```
import time
from scipy.spatial import distance as dist
from imutils import face_utils
import imutils
import dlib
import cv2
import pygame
import requests

# URL for data submission
login_url = "http://skripsiiot.cloud/rafli/login.php"
upload_url =
"http://skripsiiot.cloud/rafli/upload_drowsiness_data.p
hp"

# Function for login
def login(username, password):
    credentials = {'username': username, 'password':
password}
    session = requests.Session()
    try:
        response = session.post(login_url,
data=credentials)
        if response.status_code == 200:
            print(f"Login successful for user:
{username}")
            return session
        else:
            print(f"Login failed for user: {username}")
            return None
    except requests.exceptions.RequestException as e:
        print(f"Login request failed: {e}")
        return None

# Login for the first user
username = "teguh"
password = "teguh"
session = login(username, password)

# Eye and mouth aspect ratio functions
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    return (A + B) / (2.0 * C)

def mouth_aspect_ratio(mouth):
    A = dist.euclidean(mouth[14], mouth[18])
    B = dist.euclidean(mouth[12], mouth[16])
```

```
    C = dist.euclidean(mouth[0], mouth[6])
    return (A + B) / (2.0 * C)

# Function to send data to the server
def upload_data(session, mata, mulut, kondisi):
    data = {'mata': mata, 'mulut': mulut, 'kondisi':
kondisi}
    try:
        print(f"Uploading data: {data}")
        response = session.post(upload_url, data=data,
timeout=10)
        if response.status_code == 200:
            print("Data uploaded successfully.")
        else:
            print(f"Failed to upload data:
{response.status_code}, {response.text}")
    except requests.exceptions.RequestException as e:
        print(f"Request failed: {e}")

# Thresholds and frame checks
eye_thresh = 0.28
mouth_thresh = 0.5
frame_check = 20

# Initialize dlib's face detector and Landmark
predictor
detector = dlib.get_frontal_face_detector()
predictor =
dlib.shape_predictor("shape_predictor_68_face_Landmarks
.dat")

# Indices for facial Landmarks
(lStart, lEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
(mStart, mEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["mouth"]

# Start video capture
cap = cv2.VideoCapture("WhatsApp Video 2024-07-17 at
19.31.22_96ec2c84.mp4")
flag = 0
alarm_playing = False
last_condition_time = None

# Flags to track if data has been uploaded
drowsiness_uploaded = False
yawning_uploaded = False
```

```
# Initialize pygame for audio alerts
pygame.mixer.init()
pygame.mixer.music.load('alarm.wav')

while True:
    ret, frame = cap.read()
    if not ret:
        break

    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    subjects = detector(gray, 0)

    for subject in subjects:
        shape = predictor(gray, subject)
        shape = face_utils.shape_to_np(shape)
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        mouth = shape[mStart:mEnd]

        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)
        ear = (leftEAR + rightEAR) / 2.0
        mouthMAR = mouth_aspect_ratio(mouth)

        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        mouthHull = cv2.convexHull(mouth)
        cv2.drawContours(frame, [leftEyeHull], -1, (0,
255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0,
255, 0), 1)
        cv2.drawContours(frame, [mouthHull], -1, (0,
255, 0), 1)

        if ear < eye_thresh:
            flag += 1
            if flag >= frame_check:
                if last_condition_time is None:
                    last_condition_time = time.time()
                if time.time() - last_condition_time >=
3 and not drowsiness_uploaded:
                    upload_data(session, ear, mouthMAR,
'Drowsiness')
                    drowsiness_uploaded = True
                    yawning_uploaded = False
                    if not alarm_playing:
                        pygame.mixer.music.play(-1)
```

```
                        alarm_playing = True
                    print("ALERT! Drowsiness
Detected!")

        elif mouthMAR > mouth_thresh:
            flag += 1
            if flag >= frame_check:
                if last_condition_time is None:
                    last_condition_time = time.time()
                if time.time() - last_condition_time >=
3 and not yawning_uploaded:
                    upload_data(session, ear, mouthMAR,
'Yawning')
                    yawning_uploaded = True
                    drowsiness_uploaded = False
                    if not alarm_playing:
                        pygame.mixer.music.play(-1)
                        alarm_playing = True
                    print("ALERT! Yawning Detected!")

        else:
            flag = 0
            last_condition_time = None
            drowsiness_uploaded = False
            yawning_uploaded = False
            if alarm_playing:
                pygame.mixer.music.stop()
                alarm_playing = False

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break

# Cleanup
pygame.mixer.music.stop()
pygame.mixer.quit()
cv2.destroyAllWindows()
cap.release()
```

1. Source Code app.php

```
<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Metadata untuk pengaturan karakter dan viewport -->
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Grafik Sensor Pendeteksi Kantuk</title>
    <!-- Memuat Chart.js untuk membuat grafik -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.7.0/chart.m
in.js"></script>
    <!-- Memuat Bootstrap CSS untuk tata letak dan gaya responsif
-->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/5.3.0/css/bootstrap.min.css" rel="stylesheet">
    <style>
        /* Gaya umum untuk seluruh halaman */
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
            padding-top: 56px; /* Tambahkan padding untuk mencegah
konten tersembunyi di balik navbar */
        }
        .container {
            width: 80%;
            MARgin: auto;
            background-color: #fff;
            border-radius: 8px;
            padding: 20px;
            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
        }
        .chart-container {
            display: flex;
```

```
            flex-direction: column;
            align-items: center;
        }
        .chart {
            width: 100%;
            height: 400px;
        }
        .stats {
            display: flex;
            justify-content: space-around;
            MARgin-top: 20px;
        }
        .stat-item {
            padding: 15px 30px;
            border-radius: 8px;
            display: flex;
            flex-direction: column;
            align-items: center;
            color: #fff;
        }
        .sleep-box {
            background-color: #ffc107; /* Warna untuk Sleep Count
*/
        }
        .yawn-box {
            background-color: #007bff; /* Warna untuk Yawn Count
*/
        }
        .stat-label {
            font-weight: bold;
```

```
            MARgin-bottom: 5px;
        }
        .stat-value {
            font-size: 24px;
        }
        .title {
            font-size: 24px;
            font-weight: bold;
            MARgin-bottom: 20px;
            text-align: center;
        }
        .reset-button {
            MARgin-top: 20px;
            text-align: center;
        }
    </style>
</head>
<body>
    <!-- Navbar -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-
top">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Pendeteksi Kantuk</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
```

```
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link"
href="#realtime">Grafik Real-time</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="#dailyhistory">History Harian</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <!-- Konten Utama -->
    <div class="container">
        <h1 class="title">PENDETEKSI KANTUK REAL-TIME</h1> <!--
Judul di tengah -->
        <div class="stats">
            <div class="stat-item sleep-box"> <!-- Kotak Sleep
Count -->
                <div class="stat-label">Sleep Count:</div>
                <div class="stat-value" id="sleepCount">0</div>
            </div>
            <div class="stat-item yawn-box"> <!-- Kotak Yawn Count
-->
                <div class="stat-label">Yawn Count:</div>
                <div class="stat-value" id="yawnCount">0</div>
            </div>
        </div>
```

```
        <div class="reset-button">
            <button class="btn btn-danger"
onclick="resetCounts()">Reset</button>
        </div>

        <!-- Bagian Grafik Real-time -->
        <div id="realtime" class="chart-container">
            <h2 class="title">Grafik Real-time</h2>
            <div class="chart">
                <canvas id="KantukChart"></canvas>
            </div>
        </div>

        <!-- Bagian Grafik History Harian -->
        <div id="dailyhistory" class="chart-container">
            <h2 class="title">History Harian</h2>
            <div class="chart">
                <canvas id="dailyHistoryChart"></canvas>
            </div>
        </div>
    </div>

    <!-- Memuat Bootstrap JS -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/5.3.0/js/bootstrap.bundle.min.js"></script>

    <script>
        // Inisialisasi grafik
        var ctx =
document.getElementById('KantukChart').getContext('2d');
```

```
        var dailyCtx =
document.getElementById('dailyHistoryChart').getContext('2d');
        var myChart;
        var dailyHistoryChart;

        // Fungsi untuk memperbarui grafik
        function updateChart() {
            fetch('fetch_records.php') // Ambil data dari server
            .then(response => response.json())
            .then(data => {
                if (myChart) {
                    myChart.destroy(); // Hancurkan grafik lama
jika ada
                }
                myChart = new Chart(ctx, {
                    type: 'line', // Tipe grafik adalah garis
                    data: {
                        labels: data.map(record =>
record.timestamp), // Label adalah timestamp
                        datasets: [{
                            label: 'Eye Ratio', // Dataset untuk
rasio mata
                            data: data.map(record =>
record.eye_ratio),
                            backgroundColor: 'rgba(255, 99, 132,
0.2)',
                            borderColor: 'rgba(255, 99, 132, 1)',
                            borderWidth: 2,
                            pointRadius: 4,
                            fill: false
```

```
                }, {
                    label: 'Mouth Ratio', // Dataset untuk
rasio mulut
                    data: data.map(record =>
record.mouth_ratio),
                    backgroundColor: 'rgba(54, 162, 235,
0.2)',
                    borderColor: 'rgba(54, 162, 235, 1)',
                    borderWidth: 2,
                    pointRadius: 4,
                    fill: false
                }]
            },
            options: {
                responsive: true,
                maintainAspectRatio: false,
                scales: {
                    x: {
                        display: true,
                        title: {
                            display: true,
                            text: 'Timestamps'
                        }
                    },
                    y: {
                        display: true,
                        title: {
                            display: true,
                            text: 'Ratios'
                        }
```

```
                        }
                    }
                }
            });

            // Perbarui hitungan sleep dan yawn
            document.getElementById('sleepCount').innerText =
data.filter(record => record.condition === 'sleep').length;
            document.getElementById('yawnCount').innerText =
data.filter(record => record.condition === 'yawn').length;
        });
    }

    // Fungsi untuk memperbarui grafik histori per hari
    function updateDailyHistoryChart() {
        fetch('getDailyHistoryData.php') // Ambil data dari
server
        .then(response => response.json())
        .then(data => {
            if (dailyHistoryChart) {
                dailyHistoryChart.destroy(); // Hancurkan
grafik lama jika ada
            }
            dailyHistoryChart = new Chart(dailyCtx, {
                type: 'bar', // Tipe grafik adalah batang
                data: {
                    labels: data.dates, // Label adalah
tanggal
                    datasets: [{
```

```
                label: 'Daily Sleep Count', // Dataset
untuk hitungan sleep harian
                data: data.dailySleepCounts,
                backgroundColor: 'rgba(255, 206, 86,
0.2)',
                borderColor: 'rgba(255, 206, 86, 1)',
                borderWidth: 2
            }, {
                label: 'Daily Yawn Count', // Dataset
untuk hitungan yawn harian
                data: data.dailyYawnCounts,
                backgroundColor: 'rgba(75, 192, 192,
0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 2
            }]
        },
        options: {
            responsive: true,
            maintainAspectRatio: false,
            scales: {
                x: {
                    display: true,
                    title: {
                        display: true,
                        text: 'Dates'
                    }
                },
                y: {
                    display: true,
```

```
                                    title: {
                                        display: true,
                                        text: 'Counts'
                                    }
                                }
                            }
                        }
                    });
                });
            }

            // Fungsi untuk mereset hitungan
            function resetCounts() {
                fetch('reset_database.php', { method: 'POST' }) //
Kirim permintaan POST untuk mereset database
                .then(response => response.json())
                .then(data => {
                    if (data.status === 'success') {

document.getElementById('sleepCount').innerText = '0';
                        document.getElementById('yawnCount').innerText
= '0';
                        updateChart(); // Perbarui grafik setelah
reset
                    } else {
                        alert('Failed to reset the database: ' +
data.message); // Tampilkan pesan kesalahan jika gagal
                    }
                });
            }
```

```
        // Panggil fungsi updateChart dan updateDailyHistoryChart
untuk pertama kali
        updateChart();
        updateDailyHistoryChart();

        // Atur interval untuk memperbarui grafik setiap 5 detik
(5000 ms)
        setInterval(updateChart, 5000);
        setInterval(updateDailyHistoryChart, 86400000); //
Perbarui grafik histori setiap 24 jam (86400000 ms)
    </script>
</body>
</html>
```

2. GetDailyHistory.php

```
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Connect to SQLite database
$db = new SQLite3('/home/pi/sleep_yawn_records.db');

// Query to get daily sleep and yawn counts
$results = $db->query('SELECT date(timestamp) as date,
                        SUM(case when condition = "sleep" then 1
else 0 end) as dailySleepCounts,
                        SUM(case when condition = "yawn" then 1
else 0 end) as dailyYawnCounts
```

```
                        FROM sleep_yawn_records
                        GROUP BY date');

// Fetch all records as an associative array
$dailyHistory = ['dates' => [], 'dailySleepCounts' => [],
'dailyYawnCounts' => []];
while ($row = $results->fetchArray(SQLITE3_ASSOC)) {
    $dailyHistory['dates'][] = $row['date'];
    $dailyHistory['dailySleepCounts'][] =
$row['dailySleepCounts'];
    $dailyHistory['dailyYawnCounts'][] = $row['dailyYawnCounts'];
}

// Close the database connection
$db->close();

// Return the records as JSON
header('Content-Type: application/json');
echo json_encode($dailyHistory);
?>
```

3. Fetch_Record.php

```
<?php
header('Content-Type: application/json');

// Koneksi ke basis data SQLite
$dsn = 'sqlite:/home/pi/sleep_yawn_records.db';
try {
    $pdo = new PDO($dsn);
} catch (PDOException $e) {
```

```
    echo json_encode(['error' => $e->getMessage()]);
    exit();
}

// Query untuk mengambil data dari tabel
$query = "SELECT timestamp, eye_ratio, mouth_ratio, condition FROM
sleep_yawn_records";
$stmt = $pdo->query($query);

$records = [];
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $records[] = $row;
// Mengembalikan data dalam format JSON
echo json_encode($records);
?>s
```