

## LAMPIRAN

### Lampiran Kode *Desktop*

```
import sys
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *

from ultralytics import YOLO
from gtts import gTTS
from datetime import datetime
from mysql import connector
from ping3 import ping

import cv2, socket, pandas as pd, numpy as np, pygame, os,
threading, telebot, time, openpyxl

app = QApplication(sys.argv)
window = QWidget()
window.setWindowTitle('Alarm Kebakaran Online')
window.setFixedSize(800,500)
screen_geometry = QDesktopWidget().screenGeometry()
center_x = (screen_geometry.width() - window.width()) // 2
center_y = (screen_geometry.height() - window.height()) // 2
window.setGeometry(center_x, center_y, window.width(),
window.height())

#
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
# ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////// custom widgets
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

def customFrame():
    frame = QFrame(window)
    frame.setFixedSize(window.width(), window.height())

    colored_label = QLabel(frame)
    colored_label.setFixedSize(window.width(), window.height())
    colored_label.setStyleSheet('QLabel{background: darkblue;}')

    pixmap = QPixmap('fire_background.png')
    colored_label.setPixmap(pixmap)
    colored_label.setScaledContents(True)

    return frame

def customTextfield(text,placeholder, parent):
    textfield = QLineEdit(text, parent)
```

```

        textfield.setPlaceholderText(placeholder)
        textfield.setStyleSheet('QLineEdit{font-size:15px; border-
radius:10px; border:1px solid black;}')
        textfield.setFixedHeight(30)
        return textfield

def customButton(text, parent):
    button = QPushButton(text, parent)
    button.setStyleSheet('QPushButton{border-radius:10px;
border:1px solid black; color: white; background: grey;}')
    button.setFixedSize(100,30)
    return button

def customLabel(text, parent):
    label = QLabel(text, parent)
    label.setStyleSheet('QLabel{font-size:15px; color: white}')
    return label

def customTitleLabel(text, parent):
    label = QLabel(text, parent)
    label.setStyleSheet('QLabel{font-size:30px; color: white}')
    return label

def customImageLabel(text, parent):
    image_label = QLabel(text, parent)
    image_label.setStyleSheet('QLabel{background-color: black;
color:white;}')
    image_label.setAlignment(Qt.AlignCenter)
    return image_label

def customCheckBox(text, parent):
    checkbox = QCheckBox(text, parent)
    checkbox.setStyleSheet('QCheckBox{color: white; font-size:
20px;}')
    return checkbox

#
#####
#####
# ##### login frame
#####
#
#####
#####

loginframe = customFrame()

loginLabel = customTitleLabel(text='Login', parent=loginframe)
loginLabel.move(335,100)

usernameTextfield = customTextfield(text='',
placeholder='username', parent=loginframe)
usernameTextfield.move(300,150)

passwordTextfield = customTextfield(text='',
placeholder='password',parent=loginframe)
passwordTextfield.setEchoMode(QLineEdit.Password)
passwordTextfield.move(300,200)

```

```

loginButton = customButton(text='Login', parent=loginframe)
loginButton.move(320,250)

def loginButton_clicked():

    try:
        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        except Exception as e:
            QMessageBox.warning(window, 'Login Failed!', 'Username
atau Password Salah!!')
        else:
            editframe.setVisible(True)
            fetchKamera()
            fetchLogs()
            conn.close()
loginButton.clicked.connect(loginButton_clicked)

#
//////////////////////////////////////
//////////////////////////////////////
# //////////////////////////////////////// edit frame
//////////////////////////////////////
#
//////////////////////////////////////
//////////////////////////////////////
kelas = None
link_kamera = None

editframe = customFrame()

editlabel = customTitleLabel(text='Edit', parent=editframe)
editlabel.move(10,10)

kelasTextfield = customTextfield(text='', placeholder='nama kelas',
parent=editframe)
kelasTextfield.setGeometry(10,50,320,30)

kameraTextfield = customTextfield(text='', placeholder='link RTSP
kamera', parent=editframe)
kameraTextfield.setGeometry(10,100,320,30)

insertButton = customButton(text='Insert', parent=editframe)
insertButton.move(10,150)
def insertButton_clicked():
    print('edit insert button was clicked')

    if kelasTextfield.text()==' ' and kameraTextfield.text()==' ':
        QMessageBox.warning(window, 'Kosong!', 'isi kotak nama
kelas dan link RTSP kamera untuk mengisi tabel.')
    else:

```

```

        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        cursor = conn.cursor()

        cursor.execute('insert into kamera (kelas, link_kamera)
values (%s,%s)', (kelasTextfield.text(),kameraTextfield.text()))
        conn.commit()
        fetchKamera()
        conn.close()
insertButton.clicked.connect(insertButton_clicked)

deleteButton = customButton(text='Delete', parent=editframe)
deleteButton.move(120, 150)
def deleteButton_clicked():
    print('edit delete button was clicked')

    if kelasTextfield.text()==' ' and kameraTextfield.text()==' ':
        QMessageBox.warning(window, 'Kosong!', 'mohon pilih salah
satu data sebelum dihapus.')
    else:
        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        cursor = conn.cursor()

        # cursor.execute('insert into kamera (kelas, link_kamera)
values (%s,%s)', (kelasTextfield.text(),kameraTextfield.text()))
        cursor.execute('delete from kamera where kelas = %s',
(kelasTextfield.text(),))
        conn.commit()
        fetchKamera()
        conn.close()
deleteButton.clicked.connect(deleteButton_clicked)

changeButton = customButton(text='Change', parent=editframe)
changeButton.move(230, 150)

def changeButton_clicked():
    global kelas, link_kamera # Declare them as global
    print('edit change button was clicked')

    if kelasTextfield.text() == ' ' and kameraTextfield.text() ==
' ':
        QMessageBox.warning(window, 'Kosong!', 'Mohon pilih salah
satu data sebelum diubah.')
    else:
        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        cursor = conn.cursor()

        cursor.execute('UPDATE kamera SET kelas=%s, link_kamera=%s
WHERE kelas=%s AND link_kamera=%s',

```

```

        (kelasTextfield.text(),
kameraTextfield.text(), kelas, link_kamera))
        conn.commit()
        fetchKamera()
        conn.close()

changeButton.clicked.connect(changeButton_clicked)

stop_event = threading.Event()

startDetect_button = customButton(text='Start Detect',
parent=editframe)
startDetect_button.move(10,190)
startDetect_button.setFixedSize(320,100)

def read_image_file(file_path):
    with open(file_path, 'rb') as file:
        return file.read()

def start_detect_thread():
    global stop_event

    print('Detect button was clicked')

    HOST = '0.0.0.0'
    PORT = 8080

    # Initialize YOLO model outside the loop
    model = YOLO('last.pt')
    # source = 'person_fire.jpg'

    # Initialize socket outside the loop
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as
server_socket:
        server_socket.bind((HOST, PORT))
        server_socket.listen(100)
        print(f"Listening on {HOST}:{PORT}")

        while not stop_event.is_set():
            client_socket, addr = server_socket.accept()
            with client_socket:
                print(f"Connected by {addr}")
                data = client_socket.recv(1024)
                if not data:
                    break

                input_sensor = data.decode('utf-8')

                if input_sensor == 'No Smoke':
                    pass
                else:
                    # print(input_sensor)
                    conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
                    cursor = conn.cursor()

```

```

        cursor.execute('SELECT link_kamera FROM kamera
where kelas=%s', (input_sensor,))
        kamera_data = cursor.fetchone()

        cursor.close()
        conn.close()

    if kamera_data:
        camera_link = kamera_data[0]
        # camera_link = 'video copy.mp4' # test
        print('camera link is ', camera_link)

        cap = cv2.VideoCapture(camera_link)
        if not cap.isOpened():
            print('cant open the file')

        start_time = time.time()

        while True:
            if time.time() - start_time > 10:
                print('10 seconds have passed')
                break

            ret, frame = cap.read()
            if not ret:
                print('video completed')
                break

            results = model(frame, conf=0.1,
stream=True, save=False, show=False)
            print(results)

            for r in results:
                class_id = r.boxes.cls
                print(class_id)

                fire_found = 0 in class_id # index
class api
index class orang
                person_found = 1 in class_id #
class_id_numpy =
class_id.cpu().numpy()
                print(class_id_numpy)
                count_person =
np.sum(class_id_numpy == 1)

                if fire_found:

cv2.imwrite('saved_frame.jpg', frame)
                    cap.release()
                    cv2.destroyAllWindows()

                    pixmap =
QPixmap('saved_frame.jpg')
                    videolabel.setPixmap(pixmap)

videolabel.setScaledContents(True)

```

```

input_sensor + '.'
str(count_person) + ' orang di dalam' if person_found else ''
fire_text = 'kebakaran di ' +
people_text = ' masih ada ' +
text = fire_text + people_text
print(text)

conn =
connector.connect(host='localhost', user=usernameTextfield.text(),
password=passwordTextfield.text(), database='kebakaran')
cursor = conn.cursor()

image_source =
image_source =
'saved_frame.jpg'
read_image_file(image_source)
today =
datetime.now().strftime("%Y-%m-%d %H:%M:%S")

cursor.execute('insert into
logs (tanggal, keterangan, gambar) values (%s,%s,%s)', (today,
text, image_source))
conn.commit()
fetchLogs()
conn.close()

bot_text = today+' | '+text
try:
bot =
telebot.TeleBot('6342314925:AAHg4vI0InZvqwAk92OTymWpoSsq-tm805Y')
with
open('saved_frame.jpg','rb') as photo:
bot.send_photo('-
1002132512911', photo)
bot.send_message('-
1002132512911', bot_text)
except Exception as e:
print(e)
pass

language = 'id'
speech = gTTS(text=text,
lang=language, slow=False)

speech.save("output.mp3")

for i in range(2):
pygame.mixer.init()
pygame.mixer.music.load("output.mp3")
pygame.mixer.music.play()
while
pygame.mixer.music.get_busy():
pygame.time.Clock().tick(10)

pygame.quit()
i = i + 1

```

```

os.remove('output.mp3')
os.remove('saved_frame.jpg')
print('deleted')
break

stop_event.clear()

def startDetect_button_clicked():
    print('start streaming button was clicked')
    startDetect_button.setVisible(False)
    stopDetect_button.setVisible(True)

    # Create a thread for the detection process
    detect_thread = threading.Thread(target=start_detect_thread)
    detect_thread.start()

# Connect the button click event to the function
startDetect_button.clicked.connect(startDetect_button_clicked)

stopDetect_button = customButton(text='Stop Detect',
parent=editframe)
stopDetect_button.move(10, 190)
stopDetect_button.setFixedSize(320,100)
stopDetect_button.setVisible(False)

def stopDetect_button_clicked():
    print('stop streaming button was clicked')
    startDetect_button.setVisible(True)
    stopDetect_button.setVisible(False)

    stop_event.set()

# Connect the button click event to the function
stopDetect_button.clicked.connect(stopDetect_button_clicked)

videolabel = customImageLabel(text='video label', parent=editframe)
videolabel.setGeometry(340,10,450,280)

kameraTable = QTableWidgetItem(0, 2, editframe)
kameraTable.setHorizontalHeaderLabels(['kelas', 'link_kamera'])
kameraTable.setColumnWidth(1, 660)

def fetchKamera():
    conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
    cursor = conn.cursor()

    # Use parameterized query to prevent SQL injection
    cursor.execute('SELECT * FROM kamera')

    kameraData = cursor.fetchall()
    kameraTable.setRowCount(len(kameraData))
    for row, row_data in enumerate(kameraData):
        for col, cell_data in enumerate(row_data):
            item = QTableWidgetItem(cell_data)
            kameraTable.setItem(row, col, item)

```



```

conn.close()

kameraTable.setGeometry(10, 300, 780, 190)

def nextFrame():
    ret, frame = cap.read()
    if ret:
        rgbImage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        h, w, ch = rgbImage.shape
        bytesPerLine = ch * w
        qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
        pixmap = QPixmap.fromImage(qImg)
        videolabel.setPixmap(pixmap)
        videolabel.setScaledContents(True)

def stopVideo():
    timer.stop()
    cap.release()
    videolabel.clear()

def kameraTable_itemclicked(item):
    global kelas, link_kamera, cap, timer

    row = item.row()
    kelas = kameraTable.item(row, 0).text()
    link_kamera = kameraTable.item(row, 1).text()
    kelasTextField.setText(kelas)
    kameraTextField.setText(link_kamera)

    # url =
'rtsp://admin:aaaaaaa1@192.168.1.234/Streaming/Channels/202'
    url = link_kamera

    start_index = url.find('@') + 1
    end_index = url.find('/', start_index)

    ip_address = url[start_index:end_index]

    print("IP Address:", ip_address)

    # ip_address = '192.168.1.234'

    # ping_result = ping(ip_address)
    # if ping_result == False:
    #     print ('pinging '+ip_address+' failed')
    #     QMessageBox.warning(window, 'Koneksi Putus!', 'mohon
pastikan octv dan komputer terhubung dengan router.')
    #     return
    # else:
    #     # cap = cv2.VideoCapture('video.mp4')
    #     cap = cv2.VideoCapture(link_kamera)
    #     timer = QTimer()
    #     timer.timeout.connect(nextFrame)
    #     timer.start(1000/30) # 30 fps

```

```

# QTimer.singleShot(5000, stopVideo) # Stop after 5
seconds

cap = cv2.VideoCapture(link_kamera)
timer = QTimer()
timer.timeout.connect(nextFrame)
timer.start(1000/30) # 30 fps
QTimer.singleShot(20000, stopVideo) # Stop after 20 seconds

kameraTable.itemClicked.connect(kameraTable_itemclicked)

logButton=customButton(text='Logs', parent=editframe)
logButton.move(230,10)
def logButton_clicked():
    print('edit log button was clicked')
    logframe.setVisible(True)
logButton.clicked.connect(logButton_clicked)

editframe.setVisible(False)

#
////////////////////////////////////
////////////////////////////////////
# ////////////////////////////////// logs frame
////////////////////////////////////
#
////////////////////////////////////
////////////////////////////////////

logframe = customFrame()
logframe.setVisible(False)

loglabel = customTitleLabel(text='Logs', parent=logframe)
loglabel.move(10,10)

imageLabel = customImageLabel(text='gambar', parent=logframe)
imageLabel.setGeometry(210,50,375,240)

logTable = QTableWidgetItem(0, 2, logframe)
logTable.setHorizontalHeaderLabels(['tanggal', 'keterangan'])
logTable.setColumnWidth(0, 130)
logTable.setColumnWidth(1, 600)

tanggal = None

def fetchLogs():
    conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
    cursor = conn.cursor()

    # Use parameterized query to prevent SQL injection
    cursor.execute('SELECT tanggal, keterangan FROM logs')

    logData = cursor.fetchall()

```

```

logTable.setRowCount(len(logData))
for row, row_data in enumerate(logData):
    for col, cell_data in enumerate(row_data):
        item = QTableWidgetItem(cell_data)
        logTable.setItem(row, col, item)

conn.close()

logTable.setGeometry(10, 300, 780, 190)

def logTable_clicked(item):
    global tanggal

    row = item.row()
    tanggal = logTable.item(row, 0).text()

    conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
    cursor = conn.cursor()

    # Use parameterized query to prevent SQL injection
    cursor.execute('SELECT gambar FROM logs where tanggal=%s',
(tanggal,))
    gambar_data = cursor.fetchone()

    if gambar_data:
        # Assuming `imageLabel` is an instance of QLabel
        pixmap = QPixmap()
        pixmap.loadFromData(gambar_data[0]) # Assuming that
gambar_data is a tuple with a single element (blob data)
        imageLabel.setPixmap(pixmap)
        imageLabel.setScaledContents(True) # Optional: Scale the
image to fit the QLabel

    conn.close()

logTable.itemClicked.connect(logTable_clicked)

logs_deleteButton = customButton(text='Delete One',
parent=logframe)
logs_deleteButton.move(390, 10)
def logs_deleteButton_clicked():
    print('logs delete button was clicked')
    global tanggal

    if tanggal is None or tanggal == '':
        QMessageBox.warning(window, 'Kosong!', 'mohon pilih salah
satu data sebelum dihapus.')
    else:
        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        cursor = conn.cursor()

        cursor.execute('delete from logs where tanggal = %s',
(tanggal,))
        conn.commit()

```

```

        fetchLogs()
        conn.close()
        imageLabel.clear()
        imageLabel.setText('gambar')

logs_deleteButton.clicked.connect(logs_deleteButton_clicked)

all_deleteButton = customButton(text='Delete All', parent=logframe)
all_deleteButton.move(500, 10)
def all_deleteButton_clicked():
    answer = QMessageBox.question(window, "Confirmation", 'delete
everything from table logs?', QMessageBox.Yes | QMessageBox.No)
    if answer == QMessageBox.Yes:
        conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')
        cursor = conn.cursor()

        cursor.execute('truncate table logs')
        conn.commit()
        fetchLogs()
        conn.close()
        imageLabel.clear()
        imageLabel.setText('gambar')
    else:
        pass

all_deleteButton.clicked.connect(all_deleteButton_clicked)

exportButton = customButton(text='Export', parent=logframe)
exportButton.move(200,10)
exportButton.setFixedSize(180,30)
def exportButton_clicked():
    print('export button was clicked')
    conn = connector.connect(host='localhost',
user=usernameTextfield.text(), password=passwordTextfield.text(),
database='kebakaran')

    # Fetch data from the logs table
    query = 'SELECT tanggal, keterangan FROM logs'
    df = pd.read_sql_query(query, conn)

    # Ask the user for the file path to save the Excel file
    file_dialog = QFileDialog()
    file_path, _ = file_dialog.getSaveFileName(None, "Save Excel
File", "", "Excel Files (*.xlsx)")

    if file_path:
        # Save the DataFrame to an Excel file
        df.to_excel(file_path, index=False, engine='openpyxl')

        print(f'Logs exported to: {file_path}')

    conn.close()
exportButton.clicked.connect(exportButton_clicked)

```

```
backButton = customButton(text='Back', parent=logframe)
backButton.move(90,10)
def backButton_clicked():
    print('back button was pressed')
    logframe.setVisible(False)
backButton.clicked.connect(backButton_clicked)

window.show()
# exit()

# def on_exit():
#     stop_event.set()
#     stop_event.clear()

app.aboutToQuit.connect(stopDetect_button_clicked)
sys.exit(app.exec_())
# sys.exit()
```



## Lampiran Kode Web

```
<!-- index.php -->
```

```
<?php
// Redirect to login.php
header("Location: login.php");
exit; // Ensure that script execution stops after redirection
?>
```

```
<!-- login.php -->
```

```
<?php
// Database connection settings
session_start();

$host = "localhost";
$dbname = "kebakaran"; // Change this to your database name
$username = "root"; // Default username for XAMPP
$password = ""; // Default password for XAMPP

// $host = "localhost";
// $dbname = "id21895190_kebakaran"; // Change this to your
database name
// $username = "id21895190_root"; // Default username for XAMPP
// $password = "Kebakaran01!"; // Default password for XAMPP

$_SESSION["host"] = $host;
$_SESSION["database"] = $dbname;
$_SESSION["username"] = $username;
$_SESSION["password"] = $password;

// Create connection
$conn = new mysqli($host, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get username and password from the form
    $username = $_POST['username'];
    $password = $_POST['password'];

    // SQL query to fetch user from database
    $sql = "SELECT * FROM users WHERE user='$username' AND
pass='$password'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // User found, login successful
        $_SESSION['loggedin'] = true;
        header('Location: logs.php');
    }
}
```

```

        exit();
    } else {
        // No user found, login failed
        echo '<script
type="text/javascript">document.getElementsByClassName("warning_cl
ass")[0].style.display = "block";</script>';
    }
}

// Close connection
$conn->close();
?>

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-size: cover; /* Cover the entire background
*/
            background-image: url('fire_cartoon.jpg'); /* Set
background image */
        }
        .login-container
        { background-color:
        #fff; border-radius:
        8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        padding: 20px;
        width: 300px;
        }
        h2 {
            text-align: center;
            margin-bottom: 20px;
        }
        label {
            font-weight: bold;
        }
        input[type="text"],
        input[type="password"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-sizing: border-box;
        }
        input[type="submit"]
        { width: 100%;
        background-color: #4caf50;

```

```

padding: 10px;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 16px;
}
input[type="submit"]:hover
{ background-color:
#45a049;
}
.warning_class
{ display: none;
color: red;
margin-bottom: 10px;
}
}
</style>
</head>
<body>
<div class="login-container">
<h2>Login Form</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
<label class="warning_class">Userame or Password
Salah!</label>
<label for="username">Username:</label><br>
<input type="text" id="username" name="username"><br>
<label for="password">Password:</label><br>
<input type="password" id="password"
name="password"><br><br>
<input type="submit" value="Login">
</form>
</div>
</body>

```

```

<!-- logs.php -->

<?php
session_start();

// Check if the user is logged in, otherwise redirect to the login
page
if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] !==
true) {
    header('Location: login.php');
    exit;
}

// Database connection settings
$host = $_SESSION["host"];
$dbname = $_SESSION["database"];
$username = $_SESSION["username"];
$password = $_SESSION["password"];

// Create connection
$conn = new mysqli($host, $username, $password, $database);

// Check connection

```



```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch data from the 'logs' table
$sql = "SELECT tanggal, keterangan, gambar FROM logs ORDER BY
tanggal DESC";
$result = $conn->query($sql);

?>

<!DOCTYPE html>
<html>
<head>
    <title>Logs</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boots
trap.min.css">
    <style>
        button{
            color: white;
        }
        body {
            background-image: url('fire_cartoon.jpg'); /* Set
background image */
            color: white;
            margin: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            background-color: rgba(0, 0, 0, 0.5); /* Black color
with 50% opacity */
        }
        th{
            padding: 8px;
            text-align: left;
            border-bottom: 1px solid #ddd;
            color: black;
        }
        td {
            padding: 8px;
            text-align: left;
            border-bottom: 1px solid #ddd;
            background-color: rgba(0, 0, 0, 0.5); /* Black color
with 50% opacity */
        }
        th {
            background-color: #f2f2f2;
        }
        .image-cell img {
            max-width: 100px;
            max-height: 100px;
        }
    </style>
</head>
<body>

```

```

<h2>Logs</h2>
<table>
  <tr>
    <th>Tanggal</th>
    <th>Keterangan</th>
    <th>Gambar</th>
    <th>Actions</th>
    <!-- Add more table headers if needed -->
  </tr>
  <?php
  if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc())
    { echo "<tr>";
      echo "<td>" . $row["tanggal"] . "</td>";
      echo "<td>" . $row["keterangan"] . "</td>";
      // Display the image if 'gambar' column contains
image data
      echo "<td class='image-cell'><img
src='data:image/jpeg;base64,'" . base64_encode($row['gambar']) . "'
/></td>";
      // Add "View" and "Delete" buttons
      echo "<td>
        <button class='btn btn-info'
onclick=\"window.open('view.php?tanggal=" . $row['tanggal'] . "',
'_blank')\">View</button> |
        <button class='btn btn-danger'
onclick=\"window.location.href='delete.php?tanggal=" .
$row['tanggal'] . "'\">Delete</button>
        </td>";
      echo "</tr>";
    }
  } else {
    echo "<tr><td colspan='4'>0 results</td></tr>";
  }
  ?>
</table>
</body>
</html>

<?php
// Close connection
$conn->close();
?>

```

```

<!-- view.php -->

<?php
session_start();

// Check if the user is logged in, otherwise redirect to the login
page
if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] !==
true) {
  header('Location: login.php');
  exit;
}

```

```

// Check if the 'tanggal' parameter is set in the URL
if (isset($_GET['tanggal'])) {
    $tanggal = $_GET['tanggal'];

    // Database connection settings
    $host = $_SESSION["host"];
    $database = $_SESSION["database"];
    $username = $_SESSION["username"];
    $password = $_SESSION["password"];

    // Create connection
    $conn = new mysqli($host, $username, $password, $database);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Fetch the image data from the 'logs' table based on the
    provided 'tanggal'
    $sql = "SELECT gambar FROM logs WHERE tanggal = '$tanggal'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // Retrieve the image data
        $row = $result->fetch_assoc();
        $imageData = $row['gambar'];

        // Set the appropriate header to display the image
        header("Content-type: image/jpeg");
        echo $imageData;
    } else {
        echo "Image not found";
    }

    // Close connection
    $conn->close();
} else {
    echo "Invalid request";
}
?>

```

```

<!-- delete.php -->

<?php
session_start();

// Check if the user is logged in, otherwise redirect to the login
page
if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] !==
true) {
    header('Location: login.php');
    exit;
}

```

```
// Check if the 'tanggal' parameter is set in the URL
if (isset($_GET['tanggal'])) {
    $tanggal = $_GET['tanggal'];

    // Database connection settings
    $host = $_SESSION["host"];
    $database = $_SESSION["database"];
    $username = $_SESSION["username"];
    $password = $_SESSION["password"];

    // Create connection
    $conn = new mysqli($host, $username, $password, $database);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Perform deletion query
    $sql = "DELETE FROM logs WHERE tanggal = '$tanggal'";
    if ($conn->query($sql) === TRUE) {
        echo "Record deleted successfully";
    } else {
        echo "Error deleting record: " . $conn->error;
    }

    // Close connection
    $conn->close();

    // Redirect back to logs.php
    header('Location: logs.php');
    exit;
} else {
    echo "Invalid request";
}
?>
```

## Lampiran Kode ESP32

```
#include <WiFi.h>

const char *ssid = "TOTOLINK N300RT";
const char *password = "totolink";
const char *desktopIPs[] = {"192.168.1.101", "192.168.1.102",
"192.168.1.103", "192.168.1.104"};
const int desktopPort = 8080; // Choose a port number for
communication

void setup()
{
  Serial.begin(115200);
  connectToWiFi();
}

void loop() {
  int asapValue = analogRead(34); //pin analog sensor asap
  int apiValue = analogRead(35); //pin analog api

  Serial.print("asap : ");
  Serial.println(asapValue);
  Serial.print("api : ");
  Serial.println(apiValue);
  if (asapValue > 500 or apiValue < 500)
    { sendMessageToDesktop("ruang 1 lantai 1", desktopIPs,
sizeof(desktopIPs) / sizeof(desktopIPs[0]));
  }
  delay(1000);
}

void connectToWiFi()
{
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
    { delay(1000);
      Serial.println("Connecting to WiFi...");
    }
  Serial.println("Connected to WiFi");
}

void sendMessageToDesktop(const char *message, const char
**desktopIPs, int numIPs)
{
  WiFiClient client;

  for (int i = 0; i < numIPs; i++) {
    if (client.connect(desktopIPs[i], desktopPort))
      { Serial.print("Sending message to desktop at IP
"); Serial.print(desktopIPs[i]);
        Serial.println("...");

        // Send the message to the Python server
        client.print(message);

        Serial.println("Message sent to desktop");

        client.stop();
      }
  }
}
```

```
        return; // Exit the function after successfully sending the
message
    }
    // else {
    //     Serial.print("Failed to connect to desktop at IP ");
    //     Serial.println(desktopIPs[i]);
    // }

    delay(100); // Optional delay between attempts to avoid
overwhelming the network
}

// If no successful connection is made, you can add a message or
additional handling here.
Serial.println("Failed to connect to any desktop");
}
```



## Lampiran Sumber Informasi

Kebakaran SMAN 6

<https://megapolitan.kompas.com/read/2023/09/30/08335791/kebakaran-di-sman-6-jakarta-yang-merenggut-nyawa-sekuriti-tewas-keracunan>

Kebakaran SMAN 112

<https://megapolitan.kompas.com/read/2023/03/07/13352201/murid-sman-112-jakarta-berlarian-selamatkan-diri-saat-api-membakar>

Kebakaran SDN 12

<https://news.detik.com/berita/d-6330404/kebakaran-di-sdn-12-cilegon-hanguskan-3-ruangan>

Kebakaran SMPN 135

<https://news.detik.com/berita/d-1732241/5-kantin-gudang-smp-135-duren-sawit-diamuk-si-jago-merah>

*How PIRs Work*

<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-Work>

## Lampiran Hasil Tes Turnitin

Fajar Ardli Syahputra\_IMPLEMENTASI SISTEM INTERNET OF THINGS ALARM KEBAKARAN TEXT TO VOICE MENGGUNAKAN METODE OBJECT DETECTION YOLOv8

### ORIGINALITY REPORT

<b>24%</b> SIMILARITY INDEX	<b>21%</b> INTERNET SOURCES	<b>7%</b> PUBLICATIONS	<b>15%</b> STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	------------------------------

### PRIMARY SOURCES

<b>1</b>	<b>Submitted to University of Dundee</b> Student Paper	<b>2%</b>
<b>2</b>	<b>repository.unpas.ac.id</b> Internet Source	<b>2%</b>
<b>3</b>	<b>adoc.pub</b> Internet Source	<b>1%</b>
<b>4</b>	<b>Submitted to Universitas Pamulang</b> Student Paper	<b>1%</b>
<b>5</b>	<b>repository.radenfatah.ac.id</b> Internet Source	<b>1%</b>
<b>6</b>	<b>ejurnal.its.ac.id</b> Internet Source	<b>1%</b>
<b>7</b>	<b>repository.its.ac.id</b> Internet Source	<b>1%</b>
<b>8</b>	<b>jurnal.darmaagung.ac.id</b> Internet Source	<b>1%</b>

Submitted to Universitas Muria Kudus



9	Student Paper	1 %
10	github.com Internet Source	1 %
11	repository.mercubuana.ac.id Internet Source	1 %
12	dspace.uii.ac.id Internet Source	1 %
13	eprints.uniska-bjm.ac.id Internet Source	1 %
14	text-id.123dok.com Internet Source	1 %
15	docplayer.info Internet Source	<1 %
16	id.123dok.com Internet Source	<1 %
17	Submitted to Universitas Darma Persada Student Paper	<1 %
18	repository.ub.ac.id Internet Source	<1 %
19	123dok.com Internet Source	<1 %
20	Submitted to The Open University of Hong Kong	<1 %

Student Paper

---

21	<a href="http://jtiik.ub.ac.id">jtiik.ub.ac.id</a> Internet Source	<1 %
22	Submitted to Purdue University Student Paper	<1 %
23	<a href="http://ilmuteknik.org">ilmuteknik.org</a> Internet Source	<1 %
24	<a href="http://repository.ubharajaya.ac.id">repository.ubharajaya.ac.id</a> Internet Source	<1 %
25	<a href="http://jurnal.ikhafi.or.id">jurnal.ikhafi.or.id</a> Internet Source	<1 %
26	Submitted to Kennesaw State University Student Paper	<1 %
27	<a href="http://kc.umn.ac.id">kc.umn.ac.id</a> Internet Source	<1 %
28	Submitted to STT PLN Student Paper	<1 %
29	<a href="http://etd.umy.ac.id">etd.umy.ac.id</a> Internet Source	<1 %
30	<a href="http://jim.unindra.ac.id">jim.unindra.ac.id</a> Internet Source	<1 %
31	<a href="http://laptopnhapkhau.vn">laptopnhapkhau.vn</a> Internet Source	<1 %
32	<a href="http://es.scribd.com">es.scribd.com</a>	

---

	Internet Source	<1 %
33	<a href="http://repositori.usu.ac.id">repositori.usu.ac.id</a> Internet Source	<1 %
34	<a href="http://repository.amikom.ac.id">repository.amikom.ac.id</a> Internet Source	<1 %
35	<a href="http://widuri.raharja.info">widuri.raharja.info</a> Internet Source	<1 %
36	<a href="http://jurusan.tik.pnj.ac.id">jurusan.tik.pnj.ac.id</a> Internet Source	<1 %
37	<a href="http://theinsidemag.com">theinsidemag.com</a> Internet Source	<1 %
38	Submitted to City University of New York System Student Paper	<1 %
39	<a href="http://repozitorij.foi.unizg.hr">repozitorij.foi.unizg.hr</a> Internet Source	<1 %
40	<a href="http://stackoverflow.com">stackoverflow.com</a> Internet Source	<1 %
41	<a href="http://symbianplanet.net">symbianplanet.net</a> Internet Source	<1 %
42	Admin Teika, Boston Pahala Siahaan, Yusran Timur Samuel. "Perancangan Sistem Informasi Absensi Dan Nilai Siswa Beserta Absensi Dan Laporan Kegiatan Guru Di	<1 %

## Sekolah Advent Pancaran Kasih Berbasis Web", TeIKa, 2017

Publication

---

43	<a href="http://core.ac.uk">core.ac.uk</a> Internet Source	<1 %
44	<a href="http://repo.darmajaya.ac.id">repo.darmajaya.ac.id</a> Internet Source	<1 %
45	<a href="http://journal.stmikjayakarta.ac.id">journal.stmikjayakarta.ac.id</a> Internet Source	<1 %
46	<a href="http://openlibrarypublications.telkomuniversity.ac.id">openlibrarypublications.telkomuniversity.ac.id</a> Internet Source	<1 %
47	<a href="http://repository.pnb.ac.id">repository.pnb.ac.id</a> Internet Source	<1 %
48	<a href="http://repository.umy.ac.id">repository.umy.ac.id</a> Internet Source	<1 %
49	<a href="http://www.egitimyayinevi.com">www.egitimyayinevi.com</a> Internet Source	<1 %
50	<a href="http://eprints.binadarma.ac.id">eprints.binadarma.ac.id</a> Internet Source	<1 %
51	<a href="http://eprints.radenfatah.ac.id">eprints.radenfatah.ac.id</a> Internet Source	<1 %
52	<a href="http://eprints.upnjatim.ac.id">eprints.upnjatim.ac.id</a> Internet Source	<1 %
53	<a href="http://nasional.sindonews.com">nasional.sindonews.com</a> Internet Source	<1 %

---

54	<a href="http://publication.petra.ac.id">publication.petra.ac.id</a> Internet Source	<1 %
55	<a href="http://santrencity.blogspot.com">santrencity.blogspot.com</a> Internet Source	<1 %
56	<a href="http://tenipuri-hunter.blogspot.com">tenipuri-hunter.blogspot.com</a> Internet Source	<1 %
57	Submitted to Universiti Teknologi MARA Student Paper	<1 %
58	<a href="http://zombiedoc.com">zombiedoc.com</a> Internet Source	<1 %

Exclude quotes  Off      Exclude matches  Off  
 Exclude bibliography  On

