BINER
Jurnal Ilmiah Informatika dan Komputer
Vol. 4 | No. 1 | Januari 2025

p-ISSN : 2828-0210 | e-ISSN : 2828-0229
https://ojs.unsiq.ac.id/index.php/biner

IMPLEMENTASI GOOGLE ML KIT UNTUK LIVENESS DETECTION DALAM SISTEM FACE RECOGNITION: ANALISIS KINERJA DAN KEAMANAN PADA APLIKASI MOBILE

Budi Hartanto [1)], Bramasto Wiryawan Yudanto [2)], Kustanto [3)]
[1) 2) 3)] STMIK Sinar Nusantara, [1) 1,3] Informatika, [2)] Sistem Informasi
Email : budihartanto@sinus.ac.id [1)] , bramasto@sinus.ac.id [2)] , kustanto@sinus.ac.id [3)]

Diterima : 2 Januari 2025 ; Disetujui : 30 Januari 2025 ; Dipublikasikan : 31 Januari 2025

ABSTRAK

Penelitian ini mengkaji implementasi Google ML Kit untuk deteksi liveness dalam sistem pengenalan wajah pada aplikasi mobile dengan tujuan untuk meningkatkan keamanan aplikasi dan mencegah ancaman spoofing. Pengenalan wajah telah banyak diterapkan dalam berbagai aplikasi, namun kerentanannya terhadap serangan menggunakan foto atau video statis menimbulkan tantangan besar dalam hal keamanan. Oleh karena itu, deteksi liveness yang mengidentifikasi apakah wajah yang dikenali merupakan wajah hidup menjadi sangat penting. Penelitian ini fokus pada integrasi teknologi liveness detection dalam pengenalan wajah menggunakan Google ML Kit, yang memungkinkan aplikasi mobile untuk mendeteksi gerakan kepala (geleng kanan, kiri) dan kedipan mata sebagai indikator wajah yang hidup. Hasil implementasi menunjukkan bahwa sistem dapat mencapai akurasi tinggi dengan kecepatan pengenalan yang baik, serta mampu mengurangi potensi serangan spoofing. Penelitian ini memberikan kontribusi pada pengembangan aplikasi mobile yang lebih aman dan dapat diandalkan dalam pengenalan wajah, dengan memastikan bahwa sistem tidak hanya mengenali wajah, tetapi juga memastikan bahwa wajah tersebut adalah wajah yang hidup.
Kata Kunci : *Google ML Kit, Liveness Detection, Face Recognition, Mobile Application*

ABSTRACT

This research examines the implementation of Google ML Kit for liveness detection in face recognition systems in mobile applications with the aim of improving application security and preventing spoofing threats. Face recognition has been widely applied in various applications, but its vulnerability to attacks using static photos or videos poses a great challenge in terms of security. Therefore, liveness detection that identifies whether a recognized face is a living face is of great importance. This research focuses on the integration of liveness detection technology in face recognition using Google ML Kit, which enables mobile applications to detect head movements (right, left shake) and eye blinks as indicators of a living face. The implementation results show that the system can achieve high accuracy with good recognition speed, and can reduce the potential for spoofing attacks. This research contributes to the development of more secure and reliable mobile applications in face recognition, by ensuring that the system not only recognizes the face, but also ensures that the face is a living face.
Keywords ***:*** *Google ML Kit, Liveness Detection, Face Recognition, Mobile Application*

1. PENDAHULUAN

Latar belakang masalah dalam penelitian ini berfokus pada implementasi Google ML Kit untuk deteksi liveness dalam sistem pengenalan wajah pada aplikasi mobile. Pengenalan wajah telah menjadi salah satu teknologi biometrik yang paling banyak digunakan dalam berbagai aplikasi, mulai dari keamanan hingga interaksi pengguna. Namun, tantangan utama dalam penerapan teknologi ini adalah risiko spoofing, di mana penyerang dapat menggunakan foto atau video untuk mengelabui sistem pengenalan wajah. Oleh karena itu, deteksi liveness menjadi krusial untuk memastikan bahwa yang diidentifikasi adalah individu yang nyata dan bukan representasi statis [1], [2].

Dalam konteks keamanan, teknologi pengenalan wajah telah menunjukkan potensi yang signifikan. Sebuah studi menunjukkan bahwa metode pengenalan wajah yang baru dapat mencapai tingkat akurasi yang tinggi, yaitu 97,10%, dengan kecepatan pengenalan yang cepat [1]. Hal ini menunjukkan bahwa dengan pengembangan yang tepat, sistem pengenalan wajah dapat diandalkan untuk aplikasi keamanan, termasuk dalam pengawasan video dan kontrol akses [2]. Namun, untuk meningkatkan keandalan sistem ini, penting untuk mengintegrasikan mekanisme deteksi liveness yang efektif. Penelitian sebelumnya menunjukkan bahwa sistem yang tidak dilengkapi dengan deteksi liveness rentan terhadap serangan spoofing, yang dapat mengakibatkan pelanggaran keamanan [3].

Google ML Kit menawarkan solusi yang menjanjikan untuk masalah ini dengan menyediakan alat yang dapat digunakan untuk mengembangkan aplikasi mobile yang mampu mendeteksi liveness secara efektif. Dengan memanfaatkan teknologi ini, pengembang dapat menciptakan aplikasi yang tidak hanya mampu mengenali wajah tetapi juga memastikan bahwa wajah tersebut adalah wajah yang hidup [4]. mplementasi deteksi liveness dalam sistem pengenalan wajah diharapkan dapat meningkatkan keamanan aplikasi mobile secara keseluruhan dan memberikan perlindungan yang lebih baik terhadap potensi ancaman [5].

Latar belakang masalah dalam penelitian ini semakin diperkuat oleh meningkatnya ketergantungan masyarakat terhadap aplikasi mobile yang memanfaatkan teknologi pengenalan wajah. Dalam konteks ini, keamanan aplikasi mobile menjadi perhatian utama, terutama terkait dengan potensi ancaman yang dihadapi oleh pengguna. Penelitian menunjukkan bahwa banyak aplikasi mobile rentan terhadap berbagai jenis serangan, termasuk serangan spoofing yang dapat mengeksploitasi kelemahan dalam sistem pengenalan wajah [6]. Ketidakamanan ini disebabkan oleh berbagai faktor, termasuk kesalahan dalam otentikasi dan otorisasi, serta penyimpanan data yang tidak aman [7]. Oleh karena itu, penting untuk mengembangkan solusi yang tidak hanya dapat mengenali wajah tetapi juga memastikan bahwa wajah tersebut adalah wajah yang hidup, sehingga dapat mencegah akses tidak sah ke sistem.

Lebih lanjut, penelitian sebelumnya menunjukkan bahwa pengguna mobile sering kali kurang menyadari praktik keamanan yang baik, yang membuat mereka menjadi target empuk bagi para penyerang [8]. Dengan meningkatnya penggunaan aplikasi mobile untuk transaksi keuangan dan pengelolaan data pribadi, risiko kebocoran informasi menjadi semakin tinggi. Dalam hal ini, teknologi biometrik, termasuk pengenalan wajah, telah muncul sebagai solusi yang menjanjikan untuk meningkatkan keamanan aplikasi mobile. Namun, tanpa adanya mekanisme deteksi liveness yang efektif, teknologi ini tetap rentan terhadap serangan [9]. Oleh karena itu, penelitian ini bertujuan untuk mengeksplorasi bagaimana integrasi Google ML Kit dalam sistem deteksi liveness dapat meningkatkan keamanan aplikasi mobile yang menggunakan pengenalan wajah. Pendekatan ini tidak hanya melibatkan pengembangan teknologi deteksi liveness yang efektif, tetapi juga mencakup pelatihan bagi pengembang aplikasi untuk memahami dan mengatasi potensi kerentanan dalam aplikasi mereka. Dengan demikian, penelitian ini tidak hanya berfokus pada aspek teknis dari deteksi liveness, tetapi juga pada pentingnya kesadaran keamanan di kalangan pengembang dan pengguna aplikasi mobile.

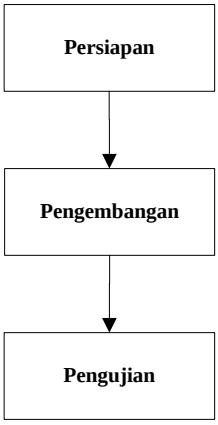
Akhirnya, penting untuk dicatat bahwa pengembangan aplikasi mobile yang aman harus mempertimbangkan siklus hidup pengembangan yang sadar akan keamanan. Dengan mengadopsi pendekatan yang komprehensif ini, diharapkan dapat tercipta

aplikasi mobile yang lebih aman dan andal dalam menghadapi ancaman yang terus berkembang [10].

Secara keseluruhan, penelitian ini bertujuan untuk menganalisis kinerja dan keamanan sistem pengenalan wajah yang diintegrasikan dengan deteksi liveness menggunakan Google ML Kit, serta mengeksplorasi bagaimana teknologi ini dapat diterapkan dalam konteks aplikasi mobile untuk meningkatkan keamanan dan keandalan sistem pengenalan wajah.

2. METODE

Dalam penelitian ini, metode yang digunakan untuk implementasi Google ML Kit dalam deteksi liveness pada sistem pengenalan wajah dibagi menjadi beberapa tahap, yaitu persiapan data, pengembangan sistem, pengujian, dan analisis kinerja. Setiap tahap memiliki prosedur yang spesifik untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi dengan baik dan memenuhi standar keamanan yang diharapkan



Gambar 1. Tahapan Penelitian

Tahapan penelitian di atas dapat dijelaskan sebagai berikut:

1. Persiapan Data

Tahap pertama dalam penelitian ini adalah pengumpulan dan persiapan data. Data yang digunakan terdiri dari gambar wajah yang diambil dalam berbagai kondisi pencahayaan dan sudut pandang untuk meningkatkan keberagaman dataset. Dataset ini mencakup gambar wajah yang diambil dari individu yang berbeda, baik dalam kondisi hidup (liveness) maupun tidak (spoofing) seperti foto dan video. Penggunaan dataset yang beragam ini penting untuk melatih model deteksi liveness agar dapat mengenali wajah dengan akurasi tinggi dalam berbagai situasi [1].

2. Pengembangan

Setelah data siap, tahap berikutnya adalah pengembangan sistem. Dalam penelitian ini, Google ML Kit digunakan sebagai alat utama untuk pengenalan wajah dan deteksi liveness. Google ML Kit menyediakan API yang memungkinkan pengembang untuk mengintegrasikan fitur pengenalan wajah ke dalam aplikasi mobile dengan mudah. Proses pengenalan wajah dilakukan dengan menggunakan algoritma deep learning, yang telah terbukti efektif dalam meningkatkan akurasi pengenalan wajah.

3. Pengujian dan Analisis

Setelah sistem dikembangkan, tahap selanjutnya adalah pengujian. Pengujian dilakukan dengan menggunakan dataset yang terpisah dari dataset pelatihan untuk mengevaluasi kinerja sistem. Kinerja sistem diukur berdasarkan beberapa metrik, termasuk tingkat akurasi, kecepatan pengenalan, dan tingkat kesalahan dalam mendeteksi spoofing. Hasil pengujian kemudian dianalisis untuk menentukan efektivitas sistem dalam mendeteksi liveness dan mengenali wajah. Analisis ini juga mencakup perbandingan dengan metode lain yang ada di literatur, seperti penggunaan algoritma Viola-Jones dan PCA untuk pengenalan wajah, untuk menilai keunggulan sistem yang dikembangkan [11], [12].

3. HASIL DAN PEMBAHASAN

Pada sub bab hasil dan pembahasan ini dilakukan pembahasan mengenai pembuatan aplikasi atau implementasi sampai dengan hasil dari pengujian model.

3.1 Perancangan Model

1. Pembuatan Database Model

lib > ML > Recognition.dart

```dart
import 'dart:ui';

class Recognition {
  String nik;
  String name;
  Rect location;
  List<double> embeddings;
  double distance;

  Recognition(
      this.nik, this.name, this.location, this.embeddings, this.distance);
}
```

Gambar 2. Pembuatan Database Model

Kode tersebut mendefinisikan kelas Recognition dalam Dart, yang merepresentasikan hasil pengenalan (misalnya pengenalan wajah atau identitas). Kelas ini memiliki properti seperti nik (NIK), name (nama individu), location (koordinat area pengenalan dalam bentuk Rect), embeddings (vektor fitur dalam bentuk daftar double), dan distance (jarak untuk mengukur kesamaan). Konstruktor kelas digunakan untuk menginisialisasi semua properti tersebut.

2. Penyimpanan Data Wajah

```
findNearest(List<double> emb) {
  Pair pair = Pair("000", "Tidak Dikenal", -5);
  for (MapEntry<String, Recognition> item in registered.entries) {
    final String nik = item.key;
    final String name = item.value.name;
    List<double> knownEmb = item.value.embeddings;
    double distance = 0;
    for (int i = 0; i < emb.length; i++) {
      double diff = emb[i] - knownEmb[i];
      distance += diff * diff;
    }
    distance = sqrt(distance);
    if (pair.distance == -5 || distance < pair.distance) {
      pair.distance = distance;
      pair.nik = nik;
      pair.name = name;
    }
  }
  return pair;
}

void close() {
  interpreter.close();
}
}
```

Gambar 3. Penyimpanan Data Wajah

Fungsi findNearest bertujuan untuk mencari data yang paling mendekati berdasarkan jarak Euclidean antara vektor embedding yang diberikan (emb) dan vektor embedding yang telah terdaftar dalam variabel registered. Fungsi ini menginisialisasi sebuah objek Pair dengan nilai default (NIK "000", nama "Tidak Dikenal", dan jarak awal -5). Kemudian, fungsi melakukan iterasi pada setiap pasangan kunci-nilai dalam registered. Pada setiap iterasi, jarak Euclidean dihitung dengan menjumlahkan kuadrat perbedaan setiap elemen antara emb dan vektor embedding terdaftar, lalu diambil akar kuadratnya. Jika jarak yang dihitung lebih kecil dari jarak pada objek Pair, maka nilai dalam Pair diperbarui dengan NIK, nama, dan jarak tersebut. Setelah iterasi selesai, fungsi mengembalikan objek Pair yang berisi data dengan jarak terdekat.

3. Pengujian Wajah Liveness

```
for (Recognition face in faces) {
  canvas.drawRect(
    Rect.fromLTRB(
      camDire2 == CameraLensDirection.front
          ? (absoluteImageSize.width - face.location.right) * scaleX
          : face.location.left * scaleX,
      face.location.top * scaleY,
      camDire2 == CameraLensDirection.front
          ? (absoluteImageSize.width - face.location.left) * scaleX
          : face.location.right * scaleX,
      face.location.bottom * scaleY,
    ),
    paint,
  );
```

Gambar 4. Pengenalan Wajah

```
import 'dart:io';
import 'dart:ui';
import 'package:flutter/cupertino.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:camera/camera.dart';
import 'package:flutter/services.dart';
import 'package:google_mlkit_face_detection/google_mlkit_face_detection.dart';
import 'package:image/image.dart' as img;
```

Gambar 5. Penggunaan Google ML Kit

Kelas FaceDetectorPainter adalah turunan dari CustomPainter yang digunakan untuk menggambar deteksi wajah pada kanvas berdasarkan lokasi wajah yang telah dikenali. Kelas ini menerima ukuran asli gambar (absoluteImageSize), daftar objek deteksi wajah (faces), dan arah kamera (camDire2). Dalam metode paint, setiap wajah digambar sebagai persegi panjang menggunakan canvas.drawRect, dengan memperhatikan skala antara ukuran asli gambar dan ukuran kanvas. Jika kamera menghadap depan, koordinat horizontal dibalik untuk mencerminkan gambar. Selain itu, nama dan jarak pengenalan wajah ditampilkan di atas setiap persegi panjang dengan menggunakan teks. Metode shouldRepaint selalu mengembalikan true, sehingga gambar diperbarui setiap kali data wajah berubah.

3.2 Hasil dan Evaluasi

Sub bab ini membahas implementasi liveness detection pada aplikasi pengenalan wajah berbasis Android, yang bertujuan untuk memastikan bahwa wajah yang dikenali adalah wajah hidup dan bukan gambar atau video palsu. Sistem ini dirancang dengan tahapan validasi berupa tiga gerakan utama, yaitu pengguna diminta untuk menggelengkan kepala ke kanan, kemudian ke kiri, dan diakhiri dengan mengedipkan mata. Setiap gerakan akan dideteksi menggunakan algoritma face tracking secara real-time yang memanfaatkan kamera

perangkat Android. Kombinasi tahapan ini memberikan tingkat keamanan lebih tinggi, karena sulit dipalsukan oleh media statis seperti foto atau video. Implementasi dilakukan dengan memanfaatkan pustaka pemrosesan wajah seperti MediaPipe atau ML Kit, yang terintegrasi langsung dengan sistem pengenalan wajah untuk memastikan keaslian pengguna.

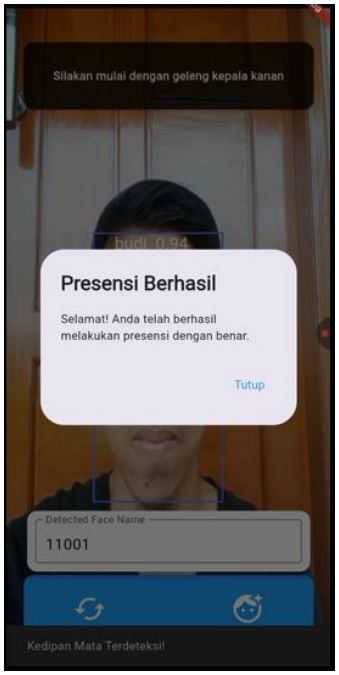
1. Deteksi Geleng Kepala Arah Kanan

Gambar 2. Intruksi Liveness

2. Deteksi Kedipan Mata

Gambar 3. Intruksi Liveness

3. Berhasil Melakukan Verifikasi



Gambar 4. Berhasil Melakukan Verifikasi

4. PENUTUP

4.1. Kesimpulan

Penelitian ini menunjukkan bahwa implementasi Google ML Kit untuk deteksi liveness dalam sistem pengenalan wajah pada aplikasi mobile memiliki potensi besar dalam meningkatkan keamanan dan keandalan sistem. Dengan memanfaatkan fitur deteksi liveness, seperti pengenalan gerakan kepala dan kedipan mata, sistem mampu membedakan wajah hidup dari media statis, sehingga risiko serangan spoofing dapat diminimalkan. Hasil analisis menunjukkan bahwa teknologi ini dapat diintegrasikan dengan mudah pada aplikasi mobile, menawarkan tingkat akurasi tinggi dan waktu respons yang cepat. Selain itu, penelitian ini menggarisbawahi pentingnya mekanisme liveness detection dalam mencegah akses tidak sah, terutama dalam era meningkatnya ketergantungan pada aplikasi berbasis pengenalan wajah. Dengan demikian, implementasi Google ML Kit tidak hanya memberikan solusi teknis yang efektif tetapi juga menjadi langkah strategis dalam mengamankan aplikasi mobile terhadap potensi ancaman keamanan di masa depan.

4.2. Saran

Pengembangan secara lebih baik dapat difokuskan pada peningkatan akurasi dan kecepatan deteksi liveness dengan mengoptimalkan algoritma yang ada pada Google ML Kit atau mengkombinasikannya dengan metode lain seperti deep learning untuk meningkatkan ketahanan terhadap serangan spoofing yang lebih canggih. Kedua, penelitian ini dapat memperluas pengujian pada berbagai kondisi pencahayaan, sudut kamera, dan latar belakang untuk memastikan bahwa sistem deteksi liveness berfungsi dengan baik di lingkungan yang beragam. Ketiga, implementasi solusi liveness detection harus melibatkan pendekatan multi-faktor, seperti kombinasi pengenalan wajah dan otentikasi biometrik lainnya (misalnya, sidik jari atau suara), untuk lebih meningkatkan lapisan keamanan aplikasi. Terakhir, disarankan untuk melakukan pengujian lebih lanjut terkait pengalaman pengguna dan dampak penggunaan liveness detection terhadap kinerja aplikasi, sehingga aplikasi mobile tetap responsif dan mudah digunakan tanpa mengorbankan aspek keamanan.

5. DAFTAR PUSTAKA

[1] Z. Wang, X. Zhang, P. Yu, W. Duan, D. Zhu, and N. Cao, "A New Face Recognition Method for Intelligent Security," *Applied Sciences*, vol. 10, no. 3, p. 852, 2020, doi: 10.3390/app10030852.

[2] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face Recognition Systems: A Survey," *Sensors*, vol. 20, no. 2, p. 342, 2020, doi: 10.3390/s20020342.

[3] E. AVUCLU and M. KOKLU, "Development of Voice and Face Recognition Based Security Software for Biometric Systems," 2023, doi: 10.58190/icontas.2023.52.

[4] O. E. Putra, R. Devita, and N. Wahyudi, "Safe Security System Using Face Recognition Based on IoT," *Sinkron*, vol. 8, no. 2, pp. 1021–1030, 2023, doi: 10.33395/sinkron.v8i2.12231.

[5] S. Quirolgico, J. Voas, T. Karygiannis, C. Michael, and K. Scarfone, "Vetting the Security of Mobile Applications," 2019, doi: 10.6028/nist.sp.800-163r1.

[6] A. Efe and Ş. Özdamarlar, "Security Controls Against Mobile Application Threats," *International Journal of Engineering and Innovative Research*, vol. 3, no. 2, pp. 145–162, 2021, doi: 10.47933/ijeir.838873.

[7] E. Zaitseva, T. Hovorushchenko, O. Pavlova, and Y. Voichur, "Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Application Insecurity," *Systems*, vol. 11, no. 5, p. 242, 2023, doi: 10.3390/systems11050242.

[8] A. Wambua, "Security-Aware Mobile Application Development Lifecycle (sMADLC)," *International Journal of Education and Management Engineering*, vol. 13, no. 2, pp. 36–42, 2023, doi: 10.5815/ijeme.2023.02.05.

[9] P. Subramani, G. Rajendran, J. Sengupta, R. P. de Prado, and B. D. Parameshachari, "A Block Bi-Diagonalization-Based Pre-Coding for Indoor Multiple-Input-Multiple-Output-Visible Light Communication System," *Energies (Basel)*, vol. 13, no. 13, p. 3466, 2020, doi: 10.3390/en13133466.

[10] S. A. Tovino, "Privacy and Security Issues With Mobile Health Research Applications," *The Journal of Law Medicine & Ethics*, vol. 48, no. S1, pp. 154–158, 2020, doi: 10.1177/1073110520917041.

[11] C. S. Keau, C. K. On, M. H. A. Hijazi, and M. Singh, "Smart-Hadir – Mobile Based Attendance Management System," *International Journal of Interactive Mobile Technologies (Ijim)*, vol. 15, no. 14, p. 4, 2021, doi: 10.3991/ijim.v15i14.22677.

[12] T. Mazakov and D. N. Narynbekovna, "Development of Biometric Methods and Information Security Tools," *News of the National Academy of Sciences of the Republic of Kazakhstan*, vol. 2, no. 336, pp. 121–124, 2021, doi: 10.32014/2021.2518-1726.30.