

# Pupilometri Dinamis untuk Mengukur Respons Pupil sebagai Pendeteksi Dini Demensia pada Lansia

CELINE GABRIELLA WAHYUDI, LUKAS, NOVA EKA BUDIYANTA

Cognitive Engineering Research Group (CERG), Universitas Katolik Indonesia Atma Jaya, Indonesia  
Email: [lukas@atmajaya.ac.id](mailto:lukas@atmajaya.ac.id)

Received 4 April 2023 | Revised 4 Mei 2023 | Accepted 20 Mei 2023

## ABSTRAK

*Pupilometri merupakan metode pengukuran respons pupil terhadap stimulus. Kemampuan pupil mata dalam merespons cahaya diamati melalui pupillary light response (PLR). Penelitian mendapati PLR pasien demensia berbeda dengan pasien normal. Penelitian ini bertujuan merancang algoritma computer vision yang dapat mendeteksi pupil secara akurat, menampilkan respons pupil terhadap cahaya dalam bentuk grafik dan PLR pada sebuah aplikasi desktop, yang mengendalikan goggles berisi rangkaian kamera, pencahayaan, dan sensor jarak VL53L0X. Rekaman diproses dengan Local Binary Pattern (LBP) dan deteksi kontur untuk mendeteksi pupil. Data pengukuran diproses dan disimpan pada basis data lokal dan aplikasi web, sehingga tenaga medis dapat menentukan ada atau tidaknya gejala demensia pada pasien lansia. Tingkat ketelitian algoritma pengukuran pupil sebesar 73,33% yang didapatkan dari 30 kali pengujian.*

**Kata kunci:** *computer vision, demensia, deteksi dini, pupillary light response, pupilometri*

## ABSTRACT

*Pupillometry is a method of measuring the pupil's response towards stimulus. Pupil response to light is observed through pupillary light response (PLR). Research found that PLR values of patients suffering from dementia differ from that of normal patients. This study implements a computer vision algorithm that accurately detects the pupil, calculates, and shows its response towards light in graphs and PLR values on a desktop application which controls goggles that contain a camera, lighting setup, and the VL53L0X distance sensor. Video is processed using Local Binary Pattern (LBP) and contour detection to detect the pupil. Results are processed and saved in the local and web database, so experts can determine the presence of dementia symptoms in the elderly patient. The accuracy of the pupil detection algorithm is 73,33%, as obtained from 30 tests.*

**Keywords:** *computer vision, dementia, early detection, pupillary light response, pupillometry*

## 1. PENDAHULUAN

Pupilometri merupakan metode non-invasif pengukuran respons pupil mata terhadap sebuah stimulus. Dalam bidang neurologi, pupilometri digunakan untuk menguji keadaan kognitif pasien, terutama setelah trauma atau cedera. Selain itu, sejumlah penelitian menunjukkan bahwa pasien penderita demensia memiliki respons pupil mata lebih lambat terhadap cahaya dibandingkan dengan pasien normal **(Kerr, dkk, 2016)**.

Demensia merupakan sebuah sindrom yang menyebabkan penderitanya mengalami penurunan fungsi kognitif, yakni penurunan dalam cara berpikir, daya ingat, serta kemampuan berbahasa yang melampaui usia wajar. Walaupun demensia cenderung mempengaruhi penduduk lansia, demensia bukan merupakan suatu hal yang wajar dalam proses penuaan **(Sandilayan dan Dening, 2015)**.

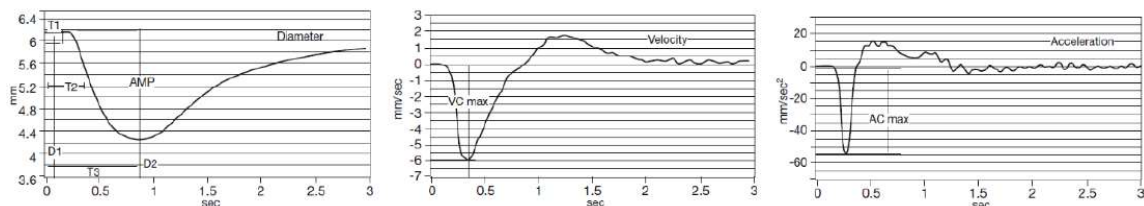
Badan Pusat Statistik (BPS) menyatakan bahwa hingga tahun 2020, persentase penduduk lanjut usia (lansia) telah meningkat sekitar dua kali lipat dalam waktu lima dekade terakhir, yakni dengan jumlah 9,92% dari total penduduk Indonesia. Namun, penduduk kategori lansia memiliki risiko yang lebih tinggi terhadap penyakit demensia **(Badan Pusat Statistik, 2020)**.

Secara tradisional, penilaian respons pupil dilakukan oleh dokter dengan cara menyinarkan cahaya pada mata pasien menggunakan senter, dan evaluasi diberikan berdasarkan pengamatan dokter itu sendiri. Tetapi, sejumlah penelitian telah menemukan bahwa metode ini cenderung subjektif dan tidak konsisten.

Tujuan dari penelitian ini adalah merancang algoritma dari sebuah aplikasi desktop pupilometri yang mampu mendeteksi pupil mata secara akurat, serta mampu mengalkulasi respons pupil mata terhadap cahaya. Hasil yang diberikan berupa nilai-nilai *pupillary light response* (PLR) serta respons pupil terhadap cahaya dalam bentuk grafik. Data tersebut kemudian dapat dianalisis lebih lanjut oleh seorang tenaga medis untuk meneliti ada atau tidaknya gejala demensia pada pasien lansia.

### 1.1 Pupillary Light Response

Pemberian cahaya pada mata dapat mempengaruhi ukuran pupil. Kemampuan manusia dalam merespons cahaya tersebut dipantau melalui nilai-nilai *pupillary light response* (PLR), yang memberikan penilaian fisiologis yang objektif akan fungsi sistem saraf. Fotiou, dkk **(Hall dan Chilcott, 2018)** telah menemukan bahwa penyakit *Alzheimer* dapat mempengaruhi hasil PLR. Menurut Fotiou, terdapat tiga buah grafik indikator respons pupil, yakni grafik diameter, kecepatan, serta akselerasi seperti tertera pada Gambar 1.



Gambar 1. Grafik (a) Diameter, (b) Kecepatan, dan (c) Akselerasi

### **1.2 Local Binary Pattern**

*Local Binary Pattern* (LBP) merupakan sebuah deskriptor visual yang digunakan untuk klasifikasi. LBP bekerja dengan cara memberikan label berupa angka pada tiap piksel dari suatu citra digital berdasarkan nilai intensitas warna setelah pemberian *grayscale* serta threshold (**Kushairy, dkk, 2014**).

### **1.3 Cascade Classifier**

*Cascade Classifier* merupakan sekumpulan pengklasifikasi yang berfungsi melakukan deteksi objek. Sebuah pengklasifikasi dilatih (*training*) menggunakan dasar-dasar *machine learning* menggunakan gambar positif dan negatif agar dapat membedakan sebuah objek. *Training* tersebut dapat dilakukan menggunakan deskriptor visual Haar maupun LBP. Saat *training*, fitur-fitur dari tiap gambar tersebut diambil berdasarkan fitur yang menyerupai deskriptor visual. Konsep *cascade* berakar dari penerapan filter berupa pengklasifikasi yang bertahap untuk melakukan segmentasi dan mendapatkan *Region of Interest* (ROI). Tiap sub-bagian pada sebuah gambar masukan akan melalui tiap tahapan dari *cascade classifier* tersebut. Bila sub-bagian gambar berhasil melewati suatu tahap, sub-bagian gambar tersebut akan diteruskan pada classifier berikutnya untuk menjadi kandidat sebuah ROI. Pada penelitian ini, *cascade classifier* digunakan untuk lokalisasi, atau menemukan ROI yakni bagian pupil mata dari gambar yang diberikan. *Cascade classifier* dilatih menggunakan *Cascade Trainer GUI*, yang merupakan aplikasi desktop berbasis OpenCV yang diciptakan oleh Amin Ahmadi (**Setjo dan Faridah, 2017**) (**Phase, 2020**).

### **1.4 DetectMultiScale**

*DetectMultiScale* merupakan sebuah fungsi yang digunakan untuk mendeteksi objek berdasarkan pengklasifikasi yang diberikan kepadanya. *DetectMultiScale* bekerja dengan cara mengambil sebuah gambar masukan dan menciptakan sebuah piramida skala. Piramida skala merupakan perwakilan dari gambar tersebut dalam berbagai skala, dan bertujuan agar deteksi tetap dapat dilakukan tanpa bergantung pada ukuran objek deteksi. Pada fungsi *detectMultiScale*, Parameter *minNeighbors* berfungsi menentukan seberapa banyak kandidat objek yang didapatkan agar objek dapat dianggap sebagai sebuah deteksi. Parameter *minSize* dan *maxSize* ditentukan berdasarkan kemungkinan ukuran objek deteksi (**Rosebrock, 2022**).

### **1.5 Canny Edge Detection**

*Canny edge detection* merupakan sebuah fungsi dari OpenCV untuk mendapatkan garis tepi dari sebuah gambar. Sebelum melakukan *Canny edge detection*, terdapat serangkaian langkah pengolahan citra yang harus dilakukan pada gambar tersebut, seperti pemberian grayscale dan blur (**Bradski, 2017**).

### **1.6 Contour Detection**

*Contour Detection* adalah sebuah fungsi dari OpenCV yang berguna mencari kontur sebuah gambar. Sebuah kontur diartikan sebagai kurva yang menghubungkan semua titik kontinu pada sebuah objek dengan warna dan intensitas yang sama. Luas kontur tersebut dikalkulasi berdasarkan Teorema Green dan dinyatakan dalam satuan piksel (**Bradski, 2017**).

### **1.7 Sensor Jarak VL53L0X**

VL53L0X merupakan modul sensor jarak berbasis *Time-of-Flight* (ToF) yang menggunakan cahaya laser IR dengan panjang gelombang 940 nm. Cahaya yang dipancarkan bersifat koheren, dan memiliki sudut pancaran sebesar 25°, sehingga cahaya dapat difokuskan pada satu bidang tanpa dipantulkan pada bidang lain. Sensor ini mampu mengukur jarak hingga 2 meter (**Laković, dkk, 2019**).

### 1.8 Regresi Linier

Regresi linier merupakan sebuah metode statistika untuk mendapatkan model linier yang mampu mewakili hubungan yang paling optimal antara data dari dua variabel. Pada sebuah regresi linier sederhana, ditentukan pendekatan linier dari variabel dependen  $y$  terhadap variabel independen  $x$  (Verbeek, 2017).

### 1.9 Moving Average

*Moving average* merupakan metode untuk mengalkulasi nilai rata-rata *subset* data (*window*) dari kumpulan data lengkap. *Moving average* umumnya digunakan untuk menghilangkan data fluktuatif dan *noise*, serta mempermudah analisis grafik (Hansun, 2013).

## 2. METODE

### 2.1 Konsep Sistem Pupilometri Keseluruhan

Secara keseluruhan, sistem pupilometri ini merupakan perkembangan dari sepasang *goggles* yang berisi rangkaian LED dan webcam yang berfungsi memberi stimulus cahaya dan merekam respons pupil. *Goggles* ini dikendalikan oleh aplikasi *desktop* yang dapat mendeteksi, mengukur, mengalkulasi, serta menampilkan respons pupil terhadap cahaya. Gambar 2 menunjukkan diagram keseluruhan sistem pupilometri.



Gambar 2. Diagram Blok Sistem Pupilometri

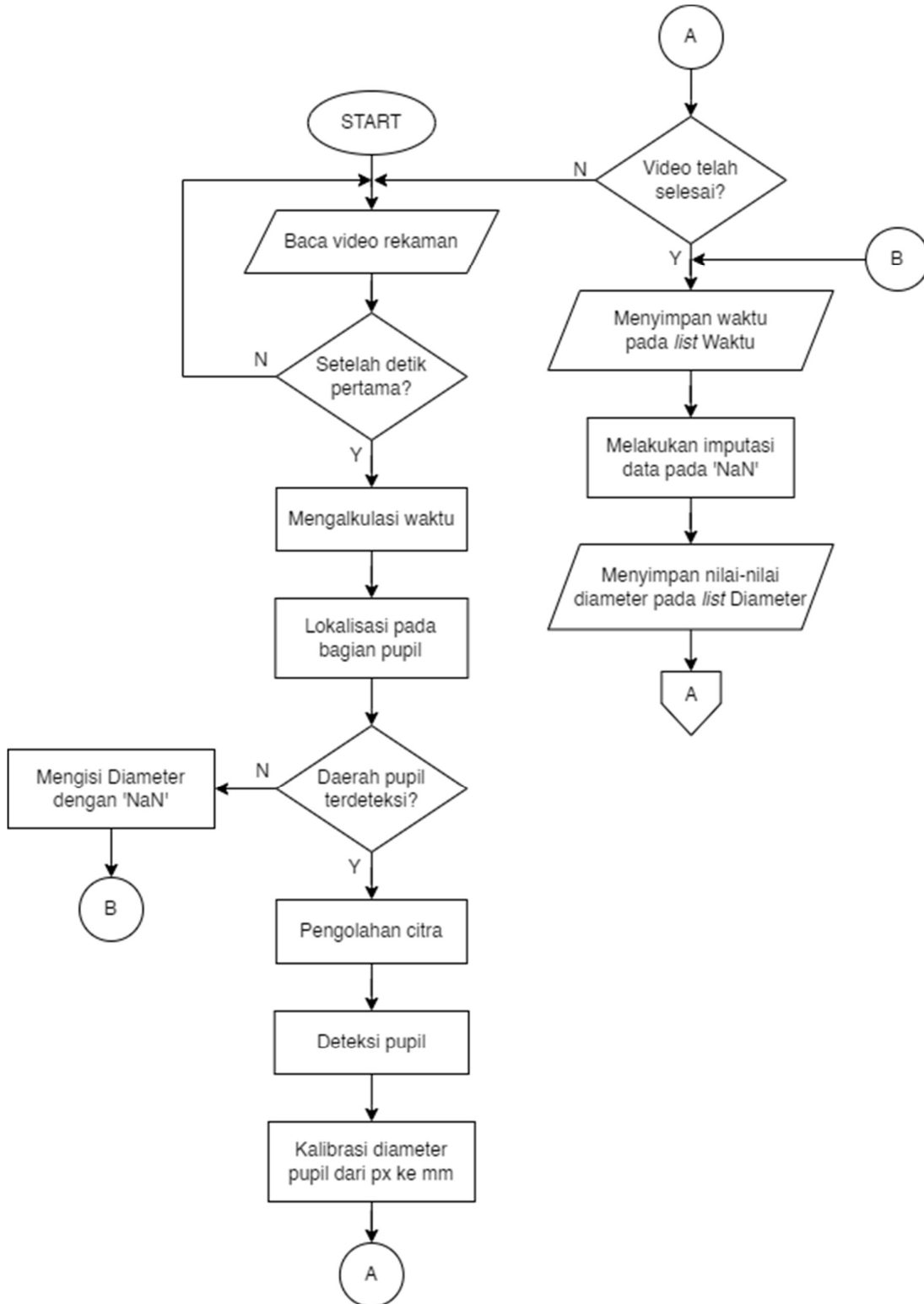
### 2.2 Cara Kerja Deteksi dan Pengukuran Pupil Mata

Dari rekaman yang diberikan, program akan melakukan lokalisasi untuk mencari ROI yakni bagian pupil. Pengolahan citra diberikan pada bagian tersebut untuk mendapatkan kontur pupil. Dari tiap kontur lingkaran yang didapat, program akan menyimpan nilai diameter dari lingkaran tersebut pada sebuah list, beserta dengan waktu dalam satuan detik. Apabila lokalisasi gagal dilakukan, program dapat melakukan imputasi data, yakni menggantikan data yang hilang. Alur ini dapat dilihat dalam diagram alir yang ditunjukkan pada Gambar 3.

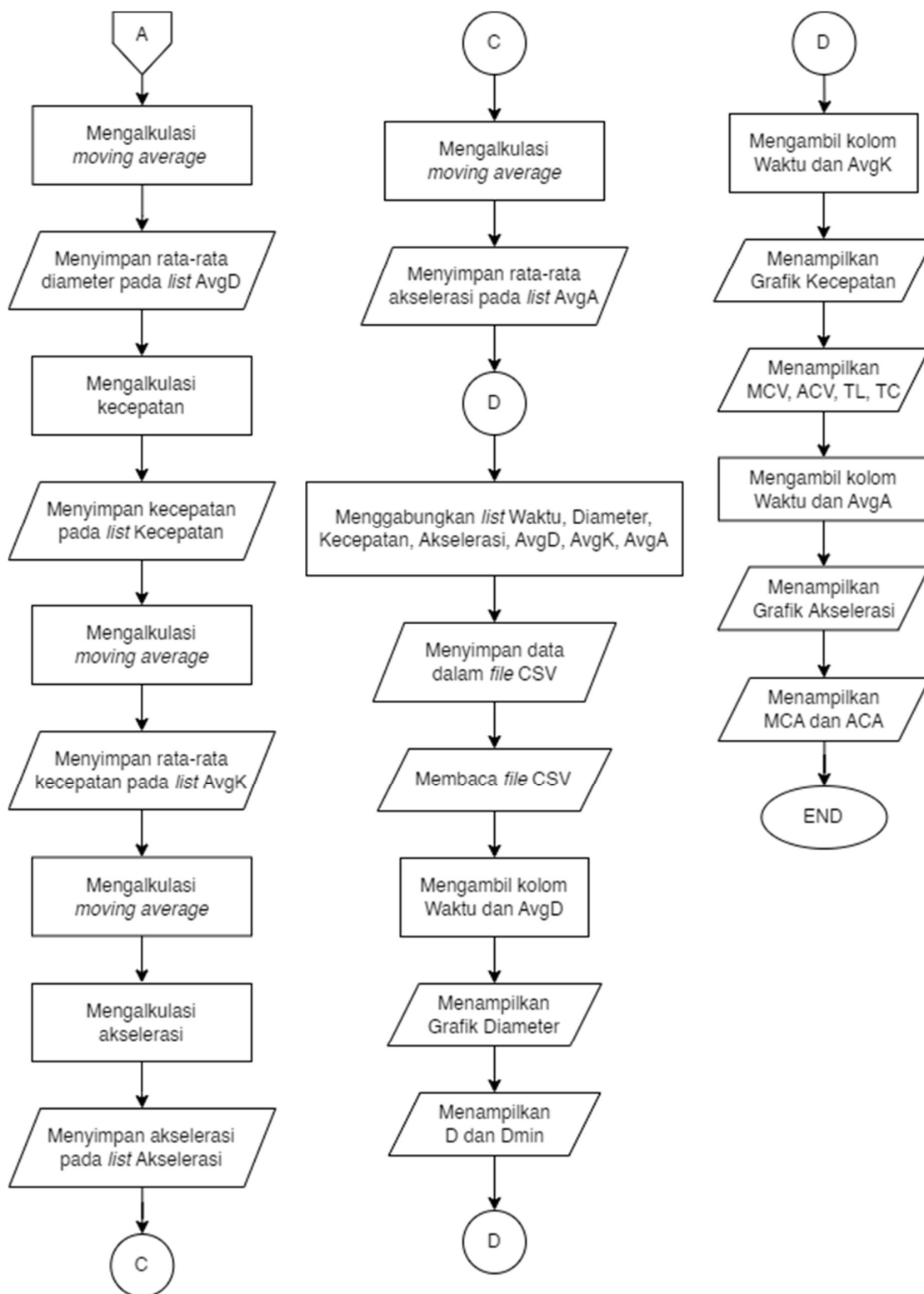
### 2.3 Cara Kerja Pemrosesan dan Analisis Data Pengukuran

List diameter yang telah dibuat akan diterapkan *moving average* dan disimpan pada list 'AvgD'. Dari list tersebut, program akan menghitung nilai kecepatan, menerapkan *moving average*, dan menyimpannya pada list 'AvgK'. Nilai akselerasi dihitung, diterapkan *moving average* dan disimpan pada list 'AvgA'. Semua list akan digabung dan disimpan pada sebuah file format CSV. Program akan membaca file tersebut dan mengambil kolom 'Waktu' dan 'AvgD' untuk diproses menjadi grafik diameter serta mendapatkan nilai D dan Dmin. Kolom 'Waktu' dan 'AvgK' diambil untuk diproses menjadi grafik kecepatan serta mendapatkan nilai MCV, ACV, TL dan TC. Kolom 'Waktu' dan 'AvgA' diambil untuk diproses menjadi grafik akselerasi serta mendapatkan nilai MCA dan ACA. Alur ini dapat dilihat pada sebuah diagram alir yang ditunjukkan oleh Gambar 4.

Perancangan Algoritma Pupilmetri untuk Mengukur Respons Pupil Mata terhadap Cahaya sebagai Pendeteksi Dini Demensia pada Lansia



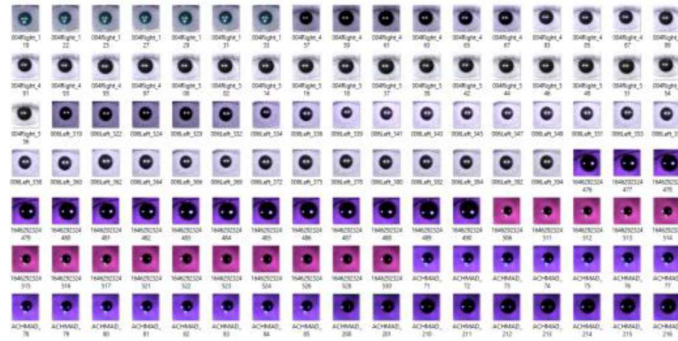
Gambar 3. Diagram Alir Deteksi Pupil dan Pengukuran



**Gambar 4. Diagram Alir Pemrosesan dan Analisis Data Pengukuran**

## 2.4 Perancangan Pengklasifikasi dan Lokalisasi Pupil

Lokalisasi dilakukan untuk mengabaikan bagian-bagian lain pada tangkapan video yang tidak diinginkan untuk terjadi deteksi. Lokalisasi ini dilakukan menggunakan pengklasifikasi yang telah diberikan *training*. Training dilakukan menggunakan gambar positif, yakni bagian pupil pada berbagai macam kondisi pencahayaan, serta gambar negatif, seperti seperti alis, kulit, bulu mata, serta kelenjar air mata. Gambar 5 menunjukkan cuplikan *dataset* positif. Gambar 6 menunjukkan cuplikan *dataset* negatif.

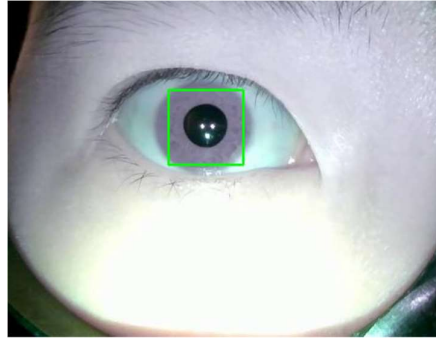


Gambar 5. Cuplikan *Dataset* Positif



Gambar 6. Cuplikan *Dataset* Negatif

*Cascade Classifier* dirancang menggunakan *Cascade Trainer GUI* dengan 450 gambar positif dan 285 gambar negatif. *Training* dilakukan sebanyak 13 tahap, dengan ukuran sampel 18x18 piksel, dan jenis fitur LBP. Target *training* yang ingin dicapai adalah tingkat akurasi minimum sebesar 99,5% atau kesalahan maksimum sebesar 0,5%. Ketika tombol 'Start' ditekan, program akan menjalankan perintah *opencv\_createsamples*, yakni membuat *dataset* sampel positif dalam bentuk biner pada sebuah *file* format VEC yang dapat dibaca oleh program saat training. Sampel yang dihasilkan berupa distorsi dan rotasi pada gambar positif yang ditempatkan secara acak di atas gambar negatif. Setelah pembuatan sampel selesai, dilakukan *training* dengan fungsi *opencv\_traincascade*. *Opencv\_traincascade* kemudian mengambil *file* format VEC tersebut dan melakukan *training*. Keluaran dari hasil *training* ini berupa *file* format XML yang dapat diunggah pada fungsi *detectMultiScale* pada OpenCV. Fungsi *detectMultiScale* mengambil rekaman masukan yang telah dijadikan *grayscale* dan membuat piramida skala dengan skala sebesar 3%. Dibutuhkan 6 kandidat deteksi di sekitar ROI untuk terjadi deteksi. Ukuran deteksi minimum yang diizinkan sebesar 70x70 piksel dan maksimum 120x120 piksel. Hasil lokalisasi pada tiap frame akan diberi *bounding box* seperti yang ditunjukkan pada Gambar 7.



**Gambar 7. Hasil Lokalisasi**

### **2.5 Perancangan Algoritma Pendeteksi Kontur Pupil**

Dari hasil lokalisasi tersebut, diterapkan pengolahan citra yakni filter *grayscale*, *blur*, dan *adaptive thresholding*. Dari pemberian *thresholding* diberikan *Canny edge detection* untuk mendapatkan garis tepi dari pupil. Program kemudian menggunakan *contour detection* untuk mencari kontur terbesar dan menandainya dengan sebuah lingkaran seperti pada Gambar 8.



**Gambar 8. Hasil Deteksi Kontur Pupil**

### **2.6 Kalibrasi Nilai Piksel ke Milimeter**

Diameter dari hasil deteksi kontur tersebut dinyatakan dalam satuan piksel, sehingga perlu dilakukan kalibrasi untuk menyesuaikan nilai tersebut dalam satuan dunia nyata (3D) milimeter. Nilai piksel dalam satu milimeter berkaitan erat dengan jarak antara kamera dengan benda. Semakin jauh jarak antara benda dengan kamera, maka benda akan terlihat semakin kecil. Sehingga, dibuat semacam lookup table untuk semua kemungkinan angka pada semua kemungkinan jarak yang dapat diperlukan selama deteksi. Tabel ini diperoleh dengan cara mengambil gambar suatu benda referensi (dalam hal ini sebuah lingkaran berwarna hitam) dalam milimeter pada rentang jarak 85–103 milimeter. Pengambilan gambar dilakukan dengan jarak antara kamera dengan objek digeser setiap 2 milimeter. Gambar-gambar benda referensi tersebut kemudian dimasukkan ke dalam sebuah program yang dapat mendeteksi lingkaran serta menghitung diameternya. Maka nilai yang ingin dicari adalah rasio piksel dalam satu milimeter. Rasio ini didapatkan dari ukuran diameter dalam satuan milimeter yang dibagi dengan diameter kontur dalam satuan piksel. Benda referensi utama yang digunakan merupakan lingkaran dengan diameter 10 milimeter dan digunakan sebagai variabel pengali untuk lingkaran dengan diameter yang lebih kecil agar error yang dihasilkan tidak besar. Gambar 9 menunjukkan contoh perolehan nilai referensi kamera kiri.



Perancangan Algoritma Pupilometri untuk Mengukur Respons Pupil Mata terhadap Cahaya sebagai Pendeteksi Dini Demensia pada Lansia



**Gambar 9. Contoh Perolehan Nilai Referensi Kamera Kiri**

Dari nilai diameter yang diperoleh dalam piksel, didapatkan rasio piksel ke milimeter dengan cara membagi diameter piksel dengan diameter milimeter. Untuk selanjutnya, nilai ini akan disebut sebagai rasio pengali, karena akan digunakan sebagai tolok ukur. Tabel 1 menunjukkan tabel rasio untuk jarak 10 milimeter.

**Tabel 1. Tabel Rasio untuk Jarak 10 Mm pada Kamera Kiri**

| <i>Diameter (px)</i> | <i>Jarak (mm)</i> | <i>Rasio Pengali</i> |
|----------------------|-------------------|----------------------|
| 114,341              | 85                | 0,088                |
| 111,194              | 87                | 0,089                |
| 109,388              | 89                | 0,091                |
| 107,184              | 91                | 0,093                |
| 105,144              | 93                | 0,095                |
| 102,973              | 95                | 0,097                |
| 100,906              | 97                | 0,099                |
| 98,921               | 99                | 0,101                |
| 96,953               | 101               | 0,103                |
| 94,983               | 103               | 0,105                |

Ukuran pupil dapat bervariasi dari 2 hingga 8 milimeter, maka diperlukan nilai-nilai rasio pada ukuran-ukuran tersebut pada tiap jarak. Gambar 10 menunjukkan contoh pengambilan gambar lingkaran dengan diameter 2 hingga 9 milimeter.



**Gambar 10. Contoh Pengambilan Gambar Lingkaran Berdiameter 2-9 Milimeter**

Dari nilai rasio yang diperoleh, didapatkan *normalized ratio*, yakni rasio dikalikan dengan rasio pengali pada jarak tersebut. Tabel 2 menunjukkan tabel rasio tangkapan kamera kiri pada jarak 85 milimeter.

**Tabel 2. Tabel Perbandingan Tangkapan Kamera Kiri pada Jarak 85 mm**

| <i>Diameter (mm)</i> | <i>Diameter (px)</i> | <i>Rasio</i> | <i>Normalized Ratio</i> |
|----------------------|----------------------|--------------|-------------------------|
| 2                    | 24,880               | 0,080        | 2,176                   |
| 3                    | 37,492               | 0,080        | 3,278                   |
| 4                    | 44,859               | 0,089        | 3,923                   |
| 5                    | 57,974               | 0,086        | 5,070                   |
| 6                    | 72,251               | 0,083        | 6,319                   |
| 7                    | 81,558               | 0,086        | 7,133                   |
| 8                    | 90,139               | 0,089        | 7,883                   |
| 9                    | 102,409              | 0,088        | 8,957                   |

Berdasarkan nilai-nilai tersebut, dikalkulasi regresi linier antara diameter dalam milimeter dan *normalized ratio*. Hal ini diterapkan pada tiap ukuran lingkaran dari 2 hingga 9 milimeter untuk tiap jarak yang telah ditentukan. Karena terdapat dua buah kamera dalam *goggles*, masing-masing dengan pengaturan *focal length* yang berbeda, maka proses kalibrasi yang sama dilakukan pada masing-masing kamera.

### 2.7 Kalibrasi Sensor Jarak VL53L0X

Kalibrasi ini dilakukan untuk meningkatkan akurasi pembacaan sensor, dengan cara mengambil rata-rata pembacaan sensor sebanyak 40 kali pada jarak-jarak tertentu. Setelah pengambilan data, dihitung model regresi linier untuk mencari keluaran ideal, yakni keluaran yang sesuai dengan standar pengukuran (dalam hal ini sesuai ukuran pada penggaris). Karena sensor ini akan digunakan dalam kondisi gelap (bagian dalam *goggles*), maka kalibrasi juga dilakukan dalam kondisi ruangan gelap. Adapun karena pantulan cahaya inframerah sensor akan mengenai daerah di bawah mata, maka kalibrasi dilakukan pada benda dengan permukaan yang menyerupai bagian bawah mata yang diletakkan di depan sensor.

### 2.8 Imputasi Data

Terdapat kemungkinan pasien berkedip selama pengujian berlangsung yang menyebabkan bagian pupil tidak terdeteksi. Apabila ini terjadi, data akan diisi dengan 'NaN', yakni *Not a Number*. Sehingga, dilakukan interpolasi untuk mengganti nilai-nilai tersebut, yakni pengisian data yang hilang di antara dua buah titik data. Titik data pertama adalah data sebelum nilai pertama 'NaN' dan titik data kedua adalah data pertama setelah nilai 'NaN' terakhir.

### 2.9 Penyimpanan Data Pengukuran dalam File Format CSV

*File* dengan format *Comma Separated Values* (CSV) digunakan untuk menampung data hasil pengukuran dan kalkulasi respons pupil. Terdapat 7 kolom pada file ini, yakni 'Waktu', 'Diameter', 'Kecepatan', 'Akselerasi', 'AvgD', 'AvgK', dan 'AvgA'.

### 2.10 Pembuatan Grafik Diameter, Kecepatan, dan Akselerasi

Program akan membaca *file* format CSV yang berisi data hasil deteksi dan mengambil kolom-kolom 'Waktu', 'AvgD', 'AvgK', serta 'AvgA'. Berdasarkan kolom-kolom tersebut, program akan menghasilkan grafik diameter terhadap waktu, kecepatan terhadap waktu, serta akselerasi terhadap waktu. Adapun terdapat beberapa penanda pada masing-masing grafik:

- Garis putus-putus vertikal serta titik berwarna hijau pada detik LED putih menyala.
- Garis putus-putus vertikal serta titik berwarna merah pada detik LED putih mati.
- Titik biru pada titik minimum grafik, menandakan titik konstiksi maksimum pupil.
- Khusus grafik Kecepatan dan Akselerasi, terdapat titik hitam pada titik pupil mulai konstiksi.
- Titik MCV dan MCA ditandai titik kuning dan teks "MCV" pada grafik Kecepatan, dan "MCA" pada grafik Akselerasi.

## 2.11 Perhitungan Nilai-Nilai PLR

*Baseline Diameter* (D) didapat dari rata-rata ukuran pupil pada kolom 'AvgD' dari detik pertama hingga sesaat sebelum LED dinyalakan. *Minimum Pupil Diameter* (Dmin) didapat dari nilai terkecil pada kolom 'AvgD' setelah LED dinyalakan. *Time Latency* (TL) merupakan jumlah detik ketika LED dinyalakan hingga awal terjadinya konstiksi pada data kecepatan. *Time Constriction* (TC) merupakan jumlah detik titik awal konstiksi hingga titik konstiksi maksimum data kecepatan. Data konstiksi dimulai ketika nilai kecepatan atau akselerasi pupil terus menurun dan berhenti pada titik konstiksi maksimum. Data konstiksi untuk mendapatkan nilai MCV didapat dari kolom 'AvgK'. Nilai MCV didapat dari nilai terkecil kolom tersebut. Nilai ACV merupakan nilai rata-rata data tersebut. Data konstiksi untuk mendapatkan nilai MCA didapat dari kolom 'AvgA'. Nilai MCA didapat dari nilai terkecil kolom tersebut. Nilai ACA merupakan nilai rata-rata data tersebut.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Kalibrasi Satuan Piksel ke Milimeter

Pengujian ini dilakukan dengan cara memasukkan nilai *normalized ratio* pada tiap model regresi linier pada masing-masing jarak. Tabel 3 menunjukkan rata-rata *error* yang diperoleh dari hasil kalibrasi untuk tiap jarak pada kamera.

**Tabel 3. Pengujian Kalibrasi Kamera Kiri**

| <b>Jarak (mm)</b> | <b>Rata-rata % error kiri</b> | <b>Rata-rata % error kanan</b> |
|-------------------|-------------------------------|--------------------------------|
| 85                | 2,4 %                         | 2,0 %                          |
| 87                | 2,3 %                         | 2,3 %                          |
| 89                | 2,3 %                         | 2,2 %                          |
| 91                | 2,0 %                         | 2,1 %                          |
| 93                | 2,0 %                         | 2,3 %                          |
| 95                | 2,4 %                         | 2,1 %                          |
| 97                | 2,5 %                         | 2,4 %                          |
| 99                | 2,3 %                         | 2,3 %                          |
| 101               | 2,9 %                         | 2,3 %                          |
| 103               | 2,3 %                         | 2,6 %                          |

### 3.2 Hasil Kalibrasi Sensor

Setiap pembacaan sensor akan dihitung berdasarkan hasil regresi linier yang telah didapatkan. Tabel 4 menunjukkan hasil uji pembacaan jarak baru sensor.

**Tabel 4. Pembacaan Jarak Baru Sensor**

| <b>Jarak (mm)</b> | <b>Bacaan Sensor Kiri (mm)</b> | <b>Bacaan Sensor Kanan (mm)</b> |
|-------------------|--------------------------------|---------------------------------|
| 85                | 85                             | 85                              |
| 87                | 87                             | 87                              |
| 89                | 89                             | 88                              |
| 91                | 91                             | 91                              |
| 93                | 93                             | 93                              |
| 95                | 95                             | 95                              |
| 97                | 97                             | 96                              |
| 99                | 99                             | 99                              |
| 101               | 101                            | 100                             |
| 103               | 103                            | 102                             |

Dari tabel tersebut, sensor kiri telah terkalibrasi secara akurat dan sensor kanan masih terdapat *error* hingga 1 mm.

### 3.3 Hasil Lokalisasi Pupil

Pengujian lokalisasi pada bagian pupil dilakukan dengan cara menerapkan *cascade classifier* dan fungsi *detectMultiScale* pada sebuah rekaman. 10 buah rekaman dari 5 orang pasien digunakan untuk menguji lokalisasi. Tabel 5 menunjukkan hasil lokalisasi pada tiap video. Rata-rata akurasi lokalisasi pupil sebesar 99,07%.

**Tabel 5. Hasil Uji Lokalisasi Pupil**

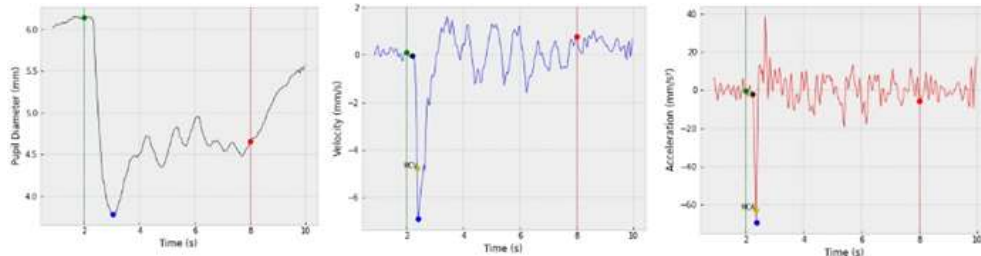
| Rekaman | Total <i>Frame</i> | Jumlah <i>Frame</i> Terlokalisasi | Akurasi |
|---------|--------------------|-----------------------------------|---------|
| 1       | 300                | 300                               | 100 %   |
| 2       | 300                | 300                               | 100 %   |
| 3       | 300                | 300                               | 100 %   |
| 4       | 300                | 300                               | 100 %   |
| 5       | 300                | 291                               | 97 %    |
| 6       | 300                | 300                               | 100 %   |
| 7       | 300                | 300                               | 100 %   |
| 8       | 300                | 300                               | 100 %   |
| 9       | 300                | 281                               | 93,7 %  |
| 10      | 300                | 300                               | 100%    |

### 3.4 Hasil pada Pasien Normal

Pengujian dilakukan pada pasien normal A berusia 21 tahun dengan kognitif normal.

1. Pengujian Grafik

Gambar 11 menunjukkan grafik yang dihasilkan setelah integrasi dengan aplikasi *desktop*.



**Gambar 10. Grafik Diameter, Kecepatan, dan Akselerasi Pasien Normal A**

2. Pengujian Nilai-Nilai PLR

Tabel 6 menunjukkan nilai-nilai PLR pasien normal A.

**Tabel 6. Nilai-Nilai PLR Pasien Normal A**

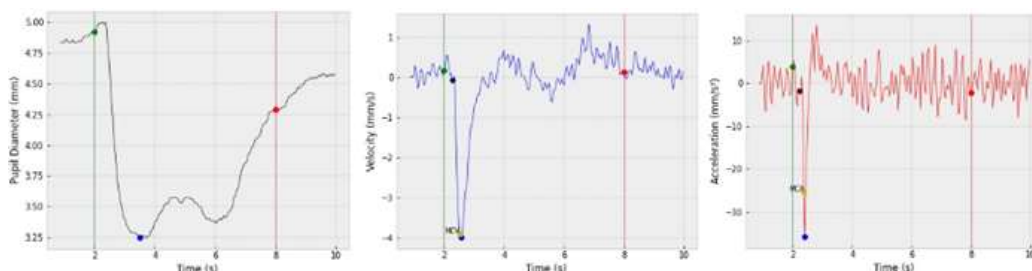
| <i>Parameter</i> | <i>Nilai</i>             |
|------------------|--------------------------|
| D                | 5,467 mm                 |
| Dmin             | 3,438 mm                 |
| TL               | 0,2 s                    |
| TC               | 0,233 s                  |
| MCV              | 6,918 mm/s               |
| ACV              | 4,574 mm/s               |
| MCA              | 57,788 mm/s <sup>2</sup> |
| ACA              | 18,574 mm/s <sup>2</sup> |

### 3.5 Hasil pada Pasien Lansia

Perlu diketahui bahwa sejumlah penelitian yang ada memiliki hasil-hasil diagnostik yang berbeda antara satu dengan yang lain. Sehingga, belum ada batas yang tegas antara nilai PLR pada pasien normal dengan pasien demensia. Namun, terdapat pola yang serupa yakni bahwa respons pupil pasien dengan demensia cenderung lebih menurun (**Capó-Aponte, dkk, 2013**) (**Frost, dkk, 2017**) (**Siva kumar, dkk, 2020**). Pengujian dilakukan pada pasien lansia A berusia 69 tahun yang memiliki gangguan kognitif.

#### 1. Pengujian grafik

Gambar 11 menunjukkan grafik yang dihasilkan setelah integrasi dengan aplikasi *desktop*.



**Gambar 11. Grafik Diameter, Kecepatan, dan Akselerasi Pasien Lansia A**

#### 2. Pengujian nilai-nilai PLR

Tabel 7 menunjukkan nilai-nilai PLR dari pasien lansia A.

**Tabel 7. Nilai-Nilai PLR Pasien Lansia A**

| Parameter | Nilai                    |
|-----------|--------------------------|
| D         | 3,427 mm                 |
| Dmin      | 2,67 mm                  |
| TL        | 0,3 s                    |
| TC        | 0,167 s                  |
| MCV       | 3,273 mm/s               |
| ACV       | 2,437 mm/s               |
| MCA       | 19,021 mm/s <sup>2</sup> |
| ACA       | 11,657 mm/s <sup>2</sup> |

### 3.6 Analisis Sistem Keseluruhan

Dalam 30 kali pengujian, pemrosesan data gagal 8 kali. Hal ini dapat disebabkan oleh imputasi data yang tidak berhasil dilakukan, yakni apabila hanya terdapat 1 titik data, sebelum atau sesudah data berisi 'NaN', sehingga program tidak dapat melakukan interpolasi.

## 4. KESIMPULAN

Berdasarkan perancangan dan pengujian yang telah dilakukan, dapat disimpulkan bahwa *Cascade classifier* menggunakan fitur LBP cocok digunakan untuk lokalisasi pada bagian pupil dengan akurasi 99,07%, dengan kalibrasi satuan dari piksel ke milimeter memiliki *error* hanya sekitar 2% hingga 2,9%. Sensor jarak VL53L0X memiliki *error* hingga 1 milimeter. Dari 30 kali pengujian, pemrosesan grafik dan nilai-nilai PLR gagal sebanyak 8 kali, sehingga perancangan sistem 73,33% akurat.

## DAFTAR RUJUKAN

- Bradski, G., & Kaehler, A. (2016). *Learning OpenCV 3*. California: O'Reilly Media, Inc.
- Capó-Aponte, J.E., et al. (2013). Pupillary Light Reflex as an Objective Biomarker for Early Identification of Blast-Induced mTBI, *Journal of Spine*, 4, 1-5.
- Frost, S., et al. (2017). Evaluation of Cholinergic Deficiency in Preclinical Alzheimer's Disease Using Pupillometry. *Journal of Ophthalmology*, 1-8.
- Hall, C. A., & Chilcott, R. P., (2018). Eyeing up the future of the pupillary light reflex in neurodiagnostics, *Diagnostics*, 8(1), 19.
- Hansun, S. (2013). A New Approach of Moving Average Method in Time Series Analysis. *The 2013 Conference on New Media Studies (CoNMedia)*, (pp. 1-4).
- Kerr, R. G., et al. (2016). Underestimation of Pupil Size by Critical Care and Neurosurgical Nurses. *American Journal of Critical Care*, 25(3), 9-213.
- Kushsairy K., et al. (2014). A Comparative Study between LBP and Haar-like features for Face Detection Using OpenCV. *4th International Conference on Engineering Technology and Technopreneurship (ICE2T)*, (pp. 335-339).
- Laković N., et al. (2019). Application of Low-Cost VL53L0X ToF Sensor for Robot Environment Detection. *18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, (pp. 1-4).
- Phase, T. (2020). Building Custom HAAR-Cascade Classifier for face Detection. *International Journal of Engineering and Technical Research*, 8(12), 1-6.
- Sandilayan, M. B., & Dening T. (2015). Signs and Symptoms of Dementia. *Nursing Standard*, 29(41), 42-51.
- Setjo, C. H., & Faridah, B.A. Thermal Image Human Detection Using Haar-Cascade Classifier. *7th International Annual Engineering Seminar (InAES)*, (pp. 1-6).
- Siva kumar, A. V, et al. (2020). Quantitative determination of pupil by dynamic pupillometry using infrared videography – Role in evaluation of autonomic activity. *Clinical Epidemiology and Global Health*, 8, 728-732.
- Subdirektorat Statistik Pendidikan dan Kesejahteraan Sosial. (2020). *Statistik Penduduk Lanjut Usia 2020*. Jakarta: Badan Pusat Statistik.
- Verbeek, M. (2017). Using linear regression to establish empirical relationships. *IZA World of Labor*, 2, 336.