

Rancang Bangun Sistem Deteksi Kantuk Terhadap Keamanan Pengemudi Mobil

Design And Development Of Drowsiness Detection System For Car Driver Safety

1st Muhammad Adhri Basari
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
adhribasari@student.telkomuniversity.ac.id

2nd Muhammad Ikhsan Sani
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
ikhkansani@telkomuniversity.ac.id

3rd Marlindia Ike Sari
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
marlindia@telkomuniversity.ac.id

Abstrak-Di Indonesia, angka kecelakaan lalu lintas meningkat setiap tahunnya, dari beberapa faktor penyebab banyaknya kecelakaan yaitu rasa kantuk saat berkendara. Proyek ini sangat membantu bagi pengemudi mobil untuk mengantisipasi terjadinya kecelakaan. Perangkat Keras Deteksi Kantuk ini dirancang dengan menggunakan Raspberry Pi 4, dengan metode prototype. Sistem ini akan mendeteksi rasio mata 0.18 pada pengemudi yang menandakan kantuk dengan atau tidak memakai kacamata dengan intensitas cahaya 199,8 lux, 989,0 lux, dan 0,6 lux juga akan berpengaruh bagi sistem ini, dan pada sistem ini juga menggunakan cita digital sebagai pendeteksi kantuk jika jarak pengemudi dengan kamera 25 cm. Hasil dari pengujian ini kamera dan perangkat output berjalan dengan baik dengan rata-rata delay kantuk dengan perangkat output sebesar 07.28 detik. Sistem akan menyalakan perangkat output sebagai tanda peringatan bagi pengemudi yang diharapkan dapat mengurangi angka kecelakaan dikarenakan kantuk.

Kata kunci-kantuk, kecelakaan, Raspberry Pi 4, output.

BAB I PENDAHULUAN

Di Indonesia, angka kecelakaan lalu lintas meningkat setiap tahunnya. Berdasarkan nmc.polri.info pada tahun 2019 data jumlah kecelakaan lalu lintas naik menjadi 3% dibandingkan dengan tahun 2018 dimana terdapat 103.672 kasus kecelakaan lalu lintas [1]. Banyaknya kasus kecelakaan lalu lintas disebabkan oleh berbagai faktor diantaranya faktor manusia, faktor kendaraan dan faktor lingkungan [2].

Dari beberapa faktor penyebab kecelakaan lalu lintas diatas, faktor kecelakaan yang disebabkan oleh manusia atau dalam hal ini pengemudi kendaraan menjadi perhatian dalam penulisan porposal ini.

Rasa kantuk saat berkendara adalah salah satu kondisi yang sering diabaikan oleh pengendara. Hal ini menjadi salah satu penyebab seringnya terjadi kecelakaan lalu lintas, terutama pada saat berkendara pada jarak yang cukup jauh. Berdasarkan faktor tersebut, maka kecelakaan kendaraan bermotor dapat diminimalisir dengan mendeteksi penyebab kecelakaan yaitu rasa kantuk dari pengendara [3].

Sistem pendeteksi kantuk di dalam mobil sangat dibutuhkan sebagai peringatan bagi pengemudi ketika mengantuk. Pengolahan citra digital bisa dimanfaatkan untuk membuat sistem pendeteksi kantuk dengan algoritma yang ada dalam pemrograman Python dan OpenCV, untuk menentukan rasio mata atau koordinat mata pengemudi menggunakan metode Facial Landmarks, jika mata tertutup atau rasio mata dibawah 0.18 maka akan tedeteksi kantuk berdasarkan Eye Aspect Ratio [4].

Berdasarkan faktor kecelakaan dikarenakan kantuk, maka dari faktor tersebut dapat diminimalisir. Dengan adanya sistem pendeteksi kantuk diharapkan mampu mengurangi angka kecelakaan yang diakibatkan oleh kantuk. Dalam hal ini maka dibuatlah sistem deteksi kantuk untuk keamanan pengemudi mobil.

II. KAJIAN TEORI

A. Data Kecelakaan



Berkendara di jalan Tol memang pilihan jika ingin sampai ditujuan dengan cepat, sebab di jalan Tol mobil bisa dipacu lebih cepat dibanding jalan biasa. Namun di jalan Tol sudah banyak kecelakaan yang terjadi. Ada beberapa penyebab yaitu pengemudi mengantuk, ban pecah, rem blong, dan kondisi jalan tidak layak.

Pada Gambar 2. 1 merupakan grafik kecelakaan berdasarkan data yang diambil dari Korps Lalu Lintas Kepolisian Negara Indonesia (Korlantas Polri), pada 4 November 2021, 183 kecelakaan di 27 wilayah Polda. Sebagian besar kecelakaan terjadi pada daerah Jawa Timur pada Tol Trans Jawa [10].

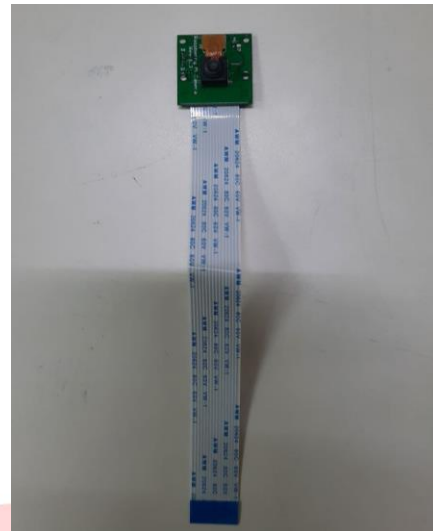
Berdasarkan data dari Komite Nasional Keselamatan Transportasi (KNKT) 80 persen penyebab kecelakaan di jalan Tol diakibatkan pengemudi Lelah atau mengantuk [11].

B. Raspberry Pi 4 Model B



Mikrokontroler Raspberry Pi adalah komputer papan tunggal, memiliki ukuran sebesar kartu kredit. Menggunakan sistem operasi Raspbian, dengan prosesor 700MHz ARM11. Terdapat dua tipe Raspberry Pi yakni tipe A dan B. Perbedaannya pada kapasitas memori yang digunakan untuk tipe A 256MB dan tipe B 512MB. Pada penelitian ini menggunakan Raspberry Pi 4 model B [7].

C. Kamera Raspberry



Modul kamera Raspberry atau sering disebut dengan Raspicam merupakan kamera yang digunakan untuk mengambil foto atau video. Besar resolusi pada Raspicam sebesar 5 megapixel dan mendukung video 720p, 1080p dan VGA90. Port pada Raspicam yaitu melalui port CSI (*Camera Serial Interface*) 15 pin yang ada pada Raspberry Pi [12]. Pada sistem ini kamera Raspberry digunakan untuk mengambil video area wajah terutama pada mata.

D. Buzzer



Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Cara kerja *buzzer* hampir sama dengan *speaker*, jadi *buzzer* terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik kedalam atau keluar, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak balik sehingga membuat udara bergetar yang akan menghasilkan suara [13]. Pada sistem ini *buzzer* digunakan sebagai tanda peringatan pengemudi sedang mengantuk atau menutup mata selama waktu yang telah ditentukan.

E. LED (Light Emitted Diode)



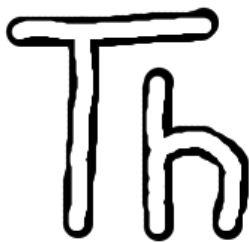
LED atau Light Emited Diode adalah suatu semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju atau searah. Dengan secara Bahasa bisa diartikan sebagai diode yang memancarkan cahaya bila dialiri listrik [14]. Pada sistem ini menggunakan LED warna merah sebagai bentuk peringatan bagi pengemudi jika terdeteksi kantuk.

F. Python



Python adalah Bahasa pemrograman dengan script yang sederhana dan aplikasinya sudah meluas, Python juga bersifat open source. Karena itu Python dianggap sebagai salah satu Bahasa terbaik untuk pemula (Adawadkar, 2017) [12]. Pada sistem ini Python digunakan untuk memprogram perangkat output agar bisa berjalan sesuai yang diinginkan. Mikrokontroler ini sangat bisa digunakan.

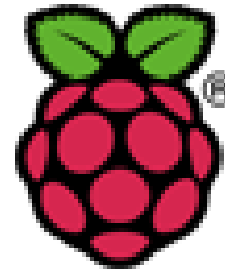
G. Thonny Python IDE



Thonny merupakan IDE untuk Bahasa pemrograman Python. Pada sistem ini Thonny

Python IDE digunakan untuk menulis dan menjalankan program [16].

H. Sistem Operasi Raspbian



Raspbian adalah sistem operasi resmi Raspberry Pi, Raspbian adalah versi Linux yang dibuat khusus untuk Raspberry Pi. Perangkat ini dilengkapi dengan semua perangkat lunak yang diperlukan contohnya Thonny Python IDE [18]. Pada perancangan ini menggunakan sistem operasi dengan versi kernel 5.15 dan sistem 64-bit yang akan digunakan untuk menjalankan Raspberry Pi 4.

III. METODE

A. Gambaran Sistem Saat Ini

Sistem pada saat ini merupakan pendeteksi kantuk bagi pengemudi mobil, menggunakan metode *image processing* untuk pendeteksi wajah. Penentuan rasio mata menggunakan facial landmarks, dan menggunakan rumus EAR untuk melihat nilai rasio mata. Kondisi yang menandakan pengemudi kantuk yaitu sistem menghitung presentase penutupan mata [19].

B. Identifikasi Kebutuhan Sistem

No	Kebutuhan Fungsional
1	Mendeteksi area mata pengemudi menggunakan Kamera Raspberry dan Raspberry Pi 4
2	Mendeteksi pengemudi saat menunduk karena kantuk menggunakan citra digital
3	Membuat komponen output sebagai tanda peringatan bagi pengemudi jika mata tertutup

Pada tabel 3-1 kebutuhan fungsional dalam perancangan sistem ini yaitu pendeteksian objek wajah pengemudi terutama area mata dengan Kamera Raspberry , mendeteksi pengemudi menunduk dalam kondisi kantuk dan pembuatan komponen output sebagai tanda peringatan pengemudi mengantuk

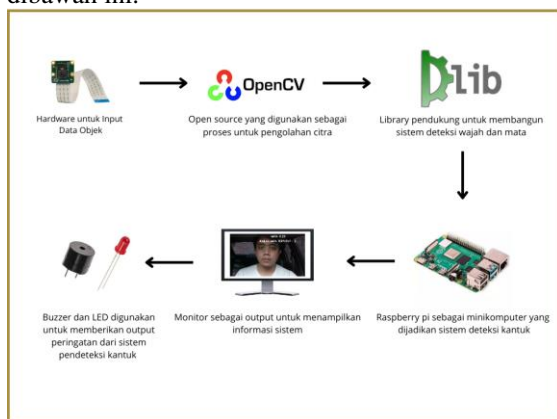
No	Kebutuhan Non Fungsional
1	Raspberry Pi 4 sebagai mikrokontroler pengolah data
2	Kamera Raspberry untuk menangkap objek

3	Monitor untuk menampilkan dan menjalankan sistem
4	Buzzer sebagai output suara
5	LED sebagai output cahaya
6	Open CV sebagai <i>library</i> pendukung
7	<i>Math</i> sebagai <i>library</i> pendukung
8	RPi.GPIO sebagai <i>library</i> pendukung
9	<i>Thonny Python IDE</i> untuk program editor

Pada tabel 3-2 kebutuhan non fungsional dalam perancangan sistem ini yaitu Raspberry Pi 4 sebagai pengolah data, kamera Raspberry sebagai pengambil objek, monitor sebagai penampil data dan menjalankan sistem, *buzzer* dan LED sebagai peringatan bagi pengemudi, Open CV sebagai *library* untuk integrasi ke kamera, *Math* sebagai *library* pendukung untuk pendeteksi jarak, RPi.GPIO sebagai *library input output*, dan *Thonny Python IDE* sebagai program editor

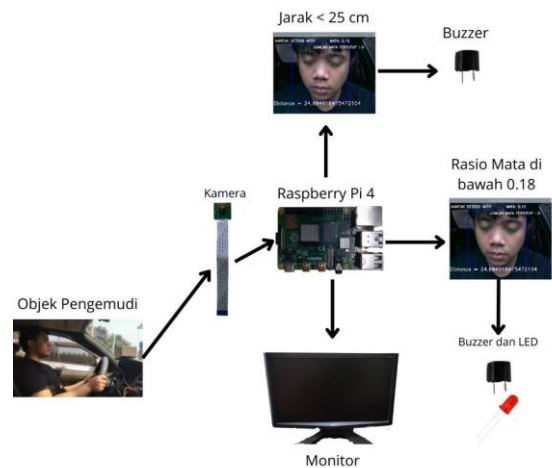
C. Perancangan Sistem

Untuk melihat sistem secara keseluruhan dan pembagian sistem dapat dilihat pada diagram blok dibawah ini.



Pada Gambar 3-1 adalah Gambaran sistem keseluruhan pada sistem deteksi kantuk. Namun pada bagian proyek akhir ini lebih berfokus terhadap kamera yang akan menangkap objek, library OpenCV akan digunakan sebagai pengolah citra untuk jarak pengemudi dengan kamera, dan komponen *output buzzer* dan LED yang akan bekerja jika rasio mata kurang dari 0.18 dan jarak pengemudi dengan kamera kurang dari 25 cm.

D. Diagram Blok Sistem



Pada Gambar 3-2 merupakan diagram blok dari sistem yang dibuat pada objek pengemudi terutama wajah akan ditangkap oleh kamera dan diteruskan ke Raspberry Pi yang akan diolah menggunakan program untuk menentukan rasio mata dan jarak sebagai deteksi kantuk, *buzzer* dan LED akan bekerja sebagai tanda peringatan ketika pengemudi kantuk.

E. Metode Pengerjaan

Metode pengerjaan dilakukan menggunakan metode prototype dengan beberapa tahap pengerjaan seperti berikut.

1. Analisa Kebutuhan

Pada tahap ini dilakukan identifikasi apa saja yang dibutuhkan sistem yang akan dibuat.

2. Membangun Prototyping

Pada tahap ini dilakukan pembuatan perancangan sementara.

3. Evaluasi Prototyping

Pada tahap ini untuk meng-evaluasi prototype apakah berjalan dengan baik.

4. Mengkodekan Sistem

Pada tahap ini melakukan penulisan program untuk berjalannya sistem.

5. Menguji Sistem

Pada tahap ini apabila pengkodean sistem selesai maka sistem akan diuji.

6. Evaluasi Sistem

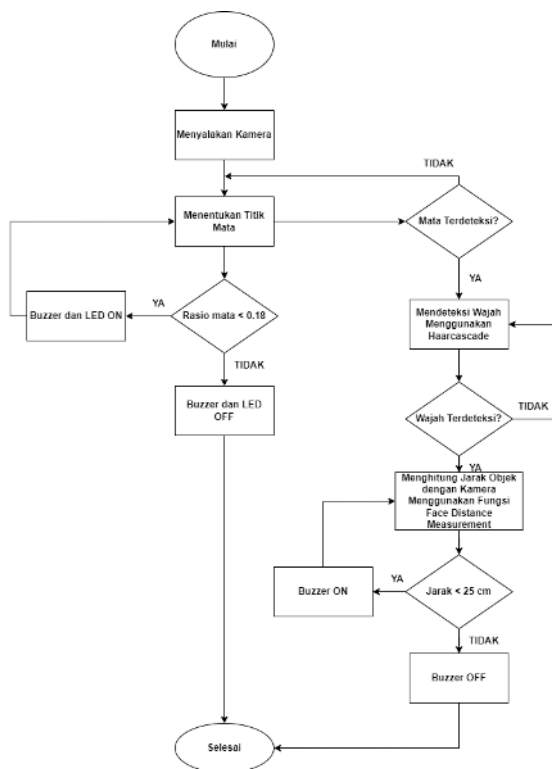
Pada tahap ini evaluasi dilakukan untuk melihat apakah sistem bekerja sesuai dengan yang diinginkan.

7. Menggunakan Sistem

Pada tahap ini sistem yang diuji dan sesuai dengan yang diinginkan maka sistem sudah siap digunakan

F. Flowchart

Alur kerja yang lebih rinci dari “Rancang Bangun Sistem Deteksi Kantuk Terhadap Keamanan Pengemudi Mobil” dapat dilihat pada flowchart berikut.



A. Perangkat Keras

Perangkat keras yang digunakan pada “Rancang Bangun Sistem Deteksi Kantuk Terhadap Keamanan Pengemudi Mobil” adalah sebagai berikut.

1. Raspberry Pi 4

Raspberry pada sistem ini digunakan sebagai pusat kontrol untuk menjalankan program.

2. Kamera Raspberry

Kamera Raspberry pada sistem ini digunakan sebagai pendeteksi area wajah terutama koordinat mata.

3. Buzzer

Buzzer digunakan sebagai tanda peringatan saat sistem mendeteksi kantuk pada mata pengemudi.

4. LED (Light Emitted Diode)

LED yang digunakan pada perancangan ini yaitu LED berwarna merah yang digunakan sebagai tanda peringatan saat sistem mendeteksi kantuk pada mata pengemudi.

B. Perangkat Lunak

Perangkat lunak yang digunakan pada “Rancang Bangun Sistem Deteksi Kantuk Terhadap Keamanan Pengemudi Mobil” adalah sebagai berikut:

1. Thonny Python IDE

Thonny Python IDE pada sistem ini digunakan untuk memprogram Raspberry, yang nantinya akan menjalankan sistem tersebut.

2. Sistem Operasi Raspberry

Pada perancangan ini menggunakan sistem operasi dengan kernel version 5.15 dan sistem 64-bit yang akan digunakan untuk menjalankan Raspberry Pi 4.

3. Open CV

Library ini digunakan untuk integrasi ke kamera.

4. Math

Library ini sebagai pendukung deteksi jarak objek

5. RPi.GPIO

Library ini digunakan untuk integrasi ke input dan output.

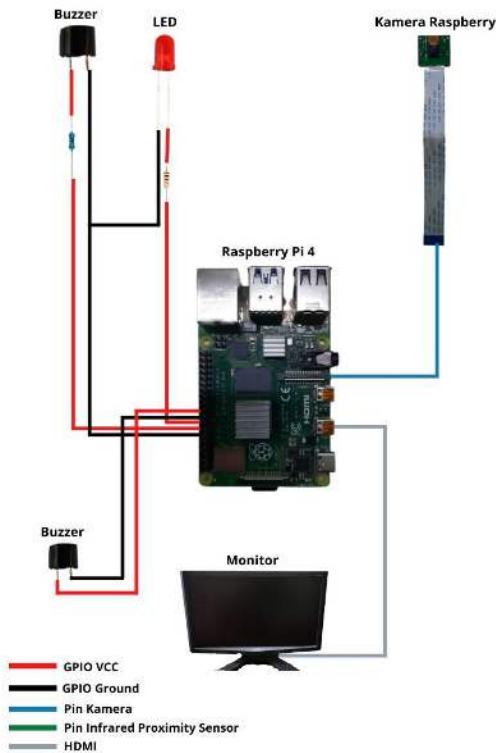
IV. HASIL DAN PEMBAHASAN

A. Implementasi

Membahas tahapan-tahapan perancangan pembuatan sistem sampai dengan pengujian. Berikut merupakan tahapan-tahapan implementasi pada Rancang Bangun Sistem ini.

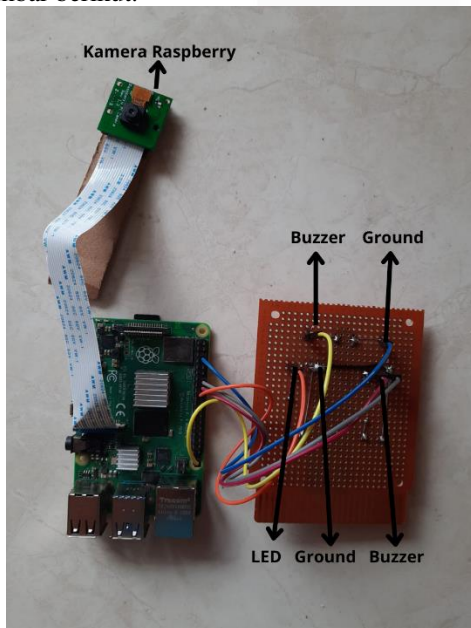
1. Rangkaian Skematik Sistem

Rangkaian skematik sistem ini terdapat komponen utama yaitu Raspberry Pi 4, Kamera Raspberry, Monitor, Buzzer, dan LED. Untuk pengkabelan dan pemasangan pada pin Raspberry dapat dilihat pada Gambar sebagai berikut.



2. Pemasangan Komponen ke Pin Raspberry Pi

Pada bagian ini setelah proses pensolderan telah selesai maka dilakukan pemasangan hardware ke pin pada Raspberry Pi. Buzzer 1 dipasang pada GPIO23, buzzer 2 dipasang pada GPIO21, dan LED dipasang pada GPIO24. pada proses ini membutuhkan kabel jumper female to female seperti Gambar berikut.



3. Perangkaian Komponen ke Kerangka

Pada proses ini kerangka dan komponen yang sudah siap akan digabungkan, untuk komponen akan di tempel pada kerangka menggunakan double tape

agar tidak mudah lepas, pada bagian ini kerangka di ibaratkan sebagai dashboard mobil sebagai berikut.



4. Implementasi Program

Pembuatan program integrasi pada perancangan sistem ini menggunakan aplikasi bawaan Raspberry yaitu Thonny Python IDE untuk mempermudah penulisan dan menjalankan program. Penjelasan program yang lebih detail ada pada penjelasan dibawah ini.

1) Library

```
from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import dlib
import cv2
import time
from threading import Thread
import RPi.GPIO as GPIO
from time import sleep
import math
```

Library cv2 digunakan untuk mengintegrasikan ke kamera Raspberry, library RPi.GPIO as GPIO untuk integrasi ke buzzer dan LED, library math digunakan untuk rumus membuat jarak.

2) Integrasi Kamera

```
print("[INFO] Menyalakan video raspberry pi...")
video_capture = cv2.VideoCapture(0)
time.sleep(3)
```

menandakan kamera segera hidup, untuk waktu kamera hidup saat program dijalankan membutuhkan waktu sekitar 3 detik. Video capture = cv2.VideoCapture(0) ini digunakan untuk memanggil kamera dan menampilkan pada layar monitor dengan bantuan library cv2.

Kode ini digunakan untuk mengubah skala dan mengubah image citra warna menjadi grayscale.

3) Jarak

```

if (eyeAspectRatio < EYE_ASPECT_RATIO_THRESHOLD):
    COUNTER += 1
for (x, y, w, h) in face_rectangle:
    distancei = (5*3.14 *
180)/(w+h*360)*1000 + 3
    print (distancei)
    jarak = distancei *2.54
    jarak = math.floor(jarak)
    cv2.rectangle(frame, (x,y),
(x+w,y+h), (255,0,0), 2)

    if distancei < batas:
        GPIO.output (buzzer2,GPIO.HIGH)
        print("bahaya bahaya")
    else:
        GPIO.output (buzzer2,GPIO.LOW)

```

Pada kondisi mata terbuka program haarcascade mendeteksi wajah agar dapat menentukan jarak pengemudi dengan objek dengan rumus jarak, kemudian jarak diatur kurang dari 25 cm untuk menandakan kantuk, buzzer akan menyala dan menampilkan peringatan bahaya pada monitor sebagai tanda peringatan.

```

else:
    COUNTER = 0
    ALARM_ON = False
for (x, y, w, h) in face_rectangle:
    distancei = (5*3.14 *
180)/(w+h*360)*1000 + 3
    print (int (distancei))
    jarak = distancei *2.54
    jarak = math.floor(jarak)
    cv2.rectangle(frame, (x,y),
(x+w,y+h), (255,0,0), 2)

    if distancei < batas:
        GPIO.output (buzzer2,GPIO.HIGH)
        print("bahaya bahaya")
    else:
        GPIO.output (buzzer2,GPIO.LOW)

```

Pada kondisi mata tertutup program haarcascade mendeteksi wajah agar dapat menentukan jarak pengemudi dengan objek dengan rumus jarak, kemudian jarak diatur kurang dari 25 cm untuk menandakan kantuk, buzzer akan menyala dan menampilkan peringatan bahaya pada monitor sebagai tanda peringatan.

4) Intergrasi Output

```

GPIO.setwarnings (False)

buzzer = 16
buzzer2 = 40
led = 18
GPIO.setmode (GPIO.BOARD)
GPIO.setup (buzzer,GPIO.OUT)
GPIO.setup (buzzer2,GPIO.OUT)
GPIO.setup (led,GPIO.OUT)

```

Pada `setwarnings(FALSE)` untuk menonaktifkan peringatan, saat kondisi program berhenti dan dijalankan kembali. Pada sistem bagian integrasi pin komponen output ini buzzer dipasang pada GPIO23, buzzer2 dipasang pada GPIO21, dan LED dipasang dengan GPIO24, lalu diikuti dengan `GPIO.setup(pin output,GPIO.OUT)` yang artinya library akan menyambungkan komponen output dengan Raspberry yang nantinya akan bisa berjalan sesuai dengan program dibawah.

```

if not ALARM_ON:
    TOTAL +=1
    ALARM_ON = True

GPIO.output (buzzer,GPIO.HIGH)

GPIO.output (led,GPIO.HIGH)

```

Pada kondisi `if` mendeteksi keadaan mata tertutup maka dimasukkan *library* GPIO lalu dipasang dengan perangkat *output* dan untuk memanggil perangkat *output* dimasukkan dengan pin perangkat *output*, dan terakhir pada kondisi di tambahkan *library* GPIO dan masukkan kondisi HIGH untuk memanggil komponen *output*.

```

else:
    COUNTER = 0
    ALARM_ON = False
    GPIO.output (buzzer,GPIO.LOW)
    GPIO.output (led,GPIO.LOW)

```

Pada kondisi `else` mendeteksi keadaan mata sudah kembali terbuka atau normal. Pada `else` ini dibuat program `GPIO.output(pin output,GPIO.LOW)` yang artinya *library* akan memberhentikan kerja dari perangkat *output*.

B. Pengujian

Untuk mengetahui apakah sistem berjalan dengan baik maka diuji dengan beberapa variabel. Pengujian pada rancang bangun sistem ini akan dilakukan pada subbab berikut ini.

1. Pengujian Posisi Kamera dan Jarak
 - a. Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui kinerja sistem dapat bekerja dengan baik pada posisi kamera dan jarak objek dengan kamera.

b. Skenario Pengujian



Pada pengujian ini dilakukan kepada responden laki-laki umur 22 tahun dengan tinggi badan 169 cm di dalam mobil tipe Great Corolla tahun 1996, pengujian dilakukan pada pagi hari pukul 06.30 WIB dan menentukan intensitas cahaya menggunakan aplikasi CPU-Z pada ponsel. Posisi kamera pada pengujian ini diletakkan pada dashboard dan spion tengah mobil. Pengujian dilakukan pada jarak 20 cm, 30 cm, 45 cm, 60 cm, dan 65 cm.

1) Pengujian intensitas cahaya pada pukul 06.30 WIB

Pukul 06.30 WIB
als-tcs3701 199,8 lux

Hasil pengujian intensitas cahaya pada pagi hari pukul 06.30 WIB menggunakan aplikasi CPU-Z mendapat 199,8 lux

2) Peletakan posisi kamera

Dashboard Mobil	Spion Tengah Mobil
	

Peletakan posisi kamera pada dashboard mobil dan spion tengah mobil.

c. Hasil Pengujian

Pengujian kamera di dashboard mobil Hasil pengujian ini diuraikan pada Gambar sebagai berikut.



(a) Jarak 20 cm

(b) Jarak 30 cm



(c) Jarak 45 cm

(d) Jarak 60 cm



(e) Jarak 65 cm

1. Pengujian kamera pada spion tengah mobil

Hasil pengujian ini diuraikan pada Gambar sebagai berikut.



(a) Jarak 20 cm

(b) Jarak 30 cm



(c) Jarak 45 cm

(d) Jarak 60 cm



(e) Jarak 65 cm

d. Analisis Hasil Pengujian

Hasil pengujian kamera di dashboard dapat mendeteksi rasio mata dari jarak 20 cm, 30 cm, 45 cm, 60 cm, dan 65 cm dengan baik, sedangkan pengujian kamera di spion tengah rasio mata pada jarak tertentu tidak dapat terdeteksi. Maka dari itu pengujian selanjutnya sistem akan dipasang pada dashboard dan dengan jarak 45 cm dan 60 cm untuk mengetahui rasio mata dengan baik.

2. Pengujian Objek dengan Kamera

a. Tujuan

Pengujian ini dilakukan untuk mengetahui kinerja kamera Raspberry dapat berjalan dan mendeteksi objek di dalam mobil dengan baik.

b. Skenario Pengujian

Pada pengujian ini dilakukan di dalam mobil tipe Great Corolla tahun 1996, di pengujian ini dilakukan 4 responden dengan ciri-ciri sebagai berikut.

- 1) orang ke-1 laki-laki, umur 22 tahun, dan tinggi badan 165 cm.
- 2) orang ke-2 laki-laki, umur 21 tahun, dan tinggi badan 171 cm.
- 3) orang ke-3 laki-laki, umur 21 tahun, dan tinggi badan 170 cm.
- 4) orang ke-4 laki-laki, umur 21 tahun, dan tinggi badan 169 cm.

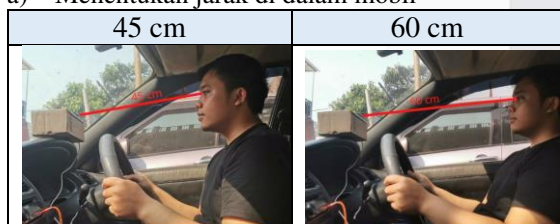
dengan memakai dan tidak memakai kacamata, Pengujian ini juga menentukan intensitas cahaya pada saat pagi, siang dan malam menggunakan aplikasi dari ponsel yaitu CPU-Z, dan menggunakan jarak yang telah ditentukan yaitu 45 cm dan 60 cm.

1) Pengujian intensitas cahaya di dalam mobil

Pukul 06.30 WIB	Pukul 13.45 WIB	Pukul 22.30 WIB
als-tcs37 199,8 lux	als-tcs37 989,0 lux	als-tcs37 0,6 lux

Pengujian intensitas cahaya dilakukan pada 3 waktu berbeda, yaitu pada pagi hari pukul 06.30 WIB mendapat intensitas cahaya 199,8 lux, pada siang hari pukul 13.45 WIB mendapat intensitas cahaya 989,0 lux, dan pada malam hari pukul 22.00 WIB mendapat intensitas cahaya 0,6 lux.

a) Menentukan jarak di dalam mobil



Pada pengujian ini ditentukan jarak kamera dengan objek antara 45 cm dan 60 cm.

2) Pengujian pada pagi hari pukul 06.30 WIB

Respon	Kaca mata	Jarak (cm)	Intensitas (lux)	Keterangan	Gambar
Orang ke-1	Tidak	45	199,8	Objek Terdeteksi	

Orang ke-2	Ya	45	199,8	Objek Terdeteksi	
Orang ke-3	Tidak	60	199,8	Objek Terdeteksi	
Orang ke-4	Ya	60	199,8	Objek Terdeteksi	

Pada tabel 4-5 merupakan hasil dari pengujian kamera pada pagi hari pukul 06.30 WIB.

3) Pengujian pada siang hari pukul 13.45 WIB

Respon	Kaca mata	Jarak (cm)	Intensitas (lux)	Keterangan	Gambar
Orang ke-1	Tidak	45	989,0	Objek Terdeteksi	
Orang ke-2	Ya	45	989,0	Objek Terdeteksi	
Orang ke-3	Tidak	60	989,0	Objek Terdeteksi	
Orang ke-4	Ya	60	989,0	Objek Terdeteksi	

Pada tabel 4-6 merupakan hasil pengujian kamera pada siang hari pukul 13.45 WIB.

4) Pengujian pada malam hari pukul 22.00 WIB

Respon	Kaca mata	Jarak (cm)	Intensitas (lux)	Keterangan	Gambar
Orang ke-1	Tidak	45	0,6	Objek Tidak Terdeteksi	
Orang ke-2	Ya	45	0,6	Objek Tidak Terdeteksi	

Orang ke-3	Tidak	60	0,6	Objek Tidak Terdeteksi	
Orang ke-4	Ya	60	0,6	Objek Tidak Terdeteksi	

Pada tabel 4-7 merupakan hasil pengujian kamera pada siang hari pukul 13.45 WIB.

c. Analisis Hasil Pengujian

Pada pengujian 4.2.2 pada intensitas cahaya 199,8 lux dan 989,0 lux, dengan memakai atau tidak memakai kacamata, dan dengan jarak 45 cm dan 60 cm kamera dapat mendeteksi objek dengan baik. Namun pada intensitas cahaya 0,6 lux kamera tidak dapat mendeteksi objek dikarenakan kurangnya cahaya di dalam mobil.

3. Pengujian Jarak Pengemudi dengan Kamera dan buzzer

a. Tujuan

Pengujian ini dilakukan untuk mengetahui kinerja sistem dapat berjalan dan mendeteksi objek ketika menunduk di dalam mobil dengan baik dan menguji buzzer menyala atau tidak saat pengemudi menunduk di karenakan kantuk.

b. Skenario Pengujian

Pada pengujian ini dilakukan di dalam mobil tipe Great Corolla tahun 1996, di pengujian ini dilakukan 3 responden dengan ciri-ciri sebagai berikut.

- 1) orang ke-1 laki-laki, umur 22 tahun, dan tinggi badan 165 cm.
- 2) orang ke-2 laki-laki, umur 21 tahun, dan tinggi badan 169 cm.
- 3) orang ke-3 laki-laki, umur 21 tahun, dan tinggi badan 168 cm

dengan memakai dan tidak memakai kacamata, pada siang hari intensitas cahaya diukur dengan menggunakan aplikasi CPU-Z pada ponsel dan mendapat hasil 719,8 lux. Jarak objek dengan kamera kurang dari 25 cm yang akan menandakan kantuk.

1) Pengujian intensitas cahaya

Pukul 14.13 WIB
als-tcs3701 719,8 lux

Pada tabel 4-8 merupakan hasil pengujian intensitas cahaya pada siang hari pukul 14.13 WIB mendapat 719,8 lux.





2) Pengujian jarak



Pada Gambar 4-6 ditentukan jarak kamera dan objek kurang dari 25 cm yang menandakan pengemudi kantuk.

c. Hasil Pengujian

Pengujian pada siang hari pukul 14.13 WIB

Responden	Kacamata	Intensitas (lux)	Buzzer	Hasil Pengujian	Gambar
Orang ke-1	Ya	719,8	ON	Terdeteksi	
	Tidak	719,8	ON	Terdeteksi	
Orang ke-2	Ya	719,8	ON	Terdeteksi	
	Tidak	719,8	ON	Terdeteksi	

Responden	Kacamata	Intensitas (lux)	Buzzer	Hasil Pengujian	Gambar
Orang ke-3	Ya	719,8	ON	Terdeteksi	
	Tidak	719,8	ON	Terdeteksi	

d. Analisis Hasil Pengujian

Hasil pengujian jarak pada intensitas cahaya 719,8 lux dapat mendeteksi pengemudi ketika menunduk dengan baik dan buzzer yang menjadi peringatan bagi pengemudi ketika menunduk juga berjalan dengan baik, namun pada malam hari sistem tidak dapat berjalan dikarenakan pengemudi tidak dapat terdeteksi.

4. Pengujian LED, Buzzer, dan Delay

a. Tujuan

Pengujian ini dilakukan untuk mengetahui kinerja LED dan buzzer dapat berjalan ketika pengemudi menutup mata atau rasio mata dibawah 0.18.

b. Skenario Pengujian

Pada pengujian ini dilakukan di dalam mobil tipe Great Corolla tahun 1996 dengan memakai dan tidak memakai kacamata, pada waktu pagi dan siang

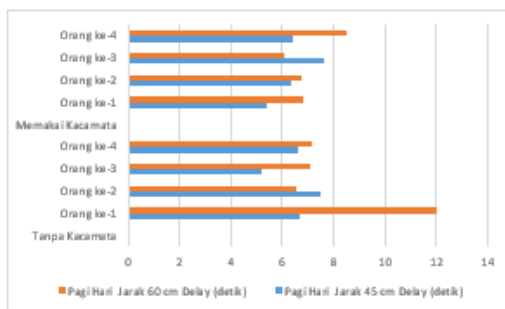
hari. Responden, intensitas cahaya, dan jarak menggunakan hasil pada pengujian 4.2.2.

c. Hasil Pengujian

Pengujian pagi hari pukul 06.30 WIB

Responden	Intensitas 199,8 lux		Buzzer dan LED
	45 cm	60 cm	
Delay (Detik)			
Tanpa Kacamata			
Orang ke-1	06.70	12.02	ON
Orang ke-2	07.49	06.57	ON
Orang ke-3	05.20	07.08	ON
Orang ke-4	06.62	07.12	ON
Memakai Kacamata			
Orang ke-1	05.40	06.84	ON
Orang ke-2	06.31	06.76	ON
Orang ke-3	07.61	06.05	ON
Orang ke-4	06.42	08.52	ON

Pada tabel 4-9 merupakan hasil pengujian buzzer, LED, dan delay di dalam mobil pada pagi hari.



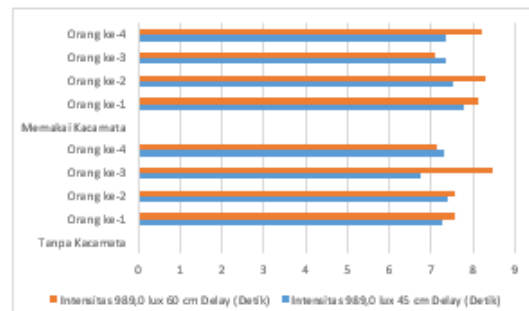
Grafik pengujian pada pagi hari

Pada Gambar 4-7 merupakan hasil grafik dari pengujian Buzzer, LED, dan delay pada pagi hari.

1) Pengujian siang hari pukul 13.45 WIB

Responden	Intensitas 989,0 lux		Buzzer dan LED
	45 cm	60 cm	
Delay (Detik)			
Tanpa Kacamata			
Orang ke-1	07.25	07.55	ON
Orang ke-2	07.41	07.56	ON
Orang ke-3	06.75	08.48	ON
Orang ke-4	07.29	07.12	ON
Memakai Kacamata			
Orang ke-1	07.77	08.13	ON
Orang ke-2	07.54	08.32	ON
Orang ke-3	07.34	07.09	ON
Orang ke-4	07.37	08.20	ON

Pada tabel 4-10 merupakan hasil pengujian buzzer, LED, dan delay di dalam mobil pada pagi hari.



Grafik pengujian pada siang hari

Pada Gambar 4-7 merupakan hasil grafik dari pengujian Buzzer, LED, dan delay pada pagi hari.

d. Analisis Hasil Pengujian

Hasil pengujian LED dan buzzer jarak 45 cm dan 60 cm dengan intensitas cahaya 199,8 lux tanpa kacamata mendapat rata-rata delay 07.35 detik, dan dengan kacamata mendapat rata-rata delay 06.74 detik. Pada jarak 45 cm dan 60 cm dengan intensitas cahaya 989,0 lux tanpa kacamata mendapat rata-rata delay 07.43 detik dan dengan kacamata mendapat rata-rata delay 07.72 detik. Namun pada saat intensitas cahaya 0,6 lux kamera tidak dapat mendeteksi objek karena kurangnya cahaya di dalam mobil.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian pada proyek akhir ini dapat disimpulkan bahwa.

1. Kamera Raspberry dapat mendeteksi objek dengan baik ke responden dengan memakai dan tidak memakai pada intensitas cahaya 199,8 lux dan 989,0 lux. Sedangkan pada intensitas cahaya 0,6 lux kamera tidak dapat mendeteksi objek.
2. Sistem dapat mendeteksi objek pada jarak kurang dari 25 cm dan intensitas cahaya 719,8 lux dengan baik. Namun pada pengujian malam hari objek tidak dapat terdeteksi oleh kamera karena kurangnya cahaya di dalam mobil.
3. Buzzer dan LED dapat berjalan dengan baik pada intensitas cahaya 199,8 lux dan 989,0 lux, sedangkan pada intensitas cahaya 0,6 lux buzzer dan LED tidak dapat berjalan dikarenakan rasio mata pengemudi tidak dapat terdeteksi.

Pada sistem ini masih terdapat kekurangan yang dapat dilihat sebagai berikut.

1. Pada saat pengujian dengan intensitas cahaya 0,6 lux pada kondisi ini kamera Raspberry tidak dapat mendeteksi objek.
2. Posisi pengemudi juga berpengaruh terhadap sistem ini, seperti pengemudi sedang menoleh ke kanan atau kiri dan ke atas atau ke bawah.

B. Saran

Berdasarkan hasil pengujian pada proyek akhir ini untuk kinerja sistem yang lebih maksimal dapat mengganti kamera Raspberry yang mendukung night vision.

REFERENSI

- [1] C. Lakalantas, K. Bersama, J. Raharja, S. Jalan, and R. Lalin, "Cegah Lakalantas, Korlantas Bersama Jasa Raharja Survey Jalan dan Rambu Lalin di Bali," 2022.
- [2] Gloria, "Pakar UGM Sebut Empat Faktor Penyebab Kecelakaan di Jalan Tol," Ugm.Ac.Id, no. November, 2021, [Online]. Available: <https://ugm.ac.id/id/berita/21920-pakar-ugm-sebut-empat-faktor-penyebab-kecelakaan-di-jalan-tol>
- [3] otomotif.kompas.com, "Kenali Dua Musuh Pengendara saat Tempuh Perjalanan Jauh," pp. 1–7, 2019.
- [4] A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON," Sainstech J. Penelit. dan PengkZein, A. (2018). Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. Sainstech J. Penelit. Dan Pengkaj. Sains Dan Teknol. 28(2), 22–26. <https://doi.org/10.37277/stch.v28i2.238a>, vol. 28, no. 2, pp. 22–26, 2018, doi: 10.37277/stch.v28i2.238.
- [5] R. M. B. Yudha Iman Taufik, Yusril Zakaria, Dyah Ayu Setyowati, Rais, "Sistem pendeteksi kantuk menggunakan webcam dan Raspberry Pi".
- [6] F. You, X. Li, Y. Gong, H. Wang, and H. Li, "A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration," IEEE Access, vol. 7, pp. 179396–179408, 2019, doi: 10.1109/ACCESS.2019.2958667.
- [7] H. Suraya, I. Ziad, J. T. Elektro, F. Teknik, and D. Telekomunikasi, "Rancang Bangun Alat Pendeteksi Kantuk Pada Mobil Berbasis IoT Menggunakan Raspberry Pi Dan Kamera," J. Ilm. Komputasi, vol. 20, no. 3, pp. 385–391, 2021, doi: 10.32409/jikstik.20.3.2797.
- [8] H. Kulhandjian, "Detecting Driver Drowsiness with Multi-Sensor Data Fusion Combined with Machine Learning," no. September, 2021, [Online]. Available: https://scholarworks.sjsu.edu/mti_publications/371/
- [9] L. Wei, T. Feng, P. Zhao, and M. Liao, "Driver sleepiness detection algorithm based on relevance vector machine," Balt. J. Road Bridg. Eng., vol. 16, no. 1, pp. 118–139, 2021, doi: 10.7250/bjrbe.2021-16.518.
- [10] F. Javier, "Ada 183 Kecelakaan Sepanjang 2021," Data.Tempo.Co, pp. 1–7, 2021, [Online]. Available: <https://data.tempo.co/data/1253/korlantas-polri-ada-183-kecelakaan-sepanjang-2021-terbanyak-di-wilayah-polda-jatim>
- [11] A. T. Daurina Lestari, "KNKT: 80 Persen Kecelakaan di Tol Akibat Mengantuk dan Letih," 30 Novemb., pp. 1–15, 2021, [Online]. Available: https://www.viva.co.id/berita/nasional/1427758-knkt-80-persen-kecelakaan-di-tol-akibat-mengantuk-dan-letih?page=1&utm_medium=page-1
- [12] A. Riki Rifandi, Sutarti, "Rancang Bangun Kamera Pengawas Menggunakan Raspberry Dengan Aplikasi Telegram Berbasis Internet of Things," J. PROSISKO, vol. 8, no. 1, 2021, [Online]. Available: www.amazone.com
- [13] R. Mardiaty, F. Ashadi, and G. F. Sugihara, "Rancang Bangun Prototipe Sistem Peringatan Jarak Aman pada Kendaraan Roda Empat Berbasis Mikrokontroler ATMEGA32," TELKA - Telekomun. Elektron. Komputasi dan Kontrol, vol. 2, no. 1, pp. 53–61, 2016, doi: 10.15575/telka.v2n1.53-61.
- [14] 2008 28 Lia Kurniawati, FT UI, "Pengaruh Pencahayaan LED," pp. 27–47, 2008.
- [15] <https://en.wikipedia.org/>, "Python-Logo."
- [16] P. Studi et al., "Sistem Embedded Berbasis Raspberry Pi (Pengontrolan Dasar Led, Led Dot-Matrix, Dan Seven Segment)," vol. 8, no. 2, 2021.
- [17] Raspbian.org, "Raspbian Logo."
- [18] Maruf.shidiq, "Operating System (OS) pada Raspberry," <https://otomasi.sv.ugm.ac.id/>, pp. 4–13, 2018, [Online]. Available: <https://otomasi.sv.ugm.ac.id/2018/06/04/operating-system-os-pada-raspberry/>
- [19] D. A. Navastara, W. Y. M. Putra, and C. Fatichah, "Drowsiness Detection Based on Facial Landmark and Uniform Local Binary Pattern," J. Phys. Conf. Ser., vol. 1529, no. 5, 2020, doi: 10.1088/1742-6596/1529/5/052015.