

DETEKSI KORBAN DALAM ROBOT POINTER PENCARIAN KORBAN GUNUNG MELETUS MENGUNAKAN ALGORITMA LOCAL BINARY PATTERN *HISTOGRAM*

DETECTION OF VICTIMS IN THE POINTER ROBOT SEARCHING FOR VICTIMS OF THE VOLCANO ERUPTION USING THE ALGORITMA LOCAL BINARY PATTERN *HISTOGRAM*

Mustain¹, Sony Sumaryo², Erwin Susanto³

¹Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

²Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

³Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

mustainal@telkomuniversity.ac.id, sonysumaryo@telkomuniversity.ac.id,
erwinelektro@telkomuniversity.ac.id

Abstrak

Penelitian ini mengembangkan sistem pendeteksi korban bencana gunung meletus yang efisien dan real-time. Sistem dirancang menggunakan Raspberry Pi, webcam, serta *algoritma Local Binary Pattern Histogram (LBPH) multi-sample*. Dengan pendekatan ini, citra korban yang ditangkap kamera diolah langsung oleh Raspberry Pi, kemudian hasil deteksi dikirimkan ke *Firebase Realtime Database* atau disimpan secara lokal saat sistem *offline*. Pengujian dilakukan dengan 558 gambar dan berhasil mendeteksi 405 di antaranya (72,6%) dengan tingkat kemiripan wajah mencapai 84%. Sistem ini mendukung operasi SAR (Search and Rescue) yang lebih cepat dan aman di medan berbahaya.

Kata kunci : deteksi korban, bencana letusan gunung berapi, raspberrypi, webcam, local binary pattern *histogram*, firebase

Abstract

This study develops a victim-searching robot equipped with a detection system aimed at enhancing the effectiveness of Search and Rescue (SAR) operations in locating survivors. Rescue efforts following volcanic eruptions present significant challenges for SAR teams, primarily due to the extreme heat and hazardous conditions caused by volcanic ash. The system utilizes a *Raspberry Pi* and webcam to process visual data, enabling real-time observation of victims in otherwise inaccessible areas. The webcam captures visual information about both the victims' condition and the surrounding environment within the search zone. The robot is tested in a simulated post-eruption environment to evaluate its capability in executing search tasks and identifying victim presence. The results are intended to demonstrate the robot's effectiveness in such extreme conditions. This technology is expected to facilitate safer rescue missions by minimizing risk to human personnel while increasing the likelihood of successful victim recovery.

Keywords: victim detection, volcanic eruption disaster, raspberrypi, webcam, local binary pattern *histogram*, firebase

1. Pendahuluan

Letusan gunung berapi merupakan salah satu bencana alam yang paling merusak dan berbahaya di Indonesia, terutama karena lokasi geografisnya yang berada di Cincin Api Pasifik. Dalam situasi darurat semacam ini, proses pencarian dan evakuasi korban sering kali terhambat oleh medan ekstrem, suhu tinggi, serta visibilitas rendah akibat abu vulkanik. Kondisi tersebut menimbulkan risiko besar bagi tim penyelamat, serta menyebabkan keterlambatan dalam menemukan korban yang masih bertahan hidup. Dengan berkembangnya teknologi embedded system dan computer vision,

solusi berbasis sistem deteksi otomatis menjadi semakin memungkinkan untuk diterapkan dalam situasi darurat. Pemanfaatan mini komputer seperti *Raspberry Pi* yang terhubung dengan kamera dan algoritma pengenalan wajah menjadi pendekatan yang menjanjikan, khususnya di area dengan sumber daya terbatas.

Penelitian ini berfokus pada pengembangan sistem deteksi korban berbasis visual yang ringan, portabel, dan dapat bekerja secara *realtime*. Sistem ini menggabungkan perangkat *Raspberry Pi*, kamera

webcam, serta algoritma *Local Binary Pattern Histogram* (LBPH) untuk mengenali pola wajah dari citra yang tertangkap. Selain itu, sistem dirancang agar mampu mengirim hasil deteksi ke *Firestore Realtime Database* sebagai *backend cloud*, serta tetap dapat menyimpan data secara lokal saat koneksi internet tidak tersedia.

Proses pengujian dilakukan dengan memanfaatkan *dataset* berjumlah ratusan citra manusia, dan lingkungan simulasi yang menggambarkan kondisi pasca letusan gunung berapi. Keberhasilan sistem ini diukur dari akurasi deteksi serta keandalannya dalam kondisi terbatas. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi korban dengan tingkat akurasi yang memadai dan potensi untuk dikembangkan lebih lanjut sebagai pendukung operasi pencarian dan penyelamatan (SAR).

2. Dasar Teori dan Perancangan Sistem

2.1 Local Binary Pattern Histogram (LBPH)

Local Binary Pattern Histogram (LBPH) merupakan kombinasi dari metode *Local Binary Pattern* (LBP) dan analisis *histogram*. LBP bekerja dengan cara membandingkan setiap *pixel* pusat dengan delapan *pixel* di sekitarnya dalam matriks 3×3 . Jika nilai *pixel* tetangga lebih besar atau sama dengan nilai *pixel* pusat, maka diberi nilai 1; jika lebih kecil, maka diberi nilai 0. Nilai-nilai ini kemudian disusun searah jarum jam menjadi bilangan biner 8-bit, lalu dikonversi ke bilangan desimal sebagai representasi pola tekstur [9]. Berikut adalah Rumus Perhitungan LBPH:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) \cdot 2^p \quad (1)$$

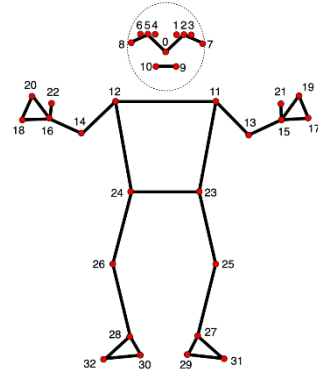
Dengan:

- i_c = nilai intensitas *pixel* pusat
- i_p = nilai intensitas *pixel* tetangga ke- p

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2)$$

Setelah nilai LBP diperoleh untuk setiap *pixel* dalam sebuah wilayah dan setelahnya dilakukan pembentukan *histogram* dari nilai-nilai LBP tersebut. Citra dibagi menjadi beberapa blok (grid) dan *histogram* dari tiap blok disatukan untuk membentuk vektor fitur akhir. Pendekatan *multi-sample LBPH* digunakan dalam penelitian ini dengan menyimpan beberapa representasi *histogram* dari wajah yang sama. Ini meningkatkan ketahanan sistem terhadap variasi pencahayaan, ekspresi wajah, dan sudut pengambilan gambar [9].

2.2 Mediapipe Pose



Gambar 1. Titik Kunci Tubuh

MediaPipe Pose adalah model deep learning yang mampu mendeteksi dan melacak 33 titik kunci (*landmark*) pada tubuh manusia dari citra RGB. Titik-titik ini mencakup bagian tubuh utama seperti kepala, bahu, siku, lutut, dan pergelangan tangan, sehingga membentuk kerangka tubuh virtual. Berbeda dengan metode berbasis deteksi wajah, MediaPipe mengenali manusia berdasarkan struktur postur tubuhnya. Keunggulan MediaPipe mencakup dukungan lintas platform (*Android, iOS, Web, Desktop*), performa *realtime*, kemudahan integrasi modular, serta ketersediaan model pra-terlatih yang mempercepat pengembangan sistem [16].

2.3 Perangkat Keras Sistem

Komponen utama sistem meliputi:



Gambar 2. Raspberry Pi 4

Spesifikasi Raspberry Pi 4: Prosesor quad-core 1.5GHz, RAM 8GB, kompatibel dengan Python dan OpenCV [7].



Gambar 3. Web Camera

Spesifikasi Web Camera Jovitek: Resolusi 1080p, frame rate 30fps, sensor CMOS [8].



Gambar 4. Powerbank

Spesifikasi Powerbank: Kapasitas 50.000 mAh, mendukung 5V/3A untuk operasi lapangan [18].

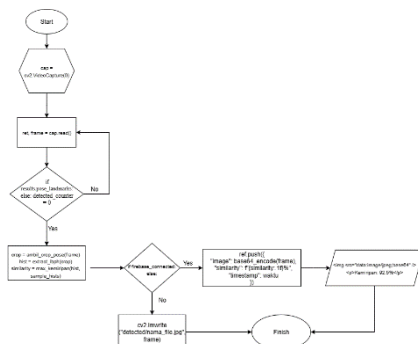
2.4 Firebase Realtime Database



Gambar 5. Firebase Realtime Database

Firebase digunakan untuk menyimpan dan menyajikan hasil deteksi secara real-time. Data berupa gambar (Base64), waktu deteksi, dan nilai kemiripan dikirim dalam format JSON menggunakan protokol HTTPS melalui *Firestore Admin SDK*. Website klien dapat menampilkan hasil deteksi langsung menggunakan *WebSocket* [12].

2.5 Alur Sistem dan Perancangan



Gambar 6. Flowchart

Flowchart sistem menunjukkan alur proses mulai dari inisialisasi kamera, pendeteksian objek sekitar, hingga pengiriman data ke Firebase. Kamera mendeteksi objek manusia menggunakan algoritma LBPH. Jika objek dikenali sebagai manusia, gambar diproses dan dicek ketersediaan jaringan. Jika jaringan tersedia, data dikirim ke *Firebase Realtime Database* menggunakan *Admin SDK*. Jika tidak, gambar disimpan ke dalam folder lokal. Data yang berhasil dikirim akan ditampilkan secara otomatis pada antarmuka web berbasis Firebase.

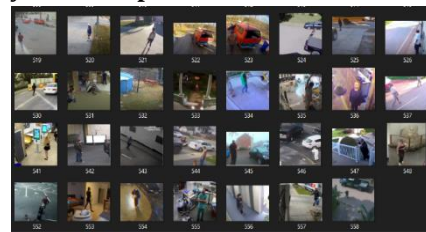
3. Pembahasan

Sistem deteksi korban yang dikembangkan dalam penelitian ini memanfaatkan *Raspberry Pi 4* sebagai pusat kendali dan pemrosesan utama. *Raspberry Pi* dipilih karena ukurannya yang ringkas, konsumsi daya yang rendah, serta kemampuan pemrosesan yang cukup untuk menjalankan algoritma deteksi berbasis visi komputer secara real-time. Kamera Jovitek beresolusi tinggi digunakan sebagai perangkat input untuk menangkap citra lingkungan sekitar secara langsung.

Setelah citra ditangkap, proses awal yang dilakukan adalah konversi gambar berwarna menjadi format *grayscale*. Langkah ini dilakukan untuk menyederhanakan proses perhitungan serta mengurangi beban komputasi, tanpa mengurangi informasi penting yang dibutuhkan untuk pengenalan pola wajah. Selanjutnya, sistem menerapkan algoritma *Local Binary Pattern Histogram* (LBPH) untuk mengekstraksi ciri-ciri tekstur dari gambar wajah. LBPH bekerja dengan membandingkan nilai *pixel* pusat dengan *pixel-pixel* di sekitarnya, menghasilkan representasi numerik dalam bentuk *histogram* yang menggambarkan pola lokal pada wajah [9].

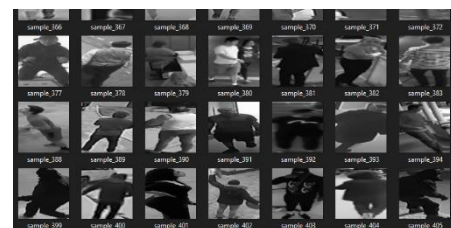
Hasil ekstraksi fitur tersebut kemudian dibandingkan dengan data latih (*Dataset training*) yang telah tersimpan sebelumnya. Pencocokan dilakukan menggunakan metode *cosine similarity* untuk mengukur kedekatan antara *histogram* citra uji dengan *histogram* dari data latih. Semakin tinggi nilai *similarity* yang dihasilkan, semakin besar kemungkinan bahwa wajah yang terdeteksi sesuai dengan salah satu sampel yang telah dilatih dalam sistem [15].

3.1 Grayscale Sampel



Gambar 7. Sampel sebelum Proses ekstraksi

Dataset diambil dari sumber eksternal (*Kaggle*) yang memiliki keragaman resolusi, latar belakang, dan intensitas pencahayaan.



Gambar 8. Sampel Setelah Proses LBPH

Grayscale pada sistem dilakukan menggunakan total 558 sampel dengan berbagai

variasi pencahayaan, sudut pengambilan gambar, dan posisi. Dari keseluruhan data sistem berhasil sebanyak 405 citra dengan tingkat akurasi keberhasilan sebesar 72,6%. Tingkat keberhasilan ini menunjukkan bahwa tidak keseluruhan sampel berhasil diekstrak ke dalam *grayscale* bisa disebabkan tidak terbacanya sampel yang ada atau sampel tidak terdeteksi pada bagian tubuh manusia.

3.2 Data Hasil Deteksi

Setelah proses deteksi berhasil dilakukan sistem akan menentukan metode penyimpanan hasil berdasarkan kondisi jaringan. Jika *Raspberry Pi* terhubung ke internet data hasil deteksi yang terdiri dari citra (format Base64), nilai *similarity*, dan waktu deteksi (*timestamp*) akan dikemas dalam format JSON dan dikirim ke *Firestore Realtime Database* menggunakan *Firestore Admin SDK*. *Firestore* dipilih karena kemampuannya dalam menyimpan dan menyajikan data secara *realtime* ke antarmuka *web* yang terhubung.

Tabel 1. Hasil Deteksi

No	Gambar Hasil Deteksi	Penjelasan
1		Sistem berhasil mendeteksi objek manusia dengan tingkat kemiripan cukup tinggi meskipun berada dalam jarak jauh dan latar vegetatif.
2		Jarak jauh dan postur samar menurunkan kemiripan namun deteksi tetap berhasil.
3		Objek manusia berhasil terdeteksi dengan tingkat kemiripan yang cukup. Posisi tubuh tegak lurus membantu algoritma mengenali pola tekstur dengan baik.
4		Deteksi dilakukan dalam jarak cukup jauh, dan kemiripan tetap berada dalam ambang aman, menunjukkan sistem cukup stabil terhadap variasi latar belakang.

Hasil pengujian pada Tabel 1. menunjukkan bahwa sistem mampu melakukan

deteksi menggunakan Algoritma *Local Binary Pattern Histogram* (LBPH) dengan kombinasi *Mediapipe* dan menghasilkan output dengan *similarity*.

Tabel 2. Data Hasil Deteksi Format Base64

ID Node	Similarity	Timestamp
-OTwYYg1ZcM6SOIHb2be	77.5%	2025-06-29 23:38:47
-OTWjYxjP1fpM5j9pP6Y	74.9%	2025-06-30 18:39:35
-OTytktz87B7bYW1iPY	62.2%	2025-06-30 10:35:20
-OTytp5AFfrezTWVCdSR	61.0%	2025-06-30 10:35:20
-OTyt2AtAforTTHvKujtT	59.9%	2025-06-30 10:35:20
-OTyhbmFLaggLQ9L9l0W	58.9%	2025-06-30 10:30:28
-OTyui4ibdLA_uQyQlXW	65.3%	2025-06-30 10:30:28
-OUyU8BydMnMWmYYN9x_n	60.4%	2025-06-30 10:30:28
-OUyua4MvpPKIBscT_K5	64.7%	2025-06-30 10:36:51

Hasil pengujian pada Tabel 2. menunjukkan sistem mengirimkan data hasil deteksi ke *Firestore Realtime Database* secara *realtime*. Data yang dikirim mencakup informasi gambar, persentase *Similarity*, dan waktu kejadian (*timestamp*) yang semuanya tersimpan dengan format Base 64 karena sistem pada *Firestore* tidak dapat menyimpan data dalam bentuk biner secara langsung sehingga harus diubah dalam bentuk teks string. Proses pengiriman ini dilakukan segera setelah sistem menyelesaikan proses deteksi. Keberhasilan pengiriman data ini menandakan bahwa sistem mampu melakukan integrasi antara modul deteksi wajah, sampel dari *Dataset*, dan layanan *firebase*.

Cosine Similarity adalah metode untuk mengukur tingkat kemiripan antara dua vektor dalam ruang berdimensi. Rumus ini banyak digunakan dalam pengenalan wajah, pencarian informasi, dan sistem rekomendasi karena mampu menilai kesamaan arah dari dua *vector* bukan besarannya. Dalam konteks pengenalan wajah berbasis *Local Binary Pattern Histogram* (LBPH), rumus ini digunakan untuk membandingkan vektor *histogram*

citra deteksi dengan vektor *histogram* referensi. Hasil *similarity* yang didapat menggunakan rumus:

$$\text{Cosine Similarity} = \frac{A \cdot B}{|A| \times |B|}$$

$$= \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Keterangan:

- A = vektor *histogram* citra input (deteksi)
- B = vektor *histogram* citra referensi (sampel)
- A·B = dot product dari kedua vektor
- ||A|| = panjang (norma) vektor A
- ||B|| = panjang (norma) vektor B
- Nilai *similarity* dikalikan 100 untuk mendapatkan presentase (%)

Untuk contoh penggunaan sebagai berikut Misalkan:

- *Histogram* vektor input: A = [0.2,0.3,0.5]
- *Histogram* vektor B = [0.1,0.4,0.5]
- A·B = (0.2) (0.1) + (0.3) (0.4) + (0.5) (0.5)
= 0.02+0.12+0.25=0.39

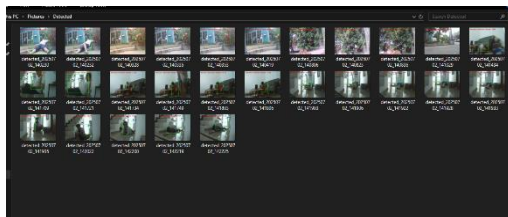
$$|A| = \sqrt{0.2^2 + 0.3^2 + 0.5^2} = \sqrt{0.04 + 0.09 + 0.25} = \sqrt{0.38} \approx 0.616$$

$$|B| = \sqrt{0.1^2 + 0.4^2 + 0.5^2} = \sqrt{0.01 + 0.16 + 0.25} = \sqrt{0.42} \approx 0.648$$

$$\text{Cosine Similarity} = \frac{0.39}{0.616 \times 0.648} = \frac{0.39}{0.399} \approx 0.977$$

$$\text{Persentase Kemiripan} = 0.977 \times 100\% = 97.7\%$$

Berdasarkan perhitungan menggunakan metode *Cosine Similarity* yang tertera di rumus (3) diperoleh nilai kemiripan sebesar 97.7% antara *histogram* citra deteksi dan data sampel referensi. Nilai ini menunjukkan bahwa citra yang terdeteksi memiliki tingkat kesamaan yang sangat tinggi dengan data manusia yang telah disimpan sebelumnya.



Gambar 10. Hasil Data Deteksi di folder lokal Ketika Sistem *Offline*

Pada Gambar 10. Folder tersebut berfungsi sebagai penyimpanan cadangan (*offline storage*) apabila sistem tidak terhubung ke internet saat proses deteksi dijalankan. Setiap gambar hasil deteksi diberi nama berdasarkan format waktu (*timestamp*), yang mencakup tanggal dan jam deteksi. Misalnya, file dengan nama *detected_20250503_215038* menunjukkan bahwa gambar tersebut diambil pada tanggal 3 Mei 2025 pukul 21:50:38.

3.3 Analisis Hasil Sistem

Hasil pengujian menunjukkan bahwa sistem deteksi korban berbasis *Raspberry Pi* dan Algoritma LBPH bekerja stabil. (3) Komunikasi data antara *Raspberry Pi* dan *Arduino* melalui *serial port* berjalan lancar tanpa kehilangan data. Nilai kemiripan wajah dan estimasi jarak berhasil dikirim secara *realtime*, dengan waktu kirim yang konsisten sekitar satu detik. Proses *parsing* di *Arduino* juga akurat dengan memisahkan informasi *similarity* dan jarak serta menampilkannya dalam format yang mudah dibaca.

Dalam hal pendeteksian manusia penggunaan Algoritma LBPH dengan sampel manual memberikan hasil kemiripan yang tinggi sementara *Dataset* dari sumber umum seperti *Kaggle* menghasilkan nilai yang lebih bervariasi. Ini menunjukkan bahwa kualitas dan kesesuaian data sampel sangat mempengaruhi akurasi deteksi. Perhitungan menggunakan *cosine similarity* menunjukkan tingkat kemiripan hingga 97%, yang berarti sistem mampu mendeteksi dengan baik jika kondisi gambar mendukung.

Setiap hasil deteksi dikirim ke *database* secara otomatis dan dapat dilihat melalui antarmuka web yang sederhana namun fungsional. Data yang dikirim mencakup *timestamp*, *similarity*, dan gambar dalam format Base64. Sistem juga menyimpan salinan gambar secara lokal jika tidak ada koneksi internet untuk sebagai sistem cadangan. Secara keseluruhan sistem ini sudah cukup solid untuk mendeteksi keberadaan manusia dan bisa menjadi solusi untuk mendukung proses pencarian korban di lapangan.

4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem deteksi korban bencana gunung meletus berbasis *Raspberry Pi* yang bekerja secara real-time menggunakan algoritma *Local Binary Pattern Histogram* (LBPH). Sistem mampu mendeteksi objek manusia melalui citra *grayscale* dengan tingkat akurasi mencapai 72,6% dari total 558 gambar uji. Data hasil deteksi dikemas dalam format JSON dan dikirim ke *Firestore Realtime Database*, serta ditampilkan secara langsung melalui antarmuka web. Sistem juga dilengkapi dengan mekanisme fallback berupa penyimpanan lokal saat jaringan tidak tersedia menjadikannya adaptif terhadap kondisi lapangan.

Keunggulan sistem terletak pada efisiensinya dalam pemrosesan data, kemampuan operasi *offline*, dan kemudahan integrasi dengan layanan cloud. Selain itu hasil pengujian membuktikan bahwa sistem cukup stabil digunakan dengan sumber daya portabel dan mampu melakukan sinkronisasi data tanpa kehilangan informasi deteksi.

Daftar Pustaka:

- [1] H. K. N. Sumartiningtyas, "Apa Itu Cincin Api Pasifik yang Membuat Wilayah Indonesia Sering Gempa? Halaman all - Kompas.com," *KOMPAS.com*, Nov. 23, 2022.
- [2] "Report on Semeru (Indonesia)," *Bulletin of the Global Volcanism Network*, vol. 47, no. 1, Jan. 2022, doi: 10.5479/si.gvp.bgvn202201-263300.
- [3] S. Sausan *et al.*, "Robot Pointer sebagai Penunjuk Jalan Tim SAR untuk Mempermudah Pencarian Korban Bencana Gempa," *Jurnal Rekayasa ElektriKA*, vol. 13, no. 2, p. 112, Aug. 2017.
- [4] S. E. Umbaugh, *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press, 2010.
- [5] G. Boesch, "Object Detection: The Definitive 2025 Guide," *viso.ai*, Oct. 18, 2024.
- [6] N. Khairunisa, Carudin, and A. Jamaludin, "ANALISIS PERBANDINGAN ALGORITMA CNN DAN YOLO DALAM MENGIDENTIFIKASI KERUSAKAN JALAN," *Jurnal Informatika Dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024
- [7] Admin, "Raspberry Pi (Definisi, Fungsi, Jenis, Spesifikasi dan Pemrograman)," *KajianPustaka.com*, Nov 03, 2024. <https://www.kajianpustaka.com/2020/12/Raspberry-Pi.html>
- [8] CMLabs, "Apa Itu Webcam: Pengertian, Fungsi, Cara Kerja, dan Harga," *ASANI*, May 21, 2024. <https://asani.co.id/blog/apa-itu-webcam/>
- [9] Cmcbinus, "Proses Ekstraksi ciri citra dengan algoritma Local Binary Pattern dalam mendukung Klasifikasi citra bertekstur," *BINUS UNIVERSITY BANDUNG - Kampus Teknologi Kreatif*, Dec. 01, 2020.
- [10] R. Armandhani, R. C. Wihandika, and Muh. A. Rahman, "Klasifikasi Gender berbasis Wajah menggunakan Metode Local Binary Pattern dan Random KNN," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 8, pp. 7575-7582, 2019.
- [11] G. Kaonang, "Raspberry Pi 4 Janjikan Performa Sekelas Komputer Desktop Tanpa Korban Fleksibilitasnya - Hybrid.co.id," *Hybrid.co.id*, Jun. 25, 2019. <https://hybrid.co.id/post/raspberry-pi-4/>
- [12] B. Interactive, "Firebase : Pengertian, Jenis, Fungsi, dan Cara Kerjanya," *Badr Interactive*, Nov. 18, 2024. <https://badr.co.id/software-development/firebase-adalah/>
- [13] "Human Detection Dataset ," *Kaggle*, Feb. 12, 2022. <https://www.Kaggle.com/Dataset/constantinwerner/human-detection-Dataset>
- [14] A. N. Putra, H. F. Abrar, and M. A. Latif, "Pencarian Korban Bencana Gempa Bumi," *Buku Tugas Akhir Capstone Design, Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom, Bandung*, 2023.
- [15] P. Kumar, V. Narayan, D. Chauhan, and A. Singh, "Multiple Human Face Detection and Recognition based on LBPH and Machine Learning," *International Journal of Innovative Research in Applied Sciences and Engineering*, vol. 4, no. 7, pp. 817–822, Jan. 2021, doi: 10.29027/ijirase.v4.i7.2021.817-822.
- [16] Codetrade, "Implementation of Human Pose Estimation using MediaPipe," *Medium*, May 20, 2022. <https://medium.com/@codetrade/implementation-of-human-pose-estimation-using-mediapipe-23a57968356b>
- [17] Raspberry Pi Stack Exchange, "Will any external battery power a Raspberry Pi?", *Raspberry Pi Stack Exchange*, 2015. [Online]. Available: <https://raspberrypi.stackexchange.com/question>

[s/26705/will-any-external-battery-power-a-raspberry-pi](#). [Accessed: 1-Jul-2025].

[18]Shopee Indonesia, "Power Bank 50000mAh Fast Charging", *Shopee.co.id*, 2025, <https://shopee.co.id/product/379847717/24061323559>.

[19]R. Kosasih and C. Daomara, "Pengenalan Wajah dengan Menggunakan Metode Local Binary Patterns *Histograms* (LBPH)," *Jurnal Media Informatika Budidarma*, vol. 5, no. 4, pp. 58–64, Oct. 2021. [Online]. Available: <https://www.researchgate.net/publication/360942543>

```
Konversi ke grayscale
Resize ke 128x128
Ekstrak histogram LBPH

Hitung cosine similarity dengan setiap sample
Ambil nilai kemiripan tertinggi

Jika mirip:
    SET detection_value = 1
    SET label = "Manusia"
    SET similarity_text = nilai kemiripan
    Tambahkan detected_counter
ELSE:
    RESET detected_counter = 0
ELSE:
    RESET detected_counter = 0

IF terdeteksi dan detected_counter == 5 dan status sebelumnya bukan deteksi:
    Ambil waktu sekarang

IF firebase_connected:
    Konversi frame ke base64
    Kirim timestamp, image, similarity ke Firebase
    Update flag "detection = 1"
ELSE:
    Simpan frame ke folder lokal "detected/"

SET last_detection_state = 1
RESET detected_counter = 0

IF tidak terdeteksi dan status sebelumnya adalah deteksi:
    IF firebase_connected:
        Update flag "detection = 0" ke Firebase
    SET last_detection_state = 0

Tampilkan frame dengan OpenCV
IF tombol ESC ditekan:
    BREAK

TUTUP kamera dan semua jendela
```

Lampiran

```
IMPORT semua library yang dibutuhkan

INISIALISASI koneksi Firebase
TRY:
    Baca kredensial
    Hubungkan ke Realtime Database
    Tandai firebase_connected = True
EXCEPT:
    Cetak pesan gagal
    firebase_connected = False

DEFINISIKAN fungsi extract_lbph(image):
    Buat LBPHFaceRecognizer
    Latih dengan gambar grayscale
    Keluarkan histogram
    RETURN histogram

DEFINISIKAN fungsi image_to_base64(image):
    Konversi gambar ke .jpg -> bytes -> base64
    RETURN string base64

MUAT semua sample gambar dari folder "samples"
FOR setiap file gambar:
    Baca -> grayscale -> resize ke 128x128
    Ekstrak histogram LBPH
    Simpan ke dalam sample_hists[]

INISIALISASI MediaPipe Pose detection
BUAT folder lokal "detected/" jika belum ada
BUKA kamera (VideoCapture)

SET last_detection_state = -1
SET detected_counter = 0
SET required_detection_frames = 5

TAMPILKAN "Menjalankan deteksi real-time..."

WHILE kamera terbuka:
    Baca frame dari kamera
    Konversi ke RGB
    Proses dengan MediaPipe Pose

    SET detection_value = 0
    SET label = ""
    SET similarity_text = "-"

    IF pose manusia terdeteksi:
        Hitung koordinat bounding box dari landmark
        Tambahkan margin
        Crop gambar berdasarkan bounding box

        IF crop tidak kosong:
```