah diregistrasi
2. Baris 14-33 : menentukan nilai variabel
3. Baris 34-38 : fungsi untuk menghitung mean pada *Naïve Bayes*
4. Baris 40-45 : untuk menghitung teta
5. Baris 47-58 : menghitung nilai prior
6. Baris 60-79 : menghitung nilai yang ada pada variable dengan menggunakan fungsi-fungsi pada baris sebelumnya sebelum mengembalikan nilai untuk dijadikan sebagai referensi data untuk proses pengujian, nilai yang dihitung adalah nilai variable yang digunakan pada *Naïve Bayes* meliputi rata-rata data, varians, probabiitas, posterior masing-masing kelas, evidence dan posterior dimana nilai-nilai tersebut akan digunakan ketika proses pengujian *Naïve Bayes*.

3.4.4.2. Pengujian Naïve Bayes

```
1. def detek(frame, arah,titik):
2.
3.     kotakBesar,ear,tampil,batasKiri,batasKanan,lebarKecil = citra.olahCi
   tra(frame)
```

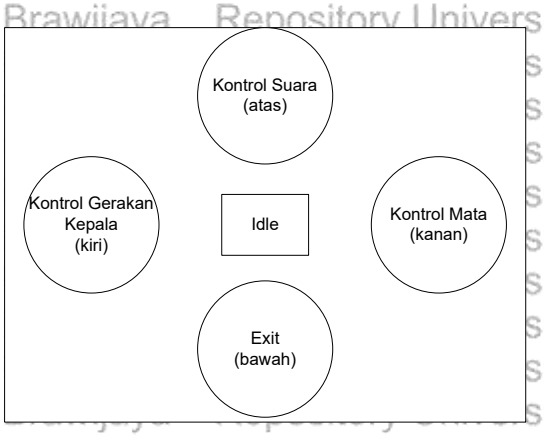
Repository Universitas Brawijaya

```
4.
5.      Gambar = kotakBesar
6.      inverse = (255-Gambar)
7.      panjangTengah=100-(batasKanan+batasKiri)
8.      region = int(panjangTengah/3)
9.      height = np.size(inverse,0)
10.     width = np.size(inverse,1)
11.     a=0
12.     b=[]
13.     for x in range(0,width):
14.         for y in range(0,height):
15.             a =a+inverse[y][x][0]
16.         b.append(a)
17.         a=0
18.     #print b
19.     kanan = np.sum(b[batasKiri:batasKiri+region])
20.     tengah = np.sum(b[batasKiri+region:batasKiri+region+region])
21.     kiri = np.sum(b[batasKiri+region+region:batasKiri+region+region+regi
    on])
22.     isi = [kanan,tengah,kiri]
23.     #print isi
24.     jumlah = sum(isi)
25.     #print "jumlah",jumlah
26.     tinggi = isi.index(max(isi))
27.     pilihan=""
28.     tengah, bawah, earMin, earMax = bayes.bayes(ear, tengah)
29.     center = [tengah,bawah]
30.     maks = center.index(max(center))
31.     if(jumlah>1 and tinggi==0):
32.         #arah="kanan"
33.         arah.append("kanan")
34.         titik.append((30,100))
35.     elif(jumlah>1 and tinggi==2):
36.         #arah="kiri"
37.         arah.append("kiri")
38.         titik.append((270,100))
39.     elif(jumlah>1 and tinggi==1):
40.         if(maks==0):
41.             if not arah:
42.                 arah.append('gerak')
43.                 titik.append((150,100))
44.             else:
45.                 arah=arah
46.         else:
47.             #arah="bawah"
48.             arah.append("bawah")
49.             titik.append((150,200))
50.     elif(jumlah<=1):
51.         if(ear<=earMin):
52.             pilihan = "pilih "+str((arah[len(arah)-1]))
53.         elif(ear>=earMax):
54.             #arah="atas"
55.             arah.append("atas")
56.             titik.append((150,25))
57.     cv2.circle(tampil,(titik[len(titik)-1]),20,(255,0,0),-1)
58.     cv2.putText(tampil, str(pilihan),(2,200),cv2.FONT_HERSHEY_SIMPLEX, 1
    ,(255,255,255),2)
59.     #lingkaran atas
60.     cv2.circle(tampil,(150,25),20,(0,255,0),3)
61.     #lingkaran bawah
62.     cv2.circle(tampil,(150,200),20,(0,255,0),3)
```

```
63.     #lingkaran kanan
64.     cv2.circle(tampil,(30,100),20,(0,255,0),3)
65.     #lingkaran kiri
66.     cv2.circle(tampil,(270,100),20,(0,255,0),3)
67.     cv2.imshow('landmark',tampil)
68.
69.     return arah,inverse
```

Penjelasan program :

1. Baris 3-5 : deklarasi nilai variabel
2. Baris 6-10 : invers data yang masuk (data real time wajah)
3. Baris 13-21 : menghitung *Eye Sector Area* (ESA)
4. Baris 31-56 : kondisi if dengan penjelasan :
   a. Apabila variable jumlah yang terpiliah adalah 1 dan nilai tertinggi pada array adalah 0, maka arah mata melihat ke kanan
   b. Apabila variable jumlah memiliki nilai lebihd dari 1 dan nilai tertinggi pada array adalah index 2, maka arah mataadalah melihat ke kiri
   c. Apabila jumlah variable jumlah lebih dari 1 dan jika nilai tertinggi adalah di index 1 (tengah) maka :
      i. Jika masih belum ada pergerakan terdeteksi maka sistem akan manmpilkan pesan untuk mempersilahkan user menggerakan matanya, namun jika mata user telah bergerak maka ketika ada pada posisi ini sistem akan menyimpan data pergerakan yang dilakukan sebelumnya
      ii. Jika variable maks bernilai 1 maka itu tandanya bahwa pada proses pencocokan data dengan metode *Naïve Bayes* menghasilkan pilihan mata melihat ke bawah
   d. Apabila jumlah kurang dari sama dengan 1 maka terdapat dua kemungkinan :
      i. Jika EAR kurang dari nilai EARMin maka arah mata adalah berkedip atau tertutup
      ii. Jika EAR lebih dari EARMax maka arah mata adalah melihat ke atas.
5. Baris 57-67 : berfungsi untuk menampilkan visualisasi tombol dan perpindahan menu yang interaktif bagi user, adapun Gambaran dari GUI tersebut adalah seperti Gambar 4.20.



Gambar 3.21. Visualisasi tombol yang tampil pada layar user, apabila tombol terpilih maka akan menyala berwarna biru

BAB 4
HASIL DAN PEMBAHASAN

Bab ini membahas hasil serta pembahasan berdasarkan sekenario pengujian yang telah diimplementasikan seperti pengujian jarak optimal sistem dan juga pengujian nilai cahaya sistem sehingga dapat menemukan nilai akurasi yang paling tinggi serta pengujian pendekatan *Eye Sector Area* (ESA) dan *Eye Aspect Ratio* (EAR) untuk mengetahui tingkat keberhasilan sistem dalam mengklasifikasi dan menentukan arah gerakan bola mata dan juga untuk mendeteksi adanya kedipan mata. Hasil pengujian dari metode yang di kembangkan ini juga dibandingkan dengan hasil yang didapat dari pengujian dengan menggunakan metode yang lainnya, sehingga dapat diketahui kelebihan serta kekurangan yang dimiliki oleh metode yang dikembangkan ini.

4.1. Spesifikasi Pengujian

Pada sub bab ini menjelaskan tentang spesifikasi pengujian yang digunakan dalam menguji metode deteksi pergerakan bola mata yang diajukan, spesifikasi yang digunakan dalam mekanisme pengujian ini meliputi spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi perangkat keras dan spesifikasi perangkat lunak ditunjukan pada Tabel 4.1 dan 4.2.

Tabel 4.1 Spesifiakasi perangkat keras

Processor	Intel® Core ™ i7-2670QM 2.2 GHz
Memory RAM	8 GB
Camera	Acer HD Webcam
LED Lamp	v-1706

Tabel 4.2 Spesifikasi perangkat lunak

Sistem Operasi	Windows 10 64 bit
Bahasa Pemrograman	Python
IDE	Spyder by Anaconda

4.2. Deteksi Mata

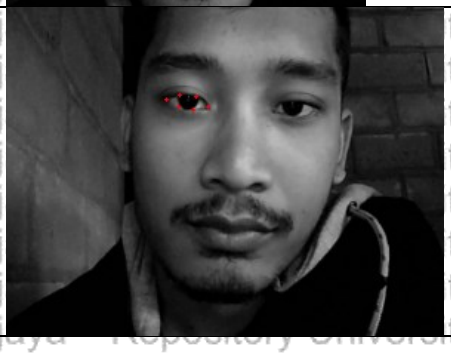
Pada subbab ini akan dilakukan pengujian untuk mengetahui kemampuan sistem dalam mendeteksi mata pada wajah user. Ada beberapa sekenario terhadap mekanisme pengujian dimana hal-hal yang diujikan mencakup kemampuan sistem dalam mendeteksi mata pada jarak tertentu, kemampuan sistem mendeteksi mata pada kondisi cahaya tertentu, kemampuan sistem dalam mendeteksi mata pada jarak tertentu dan dengan kondisi berkacamata dan tidak berkacamata. Dari beberapa sekenario pengujian tersebut, nantinya dapat ditarik

kesimpulan untuk menetapkan jarak yang ideal, kondisi pencahayaan yang ideal yang digunakan sebagai variable tetapan dan juga pengaruh penggunaan kacamata pada kemampuan deteksi sistem.

4.2.1. Pengaruh Jarak Terhadap Hasil Deteksi Mata

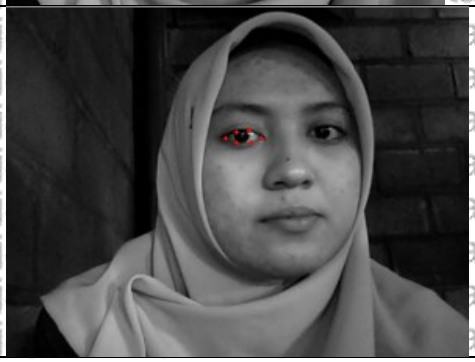
Jarak wajah dengan kamera memiliki pengaruh dalam hal pendeteksian letak mata yang berada di wajah, dengan melakukan pengujian maka dapat diketahui jarak ideal yang pas untuk digunakan dalam pendeteksian, terdapat 2 sekenario dalam pengujian ini, yang pertama adalah sekenario wajah tanpa menggunakan kacamata dan yang kedua adalah sekenario wajah dengan menggunakan kacamata. Adapun hasil dari pengujian data dengan tanpa menggunakan kacamata ditunjukan pada Tabel 4.3.

Tabel 4.3 Pengaruh jarak terhadap hasil deteksi mata (tanpa kacamata)

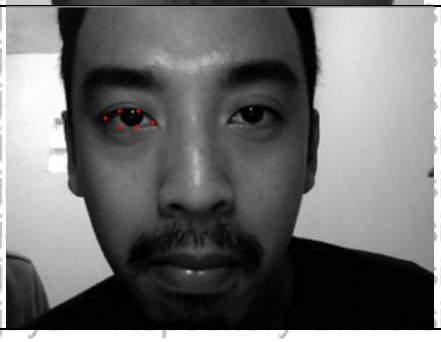
NO	DATA	JARAK (cm)	Hasil Deteksi	STATUS
1	Sampel 1	10		tidak
		20		terdeteksi
		30		terdeteksi

		40		terdeteksi
		50		terdeteksi
		60		terdeteksi
2	sampel 2	10		tidak
		20		terdeteksi

		30		terdeteksi
		40		terdeteksi
		50		terdeteksi
		60		terdeteksi

3	data 3	10		tidak
		20		terdeteksi
		30		terdeteksi
		40		terdeteksi

		50		terdeteksi
		60		terdeteksi
4	sampel 4	10		tidak
		20		terdeteksi
		30		terdeteksi

		40		terdeteksi
		50		terdeteksi
		60		terdeteksi
5	data 5	10		tidak
		20		terdeteksi

		30		terdeteksi
		40		terdeteksi
		50		terdeteksi
		60		terdeteksi

Dari 5 data percobaan dimana masing-masing data memiliki 5 jarak uji, maka dapat disimpulkan bahwa jarak efektif yang dapat diakomodasi oleh sistem dari rentang 10cm – 60cm hanyalah 20cm – 60cm diamana pada jarak 10cm dari kamera wajah tidak dapat terdeteksi sehingga pencarian area mata tidak dapat dilakukan. Selain percobaan tanpa menggunakan kacamata, dilakukan juga percobaan dengan menggunakan kacamata yang ditunjukan pada Tabel 5.4.

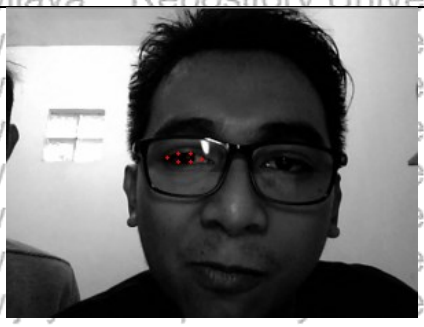
Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Repository Universitas Brawijaya

Table 4.4. Tabel uji coba deteksi mata pada user dengan kacamata

NO	DATA	JARAK	GAMBAR	HASIL
1	Sampel 6	10		Tidak Terdeteksi
		20		Terdeteksi
		30		Terdeteksi
		40		Terdeteksi
		50	Error	Tidak Terdeteksi
		60	Error	Tidak Terdeteksi

2	Sampel 2	10		Tidak Terdeteksi
		20		Terdeteksi
		30		Terdeteksi
		40		Terdeteksi
		50		Terdeteksi

		60		Terdeteksi
3	Sampel 3	10		Tidak Terdeteksi
		20		Terdeteksi
		30		Terdeteksi
		40		Terdeteksi

		50		Terdeteksi
		60	Error	Tidak Terdeteksi

Dapat terlihat pada Tabel 4.4, sistem juga mampu untuk mendeteksi area mata walau menggunakan kacamata. Dari Tabel 4.3 dan Tabel 4.4 dapat dibentuk sebuah Tabel kesimpulan untuk mekanisme pengujian deteksi area mata ini yang ditunjukan pada Tabel 4.5.

Table 4.5. Jumlah keberhasilan sistem dalam mendeteksi wajah user tanpa menggunakan kacamata

Kelompok Data	Jumlah Data Benar dalam mendeteksi mata tanpa kacamata (Berdasarkan jarak terhadap kamera)					
	10 cm	20 cm	30 cm	40 cm	50 cm	60 cm
Data Benar	0	5	5	5	5	5
Total Data	5	5	5	5	5	5
Prosentase	**0%**	**100%**	**100%**	**100%**	**100%**	**100%**

Table 4.6. Jumlah keberhasilan sistem dalam mendeteksi wajah user dengan menggunakan kacamata

Kelompok Data	Jumlah Data Benar dalam mendeteksi mata dengan kacamata (Berdasarkan jarak terhadap kamera)					
	10 cm	20 cm	30 cm	40 cm	50 cm	60 cm
Data Benar	0	3	3	3	2	1
Total Data	3	3	3	3	3	3
Prosentase	**0%**	**100%**	**100%**	**100%**	**67%**	**34%**

Dari Tabel 4.5 ditampilkan bahwa pada jarak 10 cm sistem tidak mampu untuk mendeteksi area mata, sedangkan pada jarak 20, 30, 40 dan 50 sistem secara sempurna dapat mendeteksi keseluruhan sample data. Namun pada jarak 60 cm sudah mulai terlihat kecenderungan sistem mulai tidak bisa mendeteksi area mata secara baik yang ditandai dengan berkurangnya jumlah data yang dapat dideteksi dengan benar. Hasil pengujian yang berbeda ditunjukan pada Tabel 4.6 dimana pada jarak 50 dan 60 cm mengalami penurunan prosentase keberhasilan deteksi mata. Kedua percobaan tersebut lalu di simpulkan pada Tabel 4.7 berikut.

Tabel 4.7. Rata-rata prosentase hasil deteksi benar data dengan kacamata dan data tanpa menggunakan kacamata

Kelompok Data	Jumlah Data Benar dalam mendeteksi mata dengan kacamata (Berdasarkan jarak terhadap kamera)					
	10 cm	20 cm	30 cm	40 cm	50 cm	60 cm
Prosentase hasill deteksi benar dengan kacamata	0%	100%	100%	100%	100%	100%
Prosentase hasill deteksi benar tanpa kacamata	0%	100%	100%	100%	67%	34%
Rata-rata Prosentase	**0%**	**100%**	**100%**	**100%**	**83.5%**	**67%**

Dari Tabel 4.7 diketahui bahwa pada jarak 20, 30 dan 40 cm rata-rata prosesntase adalah 100% dengan data gabungan antara yang menggunakan kacamata dan tidak menggunakan kacamata, sedangkan pada jarak 50 dan 60 cm terlihat penurunan nilai rata-rata. Merujuk pada hasil tersebut, maka dalam pengujian berikutnya untuk jarak 0, 50 dan 60 cm tidak akan digunakan sebagai parameter pengujian dikarenakan jumlah data yang dapat dideteksi mengalami penurunan dibandingkan dengan pada jarak 20, 30 dan 40 cm.

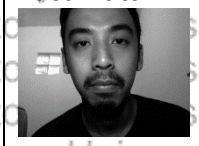
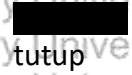
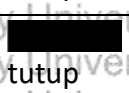
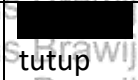
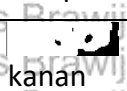
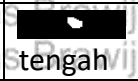
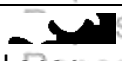
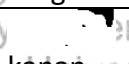
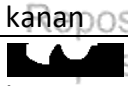
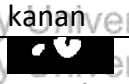
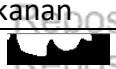
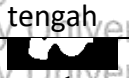
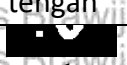
Selain pengaruh jarak yang sudah ditunjukan pada Tabel 4.5, 4.6 dan 4.7, perlu diketahui lebih lanjut apakah sistem juga dapat mendeteksi secara benar arah pandangan mata yang digunakan sebagai patokan (arah tengah). Apabila

sistem mampu untuk mendeteksi secara akurat arah pandangan mata ketika melihat ke tengah atau dalam kondisi normal maka masih terdapat kemungkinan sistem dapat mendeteksi arah gerakan mata dengan benar juga. Untuk mekanisme pengujian, dijelaskan pada subbab 4.2.2.

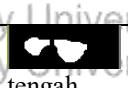
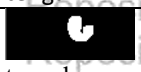
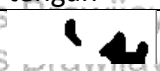
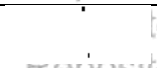
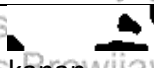
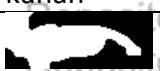
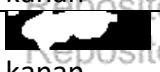
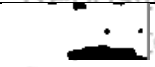
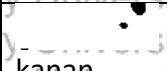
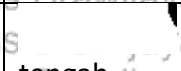
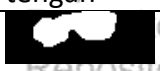
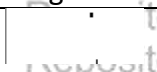
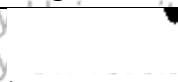
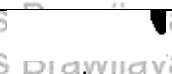
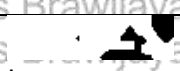
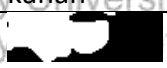
4.2.2. Pengaruh ketersediaan cahaya, jarak dan nilai threshold pada hasil deteksi arah tengah

Dari hasil pengujian yang ada pada Tabel 4.4, diketahui bahwa area wajah dan mata dapat terdeteksi dengan baik ketika berjarak antara 20-40 cm dari kamera. Dari hasil tersebut, maka kemudian digunakan untuk mengetahui jumlah cahaya yang pas agar nantinya sistem dapat bekerja dengan baik. Untuk itu perlu adanya percobaan untuk mendapatkan komposisi nilai cahaya dan juga nilai batas ambang atau *threshold* agar sistem dapat mendeteksi arah gerakan secara akurat. Jumlah cahaya dikatakan pas apabila hasil binerisasi area mata menunjukan bentuk yang tepat atau terlihat menyerupai bola mata dan menunjukan hasil deteksi yang tepat (arah tengah). Pada Tabel 4.6 ditampilkan hasil percobaan yang melibatkan nilai keadaan cahaya dan batas ambang *threshold* juga pengaruhnya terhadap hasil ekstrasi fitur mata.

Tabel 4.8. Hasil ekstrasi Gambar biner area mata dengan pengaruh jarak, cahaya dan nilai threshold

Data	Threshold	lux	Jarak		
			20 cm	30 cm	40 cm
Sampel 1 tanpa kacamata 	30	100	Tengah	tengah	kanan
		300	tengah	 tutup	tutup
		700	tengah	 tutup	 tutup
	65	100	kanan	kanan	 kanan
		300	tengah	tengah	tengah
		700	tengah	tengah	 tengah
	90	100	 kanan	 kanan	-
		300	 kanan	 tengah	tengah
		700	 kanan	 tengah	 tengah

Repository Universitas Brawijaya

Sampel 2 tanpa kacamata	30	100	tengah	 tengah	tengah
		300	 tengah	tengah	tengah
		700	tengah	tutup	tutup
	65	100	kanan	kanan	kanan
		300	tengah	tengah	tengah
		700	tengah	tengah	tengah
	90	100	kanan	kanan	 kanan
		300	kanan	tengah	tengah
		700	kanan	kanan	tengah
Sampel 3 dengan kacamata	30	100	 kanan	kanan	 kanan
		300	 kanan	tengah	tengah
		700	 kanan	tengah	tengah
	65	100	 kanan	 kanan	 tengah
		300	tengah	kanan	kanan
		700	 tengah	tengah	kanan
	90	100	 kanan	 kanan	 tengah
		300	Kanan	kanan	 kanan
		700	tengah	 kanan	

					kanan
Sampel 4 dengan kacamata	30	100	kanan	kanan	kanan
		300	kanan	tengah	tengah
		700	kanan	tengah	tengah
	65	100	kanan	kanan	tengah
		300	tengah	kanan	kanan
		700	tengah	tengah	kanan
	90	100	kanan	kanan	tengah
		300	Kanan	kanan	kanan
		700	tengah	kanan	kanan

Dari Tabel 4.8, dapat diketahui bahwa terdapat 4 data uji dengan masing-masing data uji memiliki variabel-variabel penguji ketentuan terdapat 3 nilai *threshold* yang berbeda untuk tiap-tiap data uji, nilai *threshold* tersebut adalah 30, 65 dan 90. Dalam tiap bagian bagian nilai *threshold* terdapat juga 3 nilai intensitas cahaya yang berbeda untuk diujikan yaitu 100 yang merepresentasikan keadaan lingkungan dengan cahaya yang redup, 300 yang merepresentasikan keadaan dengan cahaya yang cukup terang dan terakhir adalah 700 yang mewakili keadaan pencahayaan lingkungan yang sangat terang. Dari masing-masing variabel nilai cahaya yang ada pada tiap-tiap bagian nillai *threshold* juga terdapat tiga variabel jarak yaitu 20, 30 dan 40 yang sudah dipilih berdasarkan percobaan sebelumnya (Tabel 4.6 dan 4.7). Data kemudian diuji dengan menggunakan nilai-nilai variable yang sudah dipetakan pada Tabel 4.8, hasil dari pengujian data tersebut bernilai positif atau benar apabila sistem mendeteksinya sebagai arah tengah / idle yang artinya mata melihat tepat kedepan. Jika output keluaran adalah arah depan itu menandakan bahwa bola mata dapat di deteksi secara tepat dan akurat dengan menggunakan segmentasi dan binerisasi dimana ada titik pertemuan kombinasi yang pas antara komposisi nilai *threshold,* jarak dan cahaya sebagai variabel bebas yang mempengaruhi hasil segmentasi. Adapun kesimpulan dari Tabel 4.8 dipetakan dengan Tabel 4.9 berikut.

Tabel 4.9 Representasi hasil deteksi arah tengah sebagai kondisi idle dengan ketentuan terhitung benar apabila sistem mendeteksinya sebagai arah tengah

Kelompok Data	Threshold	Cahaya (lux)	Total Data terdeteksi benar (arah tengah)			Total Data Terdeteksi Benar	Jumlah Data	Rata-rata
			20 cm	30 cm	40 cm			
Tanpa Kacamata	30	100	2	2	1	5	6	83.4%
		300	2	0	0	2	6	33.4%
		700	2	0	0	2	6	33.4%
	Jumlah					**9**	**18**	**50%**
	65	100	0	0	0	0	6	0%
		300	2	2	2	6	6	100%
		700	2	2	2	6	6	100%
	Jumlah					**12**	**18**	**66.7%**
	90	100	0	0	0	0	6	0%
		300	0	2	2	4	6	66.7%
		700	0	1	2	3	6	50%
	Jumlah					**7**	**18**	**38.9%**
	RATA-RATA							**51.8%**
Dengan Kacamata	30	100	0	0	0	0	6	0%
		300	0	2	2	4	6	66.7%
		700	0	2	2	4	6	66.7%
	Jumlah					**8**	**18**	**44.5%**
	65	100	0	0	2	2	6	33.4%
		300	2	0	0	2	6	33.4%
		700	2	2	0	4	6	66.7%
	Jumlah					**8**	**18**	**44.5%**
	90	100	0	0	2	2	6	33.4%
		300	0	0	0	0	6	0%
		700	1	0	0	1	6	16.7%
	Jumlah					**3**	**18**	**16.7%**
	RATA-RATA							**35.2%**

Dari Tabel 4.9, dapat diketahui bahwa kelompok data yang menggunakan kacamata memiliki tingkat akurasi dibawah 50% dalam mendeteksi arah tengah secara benar yang ditunjukan dengan area berwarna biru, sedangkan kelompok data dengan tanpa menggunakan kacamata memiliki akurasi diatas 50% yang ditandai dengan warna kuning. Dari sini terlihat bahwa user atau pengguna dengan menggunakan kacamata lebih susah untuk dideteksi yang ditunjukan

dengan rata-rata tingkat akurasinya yang ada dibawah 50%. Rendahnya tingkat akurasi yang dimiliki oleh user dengan kacamata ini disebabkan oleh beberapa hal yaitu sistem kurang baik dalam mendeteksi letak posisi mata pada wajah yang sebelumnya sudah ditunjukan pada Tabel 4.4. Beberapa hasil deteksi letak area mata pada user dengan menggunakan kacamata menunjukan hasil yang kurang pas yang dicontohkan pada Gambar 4.1 berikut.

(a) **(b)**

Gambar 4.1. Perbandingan hasil deteksi area tepi mata dengan Face Landmark menunjukan bahwa (a) mata dengan kacamata memiliki kesalahan posisi titik (b) mata tanpa kacamata memiliki titik landmark yang pas

Selain sistem yang kurang baik dalam mendeteksi letak mata, penyebab lainnya adalah cahaya pantulan yang disebabkan oleh kacamata menyebabkan sistem kurang baik dalam melakukan proses segmentasi / binerisasi yang bertujuan untuk mendeteksi pupil hitam yang ada pada mata.

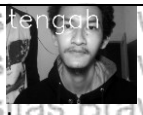
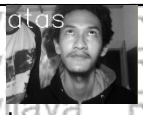
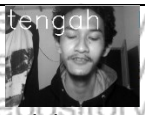
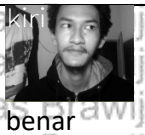
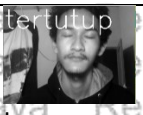
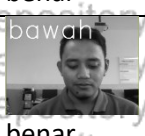
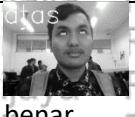
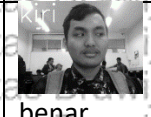
Setelah melalui beberapa tahapan proses untuk mencari nilai variabel yang paling tepat, maka dapat diambil sebuah kesimpulan berdasarkan hasil-hasil percobaan yang ditampilkan dalam beberapa Tabel sebelumnya diambil kesimpulan bahwa user tanpa menggunakan kacamata lebih memiliki tingkat akurasi yang tinggi dibandingkan dengan user dengan menggunakan kacamata dalam mendeteksi arah tengah, sehingga user tanpa kacamata lebih cocok dengan sistem ini. Selanjutnya, pengujian untuk deteksi arah dilakukan hanya untuk user tanpa menggunakan kacamata. Lalu untuk nilai *threshold* dimana pada pilihan nilai *threshold* 30, 65 dan 90, yang paling berkontribusi untuk membantu sistem dalam mendeteksi secara tepat dan akurat adalah nilai *threshold* dengan nilai sebesar 65. Juga untuk nilai pencahayaan, sesuai dengan Tabel 4.9 bahwa pencahayaan yang paling efektif dan tidak banyak membuat sistem gagal mendeteksi dengan benar adalah cahaya dengan rentang 300 hingga 700. Nilai-nilai variabel ini yang nantinya akan digunakan sebagai standar untuk pengujian metode deteksi gerakan bola mata dan kedipan yang akan dibahas pada subbab berikutnya.

4.3. Pengujian akurasi metode *Naïve Bayes Eye Sector Area* (NBESA) untuk mendeteksi arah gerakan mata dan kedipan mata dengan parameter yang sudah didapat

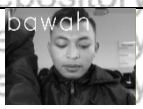
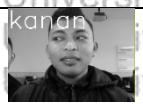
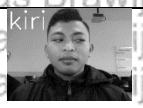
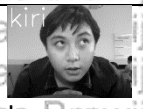
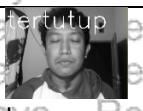
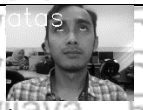
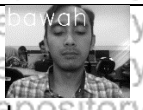
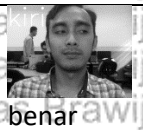
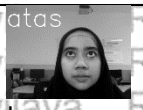
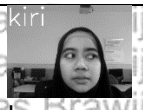
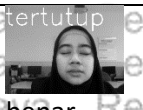
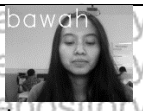
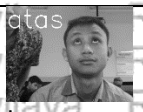
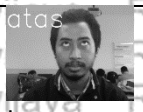
Pengujian ini dilakukan dengan melibatkan 20 orang (sampel) dimana pada tahap awal dilakukan registrasi nilai untuk arah tengah dan bawah. Seperti yang sudah dijelaskan dalam bab sebelumnya, registrasi ini bertujuan untuk mengetahui batasan-batasan area yang ada pada mata, dikarenakan metode ini memanfaatkan metode klasifikasi *Naïve Bayes* dengan jumlah pixel pada tiap sector area, maka dari itu perlu adanya pelatihan khsususnya untuk arah ke tengah dan bawah dikarenakan kedua arah ini memiliki perbedaan yang sangat kecil dan sama-sama berada di sector ke 2 atau tengah. Pada percobaan ini nilai-nilai variable yang digunakan adalah mengacu pada hasil percobaan sebelumnya dengan batasan-batasan jarak antara 20 hingga 40 cm, nilai threshold pada sistem untuk proses binerisasi adalah 65 sedangkan cahaya yang diberikan kepada user sebesar kurang lebih 300 lux. Sumber cahaya sendiri didapatkan dari LED yang diletakan didekat kamera untuk diarahkan ke wajah user sehingga cahaya yang diperoleh oleh user menjadi lebih stabil dibandingkan dengan memanfaatkan cahaya dari lingkungan yang sifatnya dinamis dan berubah-ubah.

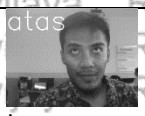
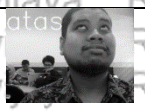
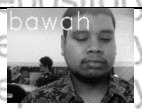
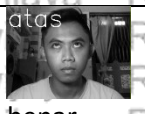
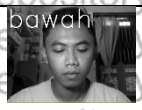
Pada Tabel 4.10 ditunjukan hasil deteksi arah gerakan bola mata dan gerakan menutup mata oleh sistem. Data yang di uji sebanyak 20 sampel dengan masing-masing sampel memiliki 6 gerakan yaitu atas, bawah, tengah, kiri, kanan dan menutup mata.

Tabel 4.10. Hasil uji akurasi deteksi gerakan mata metode *Naïve Bayes Eye Sector Area* (NBESA) dengan parameter cahaya 300 lux, jarak 30cm dan nilai threshold 65

No	Data	Kondisi gerakan mata					
		tengah	atas	bawah	kanan	kiri	tertutup
1	Sampel 1	 benar	 benar	 salah	benar	 benar	 benar
2	Sampel 2	benar	salah	benar	benar	benar	benar
3	Sampel 3	benar	benar	 benar	benar	benar	benar
4	Sampel 4	benar	 benar	salah	benar	 benar	benar

Repository Universitas Brawijaya

5	Sampel 5	tengah benar	atas benar	bawah benar	kanan benar	kiri benar	tertutup benar
6	Sampel 6	tengah benar	atas benar	 bawah benar	 kanan benar	 kiri benar	tertutup benar
7	Sampel 7	tengah benar	atas benar	salah	kanan benar	kiri benar	tertutup salah
8	Sampel 8	tengah benar	tengah salah	bawah benar	kanan benar	 kiri benar	tertutup benar
9	Sampel 9	tengah benar	atas benar	bawah benar	kanan benar	kiri benar	 tertutup benar
10	Sampel 10	tengah benar	 atas benar	 bawah benar	kanan benar	 kiri benar	tertutup benar
11	Sampel 11	tengah benar	 atas benar	bawah benar	kanan benar	 kiri benar	 tertutup benar
12	Sampel 12	tengah benar	atas benar	 bawah benar	kanan benar	kiri benar	tertutup benar
13	Sampel 13	tengah benar	 atas benar	bawah benar	kanan benar	kiri benar	tertutup benar
14	Sampel 14	tengah benar	atas benar	bawah benar	kanan benar	kiri salah	tertutup benar
15	Sampel 15	tengah benar	 atas benar	tengah salah	kanan benar	kiri benar	tertutup benar
16	Sampel 16	tengah benar	atas benar	bawah benar	kanan benar	kiri benar	tertutup benar
17	Sampel 17	tengah	atas	kiri	kanan	kiri	tertutup

		benar	benar	salah	benar	benar	benar
18	Sampel 18	benar	 benar	benar	benar	benar	benar
19	Sampel 19	benar	 benar	 benar	benar	benar	benar
20	Sampel 20	benar	 benar	 benar	benar	benar	benar
jumlah benar :		**20**	**18**	**15**	**20**	**20**	**20**
akurasi		**100%**	**90%**	**75%**	**100%**	**100%**	**100%**

Dari Tabel 4.10 didaptkan hasil pengujian dimana akurasi untuk melihat ke tengah adalah 100%. Hal ini merujuk kepada percobaan sebelumnya bahwa arah tengah merupakan arah *idle* atau *default* yang digunakan oleh sistem sebagai patokan untuk mendeteksi arah-arah lainnya seperti kiri, kanan, bawah dan juga atas. Terlihat juga pada Tabel 4.10, arah bawah memiliki tingkat akurasi yang lebih rendah dibandingkan dengan arah-arah lainnya yaitu 75%, dikarenakan tingkat kesulitan sistem dalam mengklasifikasi antara arah tengah, bawah dan tertutup. Pada saat mata melihat kebawah, nilai EAR akan mengecil mendekati 0 dan juga jumlah pixel akan menurun. Hal tersebut yang meyebabkan sistem terkadang membacanya sebagai mata tertutup.

4.3.1. Precision dan Recall Metode Naïve Bayes Eye Sector Area (NBESA)

Pengujian *precision* dan *recall* bertujuan untuk mengukur kehandalan yang dimiliki oleh meteode yang diusulkan, pengujian dilakukan dengan mekanisme standarisasi jarak sebanyak 20 – 40 cm dengan kondisi pencahayaan sebesar antara 300 hingga 700 lux, nilai variable tersebut disesuaikan dengan pengujian pengaruh jarak, pencahayaan dan nilai batas ambang sebelumnya yang ditampilkan pada Tabel 4.8 dan Tabel 4.9.

Precision sendiri menggambarkan seberapa sering suatu arah pergerakan terdeteksi sebagai arah gerakan yang salah dalam tiap data uji. Semakin kecil nilai presisi maka arah gerakan tersebut lebih sering terdeteksi dengan arah gerakan yang lain menggunakan metode yang diusulkan ini. Sedangkan recall adalah tingkat keberhasilan deteksi tiap arah pergerakan bola mata dimana semakin tinggi nilai recall maka semakin optimal deteksi arah pergerakan bola mata dan kedipan mata menggunakan metode *Naïve Bayes Eye Sector Area* (NBESA).

Tabel 4.11 Hasil pengujian *multiclass classification Naïve Bayes Eye Sector Area (NBESA)*

actual	tengah	20	0	0	0	0	0	0
	kanan	0	20	0	0	0	0	0
	Kiri	0	0	20	0	0	0	0
	atas	1	0	0	18	1	0	0
	bawah	3	0	1	0	15	0	1
	tertutup	0	0	0	0	0	20	0
		tengah	kanan	kiri	atas	bawah	tertutup	Tidak Terdeteksi
		prediction						

Berdasarkan hasil pengukuran *multiclass classification* yang ditunjukan pada Tabel 4.11 maka dapat dihitung nilai *precision* dan *recall* yang ditampilkan pada Tabel 4.12 yang dicontohkan dalam perhitungan sebagai berikut :

$$Precision(atas) = \frac{TP}{TP + FP} = \frac{18}{18 + 0} = 100\%$$

$$Recall(atas) = \frac{TP}{TP + FN} = \frac{18}{18 + 2} = 90\%$$

Tabel 4.12 Hasil perhitungan *precision* dan *recall* metode *Naïve Bayes Eye Sector Area* (NBESA)

Gerakan	Precision	Recall
Atas	100%	90%
Bawah	93.75%	75%
Tengah	83.4%	90%
Kanan	100%	100%
Kiri	100%	100%
Tertutup	100%	100%

Pada tebel 4.12 diperoleh beberapa nilai yang merepresentasikan hasil perhitungan *precision* dan *recall* dari metode *Naïve Bayes Eye Sector Area* (NBESA). Dimana *precision* paling tinggi dimiliki oleh atas, kanan, kiri, dan tertutup sedangkan nilai *precision* paling rendah dimiliki oleh arah Tengah, ini menandakan bahwa arah tengah adalah arah yang paling sering terdeteksi ketika mata melihat kea rah atas dan bawah. Hal itu juga memberikan informasi bahwa pada tiap kegagalan pendeteksian arah lain terlebih yang berada pada sector yang sama (sector 2), maka hasil akan lebih cenderung menunjukan sistem mengklasifikasikan arah tersebut ke arah tengah.

Nilai recall yang paling besar adalah nilai recall yang dimiliki oleh arah tengah, kanan, kiri dan tertutup. Sedangkan nilai recall yang paling rendah dimiliki oleh arah bawah. Terlihat disini juga error paling tinggi dimiliki oleh arah bawah (Tabel 4.10) itu terjadi dikarenakan arah bawah mirip dengan arah tengah dan hampir mirip juga dengan mata tertutup. *Naïve Bayes* memang digunakan untuk mengatasi hal tersebut, namun seperti yang tampak pada tebel 5.7 dan 5.8 sistem tidak dapat mendeteksi secara akurat untuk kasus pendeteksian arah bawah, dan juga seperti yang ada pada Tabel 4.10 bahwa dari 20 data arah bawah, sebanyak 4 data terdeteksi sebagai arah tengah dan 1 data terdeteksi sebagai arah kiri. Banyak factor yang menyebabkan itu terjadi, factor yang paling mempengaruhi adalah factor registrasi, dimana pada saat registrasi user tidak atau kurang mengarahkan arah matanya ke bawah dan cenderung tidak sabar sehingga matanya bergerak ke tengah menatap arah computer. Factor lain yang mempengaruhi adalah factor kondisi lingkungan dan cahaya, dikarenakan sistem digunakan pada kondisi realtime dengan kondisi pencahayaan yang kadang berubah disebabkan oleh berpindahnya posisi kepala atau badan bahkan juga bayangan orang jalan disekitar menjadi sebuah kendala. Bayangan orang disekita akan mempengaruhi kondisi pencahayaan sehingga hasil *thresholding* akan berubah dari Gambar awal pada saat registrasi.

4.3.2. Perbandingan dengan metode lain

Ada banyak penelitian yang membahas tentang pendeteksian arah gerakan mata dengan menggunakan berbagai pendekatan dan berbagai metode. Namun kebanyakan dari metode yang diusulkan hanya dapat mendeteksi salah satu dari kondisi mata tersebut, yaitu deteksi gerakan bola mata atau deteksi gerakan kedipan mata. Terkait hal tersebut, maka dalam sub bab perbandingan metode ini, akan dibagi menjadi 2 mekanisme perbandingan yaitu perbandingan kemampuan deteksi gerakan bola mata dan perbandingan deteksi kedipan mata.

Untuk pergerakan bola mata, maka metode yang kami usulkan akan dibandingkan dengan metode Triangle similarity dan pixel value ditunjukan pada Tabel 4.13.

Tabel 4.13 Perbandingan metode usulan dengan metode lain dalam mendeteksi arah gerakan bola mata

Metode	Akurasi					Rata-rata
	Tengah	Kanan	Kiri	Atas	Bawah	
Pixel Value	-	100%	100%	-	-	100%
Triangle Similarity	-	90%	90%	75%	58%	78.25%
Naïve Bayes Eye Sector Area (metode usulan)	**100%**	**100%**	**100%**	**90%**	**75%**	**93%**

Pada Tabel 4.13 dapat diketahui bahwa metode usulan dapat secara akurat mendeteksi arah gerakan ke kanan, kiri dan tengah. Khusus untuk arah kanan dan kiri metode ini memiliki mekanimse yang sama dengan metode *Pixel Value* dimana inti dari metode tersebut adalah menjumlahkan pixel yang ada pada sector bagian kanan dan kiri, sedangkan metode usulan *Naïve Bayes Eye Sector Area* (NBESA) memanfaatkan hal yang sama dengan metode pixel value namun dengan improvisiasi membagi area menjadi 3 bagian. Sedangkan dalam mendeteksi arah bawah juga lebih tinggi dibanding dengan metode *Triangle Similarity,* hal ini sejalan dengan yang sudah dibahas pada subbab sebelumnya bahwa kelemahan metode ini adalah untuk mendeteksi arah bawah dikarenakan berbagai factor yang mempengaruhi.

Perbandingan selanjutnya adalah perbandingan metode dalam mendeteksi adanya kedipan, metode ini dibandingkan dengan metode HLVQ *(Haar Cascade Learning Vector Quantization)* dimana hasil dari perbandingan tingkat akurasi masing-masing metode ditunjukan pada Tabel 4.14.

Tabel 4.14 Perbandingan metode usulan dengan metode lain dalam mendeteksi kedipan mata

Metode	Kedipan	Gerakan Bola mata
HLVQ (*Haar Cascade Learning Vector Quantization)*	90.32%	-
NBESA (*Naïve Bayes Eye Sector Area)*	**100%**	**90%**

Tabel 4.15 Perbandingan metode usulan dengan metode deteksi gerakan bola mata dan metode deteksi kedipan

Metode	Akurasi					
	Tengah	Kanan	Kiri	Atas	Bawah	Tertutup
Pixel Value	-	v	v	-	-	-
Naïve Bayes Triangle Similarity	-	v	v	v	v	-
HLVQ (Haar Cascade Learning Vector Quantization)	-	-	-	-	-	v
Naïve Bayes Eye Sector Area (metode usulan)	**v**	**v**	**v**	**v**	**v**	**v**

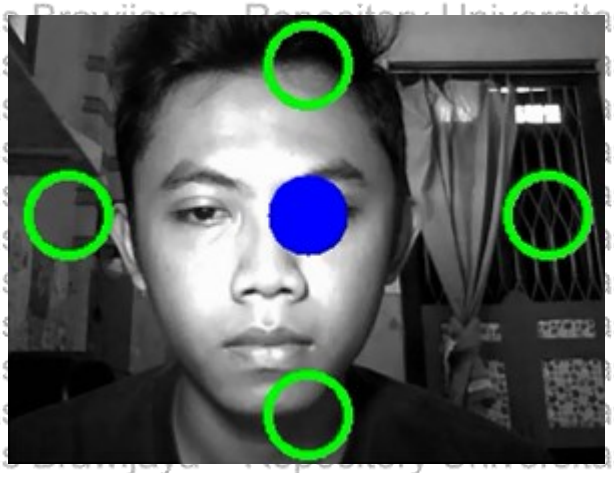
Pada Tabel 4.15 dapat dilihat bahwa metode usulan memiliki kemampuan dalam mendeteksi arah gerakan bola mata meliputi arah atas, bawah, tengah, kanan dan kiri juga sekaligus memiliki kemampuan dalam mendeteksi adanya gerakan

kedipan atau menutup mata. Banyak penelitian lain yang belum focus dalam meneliti masalah ini, banyak dari penelitian lain yang memisahkan antara deteksi gerakan bola mata dan gerakan kedipan mata, melalui metode yang kami usulkan ini maka seluruh kemampuan fungsi kerja mata seperti gerakan bola mata dan gerakan kelopak mata (kedipan) dapat dideteksi dan digunakan untuk kepentingan lain dengan hanya menggunakan 1 metode saja.

4.4. Integrasi metode NBESA pada tampilan layar berisi menu untuk deteksi gerakan bola mata dan kedipan mata

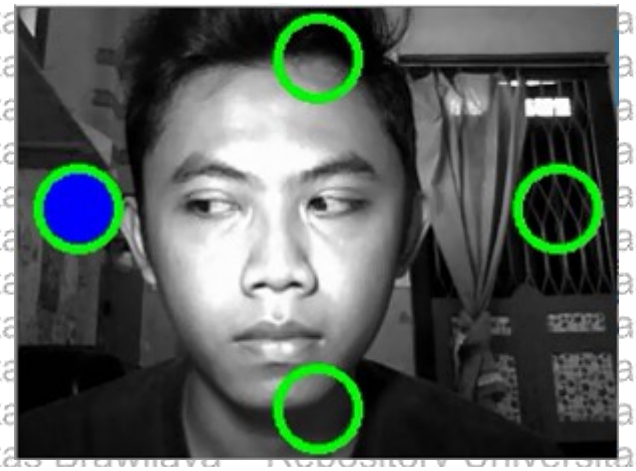
Dalam subbab sebelumnya telah dijelaskan bahwa metode NBESA (*Naïve Bayes Eye Sector Area)* memiliki tingkat akurasi sebesar 100% untuk kedipan mata dan 93% untuk deteksi arah gerakan bola mata, dalam hal ini metode NBESA akan diucicobakan secara langsung atau real time dengan memanfaatkan kamera dan cahaya external berupa lampu LED, penggunaan lampu LED disini dimaksudkan agar cahaya yang diterima oleh kamera memiliki nilai yang konstan sehingga tidak terjadi perubahan nilai intensitas cahaya yang dapat mengakibatkan perubahan nilai kontras yang menyebabkan terjadinya eror pada proses deteksi.

Sesuai dengan usulan rancangan tampilan pada BAB 3, bahwa terdapat 4 tombol berbentuk lingkaran yang memiliki fungsi masing-masing ketika tombol tersebut dieksekusi, tombol tersebut ditunjukan pada Gambar 4.2.



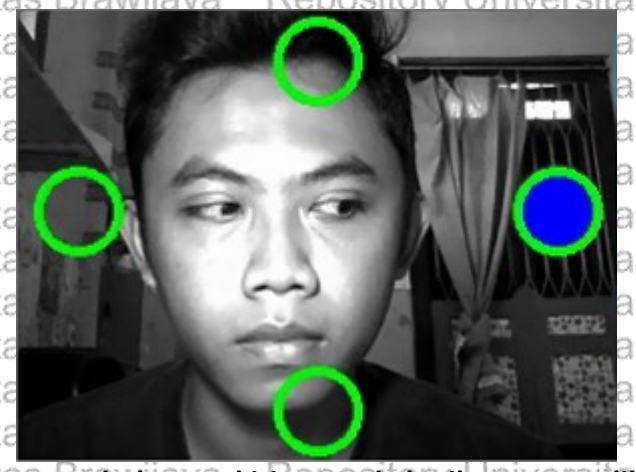
Gambar 4.2. 4 tombol pada arah kanan, atas, kiri dan bawah

Sesuai dengan rancangan pada bab 3, ketika mata melihat ke tengah atau dalam posisi idle maka sistem akan membaca arah gerakan terakhir yang dilakukan oleh mata, dalam Gambar 4.2 terdapat warna biru yang berada ditengah-tengah tombol menandakan bahwa tidak adanya gerakan yang sebelumnya dilakukan oleh mata sehingga ditandai dengan keluarnya lingkaran biru di tengah-tengah mata. Selanjutnya apabila mata bergerak ke arah kanan, kiri, atas ataupun bawah, maka tombol-tombol yang dituju tersebut akan menyala sesuai dengan arah gerakan mata yang ditunjukan oleh Gambar 4.3.

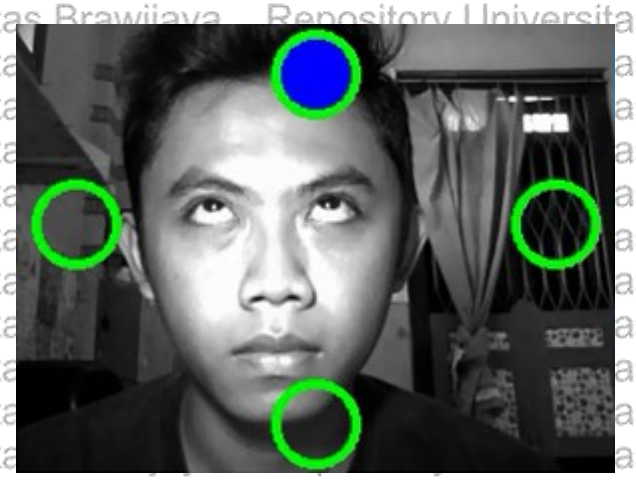


Gambar 4.3. Tombol hijau sebelah kanan menyala biru sesuai dengan arah gerakan mata

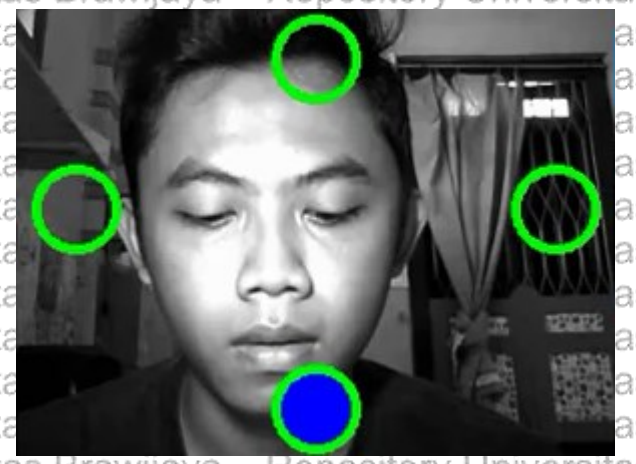
Pada Gambar 4.3 diatas, tombol pada area sebelah kanan akan menyala apabila mata melihat kearah kanan, begitupun dengan arah-arah yang lain yang akan menyala sesuai dengan arah gerakan bola mata (Gambar 4.4, 4.5, 4.6, 4.7).



Gambar 4.4. Tombol menu kiri menyala ketika mata melihat ke kiri

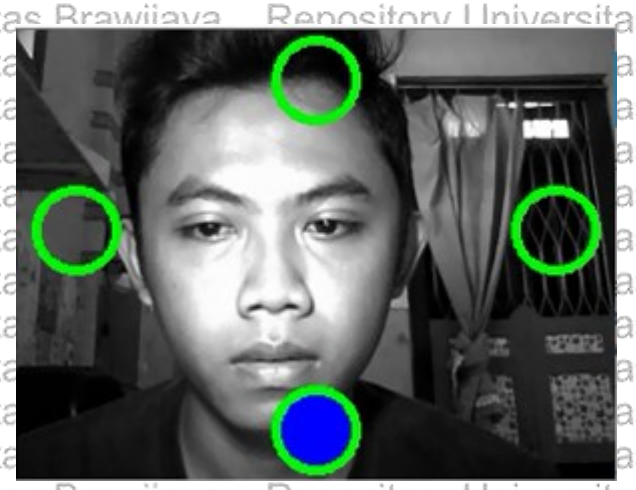


Gambar 4.5. Menu atas menyala ketika mata melihat ke atas



Gambar 4.6. Menu bawah menyala ketika mata melihat kearah bawah

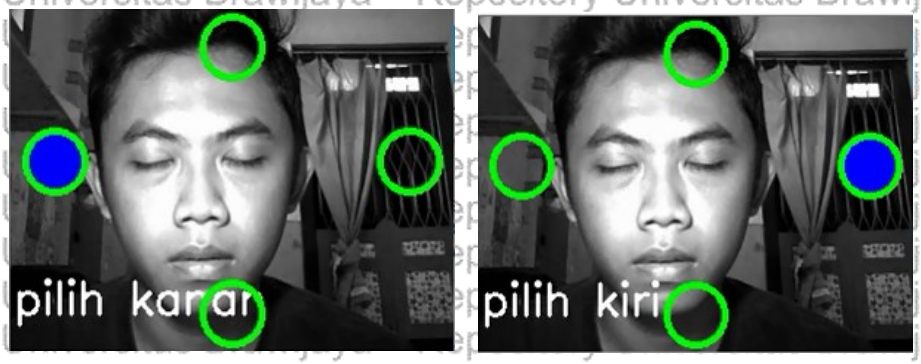
Setiap menu akan menyala berwarna biru apabila mata melihat ke arah yang sesuai dengan letak menu tersebut. Sedangkan apabila mata melihat kearah tengah maka sistem akan menyimpan arah terakhir gerakan bola mata yang ditandai tidak berubahnya menu seperti ditunjukan pada Gambar 4.7.



Gambar 4.7. Menu bawah tetap terpilih ketika mata melihat ke tengah, arah gerakan sebelumnya adalah arah bawah

Tujuan dari disimpannya menu terakhir sebelum mata melihat ke tengah adalah untuk memberikan kesempatan user untuk memilih menu lain ketika user melihat kearah depan atau melihat kearah layar menu navigasi.

Setelah user merasa menu yang dipilihnya telah cocok, maka untuk mengeksekusi menu tersebut dapat dilakukan dengan cara mengedipkan mata atau menutup mata sesaat dan sistem akan secara otomatis mengeksekusi pilihan menu user. Proses eksekusi menu tersebut ditunjukan pada Gambar 4.8.



Gambar 4.8. ketika mata menutup maka sistem akan* mengeksekusi *menu terakhir yang dipilih oleh mata sebelum

Dari percobaan integrasi sistem dengan metode usulan yang diintegrasikan dengan tampilan layar menu atraktif tersebut maka didapatkan sebuah statistic pada Tabel 5.16.

Tabel 4.16 Hasil akurasi deteksi program terhadap gerakan mata untuk pemilihan menu

Gerakan	Banyaknya Percobaan	Sukses Terdeteksi	Gagal Terdeteksi	Prosentase
Atas	4	4	0	**100%**
Bawah	14	7	7	**50%**
Kanan	7	7	0	**100%**
Kiri	4	4	0	**100%**
Tengah	6	6	0	**100%**
Tertutup / kedipan	9	7	2	**77.8%**
Rata-Rata				**87.9%**

Terlihat pada Tabel 5.16 bahwa hasil integrasi pada sebuah mekanisme pemilihan menu interaktif menunjukan rata-rata akurasi sebesar 87.9% dengan catatan bahwa prosentase paling rendah dimiliki oleh arah bawah. Hal ini sudah dapat diprediksi dikarenakan pada percobaan-percobaan sebelumnya (*accuration, precision* dan *recall)* arah bawah memiliki tingkat prosentase yang paling rendah dan cenderung terbaca sebagai arah "tengah".

Catatan : untuk hasil video dapat diakses melalui situs berbagi video Youtube pada link berikut :

1. https://youtu.be/QejIr0pDLpc
2. https://youtu.be/cLCkpe9ZW0Q

BAB 5
PENUTUP

Bab ini menjelaskan kesimpulan yang didapat dari perancangan dan hasil uji coba yang dilakukan dan telah dibahas pada BAB 4. Pada bab ini akan dibagi menjadi 2 bagian yaitu kesimpulan dari penelitian dan saran yang dapat dilakukan pada penelitian berikutnya.

5.1Kesimpulan

1. Hasil percobaan menunjukan bahwa deteksi area mata dapat dilkukan dengan baik dibuktikan dengan :
   a. Area mata dapat terdeteksi dengan baik pada user tanpa menggunakan kacamata dan cahaya yang stabil dengan akurasi sebesar 100% pada jarak 20, 30, 40, 50 dan 60 cm
   b. Area mata dapat terdeteksi cukup baik pada user yang menggunakan kacamata dan cahaya stabil dengan akurasi sebesar 100% hanya pada jarak 20, 30 dan 40 cm saja :
   c. Area mata tidak dapat dideteksi secara baik pada user dengan menggunakan kacamata dan cahaya stabil dengan akurasi sebesar 67% pada jarak 50 cm dan 34% pada jarak 60 cm.
   d. Area mata tidak dapat dideteksi pada jarak 10 cm dari kamera untuk kedua kelompok data dengan kacamata dan tanpa kacamata

   Dari hasil tersebut maka dapat disimpulkan bahwa area mata dapat terdeteksi dengan baik oleh pengguna kacamata ataupun non pengguna kacamata pada jarak 20 hingga 40 cm.
2. Menurut hasil percobaan yang dilakukan, sistem ini mampu bekerja dengan baik dan optimal untuk mendeteksi arah gerakan mata ke depan sebagai kondisi netral dan referensi dengan ketentuan :
   a. Cahaya dengan rentang antara 200 hingga 700 lux
   b. Nilai *threshold* sebesar 65
   c. Jarak yang mampu di akomodir oleh sistem ini ada di kisaran 20 hingga 40 cm
3. Hasil akurasi deteksi gerakan mata dan kedipan mata menunjukan akurasi sebesar 100% untuk arah tengah, 90% untuk arah atas, 75% untuk arah bawah, 100% untuk arah kiri, 100% untuk arah kanan dan 100% untuk kedipan pada percobaan *realtime* dengan menggunakan parameter yang telah dicari dan ditentukan sebelumnya yaitu cahaya sebesar 300 hingga 700 lux, jarak sebesar 20 hingga 40 cm dan nilai *threshold* sebesar 65.
4. Nilai *recall* dan *precision* yang dihasilkan oleh uji coba deteksi gerakan mata adalah sebagai berikut :
   a. Arah atas memiliki *recall* = 90% dan *precision* = 100%
   b. Arah bawah memiliki *recall* = 75% dan *precision* = 93.75%

c. Arah tengah memiliki *recall* = 90% dan *precision* = 83.4%
d. Arah kanan memiliki *recall* = 100% dan *precision* = 100%
e. Arah kiri memiliki *recall* = 100% dan *precision* = 100%
f. Gerakan kedipan memiliki *recall* = 100% dan *precision* = 100%

5. Dibandingkan dengan metode-metode lain, metode usulan ini memiliki keunggulan berupa :
   a. Akurasi yang diperoleh oleh metode usulan ini sebesar 100% berbanding 90% (metode Haar cascade Learning Vector Quantization) dalam mendeteksi adanya kedipan
      1. Metode pendekatan yang diusulkan memiliki rata-rata akurasi sebesar 93% untuk mendeteksi arah kanan, kiri, atas, bawah dan tengah dibandingkan dengan 78.25% (Triangle Similarity) yang hanya dapat mendeteksi arah kanan, kiri, atas dan bawah saja
   b. Metode usulan ini memiliki akurasi rata-rata sebesar 94.17% untuk mendeteksi arah kiri, kanan, atas, bawah, tengah dan juga kedipan sekaligus.
6. Hasil akurasi sistem setelah diintegrasikan terhadap mekanisme pemilihan menu secara interaktif menunjukan akurasi sebesar 87.9%.

5.2. Saran

1. Arah bawah memiliki tingkat akurasi yang paling rendah jika dibandingkan dengan arah-arah lain seperti atas, kanan, kiri dan tengah. Maka dari itu kedepan, penelitian harus bisa mengatasi hal ini dengan cara meningkatkan kemampuan sistem untuk dapat mendeteksi arah bawah dengan lebih akurat dan baik lagi.
2. Penggunaan kamera saja memang mampu untuk mendeteksi arah gerakan mata, namun perlu adanya sumber cahaya yang stabil sehingga cahaya tidak akan berubah-ubah. Kedepan, perlu adanya sebuah penelitian yang mampu mengatasi perubahan cahaya tersebut sehingga tidak diperlukan pemasangan lampu LED dibelakang kamera untuk menjaga kestabilan cahaya.
3. Pergerakan bola mata yang dapat diimplementasikan sebagai media kendali suatu perangkat elektrik/otomatis membutuhkan tipe arah pergerakan yang lebih beragam seperti arah serong, sehingga tipe arah pergerakan selain kanan, kiri, atas, dan bawah dapat diterapkan pada deteksi pergerakan bola mata.

DAFTAR PUSTAKA

Amin, M. F., & Wahono, R. S. (2015). Penerapan Reduksi Region Palsu Berbasis Mathematical Morphology pada Algoritma Adaboost Untuk Deteksi Plat Nomor Kendaraan Indonesia. *Journal of Intelligent Systems*, *1*(1), 9–14.

Catalbas, M. C., Cegovnik, T., Sodnik, J., & Gulten, A. (n.d.). Driver Fatigue Detection Based on Saccadic Eye Movements, 913–917.

Chen, W., Ning, H., & Li, J. (2010). Detecting and tracking of small moving target in avian radar images. *IASP 10 - 2010 International Conference on Image Analysis and Signal Processing*, 473–476. https://doi.org/10.1109/IASP.2010.5476073

Gupta, M. V., & Sharma, D. (2014). A Study of Various Face Detection Methods, *3*(5), 3–6.

Haq, Z. A., & Hasan, Z. (2017). Eye-blink rate detection for fatigue determination. *India International Conference on Information Processing, IICIP 2016 - Proceedings*. https://doi.org/10.1109/IICIP.2016.7975348

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 583–596. https://doi.org/10.1109/TPAMI.2014.2345390

Iveson-Iveson, J. (1983). [Anatomy and physiology of the eye]. *Xianggang Hu Li Za Zhi. The Hong Kong Nursing Journal*, (35), 47–50.

Jiang, B., Zhang, J., Deng, B., Guo, Y., & Liu, L. (2017). Deep Face Feature for Face Alignment, 1–11. Retrieved from http://arxiv.org/abs/1708.02721

Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit Andi.

Kazemi, V., & Sullivan, J. (2017). One Millisecond Face Alignment with an Assemble of Regression Trees. https://doi.org/10.13140/2.1.1212.2243

Kemenkes RI. (2017). Infografis Hari Disabilitas. Retrieved from https://data.go.id/konten/visualization/infografis-hari-disabilitas/

Kementrian Kesehatan RI. (2014). Situasi Penyandang Disabilitas. *Buletin Jendela Data & Informasi Kesehatan*, *Semester 2*(1), 1–5. https://doi.org/10.1007/s13398-014-0173-7.2

Kim, E. Y. K. E. Y., Kang, S. K. K. S. K., Jung, K. J. K., & Kim, H. J. K. H. J. (2005). Eye mouse: mouse implementation using eye tracking. *2005 Digest of Technical Papers. International Conference on Consumer Electronics, 2005. ICCE.*, 207–

208. https://doi.org/10.1109/ICCE.2005.1429790

Manchanda, K., & Bing, B. (2010). Advanced mouse pointer control using trajectory-based gesture recognition. *Conference Proceedings - IEEE SOUTHEASTCON*, 412–415. https://doi.org/10.1109/SECON.2010.5453841

Ortiz, K. J. P., Arciaga, J. M. R., Escusa, K. M., & Ubales, S. Y. (2017). Microcontroller based foot-operated mouse using a sensor pad as an alternative to a conventional mouse. *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016*, 1434–1437. https://doi.org/10.1109/IMCEC.2016.7867451

Patel, S. N., & Prakash, V. (2015). Autonomous camera based eye controlled wheelchair system using raspberry-pi. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 3–8. https://doi.org/10.1109/ICIIECS.2015.7192876

Prasetya, R. P., & Utaminingrum, F. (2017). Triangle similarity approach for detecting eyeball movement. *5th International Symposium on Computational and Business Intelligence, ISCBI 2017*, 37–40. https://doi.org/10.1109/ISCBI.2017.8053540

Robson, K. S. (1967). the Role of Eye-To-Eye Contact in Maternal-Infant Attachment. *Journal of Child Psychology and Psychiatry*, *8*(1), 13–25. https://doi.org/10.1111/j.1469-7610.1967.tb02176.x

Shih, F. Y. (2017). *Image Processing and Mathematical Morphology: Fundamentals and Applications*. CRC press.

Siripitakchai, A., & Phimoltares, S. (2017). Eye-Captcha : An Enhanced Captcha Using Eye Movement.

Sotelo, D. I., Alberto, J., Benitez, P., Hiram, J., Hernández, E., & Collection, A. D. (2018). Identification and classification of eyes movement using EEG signals, 25–30.

Uddin, M. S., & Akhi, A. Y. (2016). Horse Detection Using Haar Like Features. *International Journal of Computer Theory and Engineering*, *8*(5), 415–418. https://doi.org/10.7763/IJCTE.2016.V8.1081

Utaminingrum, F., Fauzi, M. A., & Sari, Y. A. (n.d.). Eye Movement as Navigator for Restricted Disabled Person in Handling Position, 1–5. https://doi.org/10.475/123

Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1,

p. I--511).

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154.

Zhang, T. hao, & Tang, C. W. (2018). Multiple-target tracking on mixed images with reflections and occlusions. *Journal of Visual Communication and Image Representation*, *52*(December 2016), 45–57. https://doi.org/10.1016/j.jvcir.2018.02.001