

Perancangan Dan Implementasi Sistem Pengenalan Wajah Menggunakan Program Python Sebagai Kontrol Akses Masuk Parkir Kendaraan Pada Area Perusahaan

(Design And Implementation Of A Face Recognition System Using Python Program As Access Control For Vehicle Parking In A Company Area)

Adi Saputra Hutavea¹, Haniel Rangga Pramuditya Putra², Rahmad Syaifudin³

^{1,2,3} Program Studi Teknik Elektro-Fakultas Teknik-Universitas Tulungagung

Alamat korespondensi:

Jl. Ki Mangun Sarkoro Beji, Kec. Boyolangu, Kab. Tulungagung 66233

Email: itslutapea@gmail.com

Abstract

This study discusses the design and implementation of a face recognition system using the Python programming language as an access control method for vehicle parking areas within a company environment. The system is designed to enhance the security and efficiency of vehicle entry processes by identifying registered users' faces. The technology utilized includes Python libraries such as OpenCV and face_recognition, which enable real-time face detection and verification via camera. The process begins with registering the user's face into a database, followed by face matching when the user enters the parking area. If the face is detected and recognized, access is granted; otherwise, it is denied. The implementation results show that the system can recognize faces with high accuracy under sufficient lighting and proper face positioning. This system is expected to be a practical solution for secure and automated parking access management in company premises.

Keywords: facial recognition; access control; automated parking; Python; OpenCV.

Abstrak

Penelitian ini membahas tentang perancangan dan implementasi sistem pengenalan wajah menggunakan bahasa pemrograman Python sebagai metode kontrol akses untuk area parkir kendaraan di lingkungan perusahaan. Sistem ini dirancang untuk meningkatkan keamanan dan efisiensi proses masuk kendaraan dengan cara mengidentifikasi wajah pengguna yang telah terdaftar. Teknologi yang digunakan meliputi library Python seperti OpenCV dan face_recognition yang memungkinkan deteksi dan verifikasi wajah secara real-time melalui kamera. Proses dimulai dengan pendaftaran wajah pengguna ke dalam database, dilanjutkan dengan proses pencocokan wajah saat pengguna memasuki area parkir. Jika wajah terdeteksi dan dikenali, sistem akan memberikan akses; sebaliknya, akses akan ditolak. Hasil implementasi menunjukkan bahwa sistem ini mampu mengenali wajah dengan akurasi tinggi dalam kondisi pencahayaan yang cukup dan posisi wajah yang tepat. Sistem ini diharapkan dapat menjadi solusi praktis dalam pengelolaan akses parkir di perusahaan secara otomatis dan aman.

Kata kunci: pengenalan wajah; kontrol akses; parkir otomatis; Python; OpenCV.

PENDAHULUAN

Kebutuhan akan sistem keamanan yang lebih pintar dan efisien meningkat seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi. Kebutuhan ini terutama terkait dengan mengelola akses kendaraan di tempat kerja. Salah satu teknik yang saat ini banyak dikembangkan untuk mengautentikasi dan mengontrol akses adalah teknologi pengenalan wajah, yang dikenal sebagai Face ID. Teknologi ini memberikan solusi non-kontak yang efektif dan nyaman bagi pengguna.

Kehilangan, pemalsuan, atau penyalahgunaan kartu identitas, kunci fisik, atau PIN adalah kelemahan sistem kontrol akses konvensional. Hal ini dapat menurunkan produktivitas operasi dan mengancam keamanan area parkir bisnis. Akibatnya, dibutuhkan sistem yang dapat secara otomatis dan akurat mengidentifikasi individu dengan tingkat keamanan yang lebih tinggi.

Dalam penelitian ini, sistem pengenalan wajah berbasis Python dibangun dan digunakan untuk mengatur akses kendaraan ke area parkir perusahaan. Pilihan Python sebagai platform pengembangan bergantung pada ketersediaan berbagai pustaka (library) pendukung, seperti OpenCV dan `face_recognition`. Melalui integrasi dengan kamera, buku-buku ini memungkinkan deteksi dan verifikasi wajah secara real-time. Ini dilakukan dengan membandingkan wajah pengguna yang terekam oleh kamera dengan data wajah yang terdaftar dalam basis data; jika wajah dianggap sesuai dengan data, maka sistem akan secara otomatis memberikan akses.

Tujuan penelitian ini adalah untuk membuat solusi berbasis teknologi pengenalan wajah yang dapat digunakan dalam sistem pengelolaan akses parkir kendaraan untuk meningkatkan efisiensi dan keamanan di tempat kerja. Selain itu, tujuan penelitian ini adalah untuk menguji keandalan dan keakuratan sistem dalam lingkungan nyata, seperti variasi pencahayaan dan sudut pandang.

Dengan sistem ini, perusahaan diharapkan dapat mengelola akses kendaraan dengan lebih aman, efektif, dan otomatis, sehingga mereka tidak perlu bergantung pada metode konvensional yang rawan terhadap masalah keamanan.

METODE PENELITIAN

Metode rekayasa perangkat lunak (software engineering) digunakan dalam penelitian ini untuk menguji kinerja sistem dalam situasi nyata. Metode ini dimaksudkan untuk membangun, membangun, dan menguji sebuah sistem pengenalan wajah berbasis Python yang dapat digunakan untuk mengontrol akses otomatis ke area parkir kendaraan perusahaan.

Penelitian terapan ini berfokus pada penggunaan teknologi pengenalan wajah dalam sistem keamanan parkir. Sistem dapat mengenali wajah pengguna secara real-time dan secara otomatis memberikan atau menolak akses.

Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem adalah metode waterfall, yang terdiri dari lima tahapan utama sebagai berikut:

- a. Analisis Kebutuhan. Pada titik ini, dikumpulkan kebutuhan sistem perangkat lunak dan perangkat keras (OpenCV, Python, dan pemrosesan wajah) dan perangkat keras (komputer, modul kontrol, kamera). Studi literatur dan observasi lapangan digunakan untuk menentukan kebutuhan sistem yang fungsional dan tidak fungsional.
- b. Perancangan Sistem. Perancangan mencakup diagram alur sistem (flowchart), perancangan antarmuka pengguna (UI), struktur penyimpanan data pengguna (database wajah), integrasi perangkat keras (kendali pintu atau barrier dengan GPIO atau relay).
- c. Implementasi. Pada titik ini, program dibuat menggunakan bahasa Python, menggunakan pustaka OpenCV untuk deteksi wajah dan `face_recognition` untuk verifikasi identitas pengguna. Sistem juga memiliki kontroler pintu otomatis dan kamera.

KERANGKA TEORITIS

Pengenalan Wajah (Face Recognition)

Secara umum, sistem pengenalan citra wajah terbagi menjadi dua kategori, yaitu *feature based* dan *sistem imaged-based*. Sistem pertama menggunakan elemen wajah seperti mata, hidung, dan mulut, yang kemudian dimodelkan secara geometris untuk menunjukkan hubungannya. Sistem kedua menggunakan sistem piksel asli gambar yang kemudian direpresentasikan dengan metode tertentu, seperti principal component analysis (PCA), yang kemudian digunakan untuk klarifikasi identitas.

Dalam bidang visi komputer, pengenalan wajah adalah metode untuk mengenali atau memverifikasi identitas seseorang dari citra wajah. Tiga tahap utama biasanya terlibat dalam metode ini antara lain deteksi wajah, penyelarasan (alignment), dan ekstraksi representasi wajah melalui fitur diskriminatif. Metode seperti Haar Cascade, HOG, atau detektor berdasarkan CNN sering digunakan untuk mendeteksi wajah. Kemudian, metode seperti Eigenfaces, Fisherfaces, Local Binary Patterns Histograms (LBPH) atau model pembelajaran mendalam seperti MTCNN, FaceNet, dan DeepFace [2]

digunakan untuk mengolah wajah. Pada dataset seperti LFW, modern deep learning dapat mencapai akurasi hingga 97%.

Algoritma dan Library Python

Banyak aplikasi, terutama yang berkaitan dengan keamanan, sekarang menggunakan pengenalan wajah. Penggunaan wajah sebagai identifikasi memiliki banyak keuntungan, terutama karena memerlukan kartu atau foto untuk diidentifikasi. Sebuah gambar yang menggambarkan sebuah gambar yang terdiri dari vektor yang berukuran cukup besar adalah masalah utamanya. Ada banyak cara untuk mengurangi ukuran gambar yang akan diproses. Salah satunya menggunakan algoritma eigenface. Beberapa algoritma yang dapat digunakan, antara lain:

1. OpenCV: pustaka utama untuk visi komputer real-time yang mendukung deteksi, deteksi wajah melalui cascade, HOG, dan integrasi dengan model deep learning.
2. face_recognition (dlib/FaceNet): memberikan pipeline siap pakai dalam Python, cosine similarity untuk verifikasi, dan ekstraksi embedding.
3. Eigenfaces, Fisherfaces, dan LBPH adalah model klasik yang mudah digunakan dengan OpenCV; Eigenfaces dan Fisherfaces cocok untuk kondisi terkontrol, sementara LBPH lebih tahan terhadap perubahan pencahayaan.

Arsitektur Sistem Kontrol Akses Parkir

Sistem kontrol akses parkir otomatis yang berbasis mikrokontroler Arduino Uno dan menggunakan sensor jarak, motor servo, dan layar LCD untuk mengatur keluar-masuk kendaraan secara otomatis. Arduino Uno berfungsi sebagai pusat kendali, atau pengontrol, yang mengatur logika kerja semua komponen sistem. Dua sensor ultrasonik HC-SR04 dipasang di sistem ini. Sensor pertama berada di dekat pintu masuk dan mendeteksi keberadaan kendaraan yang akan memasuki area parkir, sedangkan sensor kedua berada di dalam area parkir dan memantau ketersediaan tempat parkir.

Sensor ultrasonik pertama akan mengukur jarak antara kendaraan dan pintu masuk. Arduino akan memeriksa data dari sensor kedua jika kendaraan terdeteksi dalam jarak tertentu untuk mengetahui apakah masih ada ruang parkir yang tersedia. Arduino akan mengaktifkan motor servo untuk membuka palang otomatis saat kendaraan diperbolehkan masuk jika ruang parkir masih tersedia. Secara bersamaan, tampilan LCD 16x2 akan memberikan informasi kepada pengemudi, seperti tulisan "Selamat Datang" atau "Akses Ditolak". Setelah kendaraan melewati gerbang, motor servo akan secara otomatis menutup kembali palang tersebut untuk mencegah kendaraan lain masuk tanpa melakukan pengecekan ulang.

Namun, jika sensor kedua menunjukkan bahwa ruang parkir sudah penuh, Arduino tidak akan mengaktifkan servo dan palang akan tetap tertutup. Pada saat yang sama, informasi "Parking Full" akan ditampilkan di layar LCD. Ini memberi tahu pengemudi bahwa mereka tidak dapat memasuki area parkir. Sistem ini menggunakan LCD yang dilengkapi dengan modul I2C. Modul ini memungkinkan Arduino untuk berkomunikasi dengannya hanya melalui dua pin (SDA dan SCL). Ini meningkatkan efektivitas penggunaan pin digital Arduino.

Arsitektur ini menunjukkan bagaimana sistem otomatis yang bermanfaat untuk kehidupan sehari-hari, khususnya dalam pengelolaan parkir, dapat dibuat dengan integrasi sederhana dari sensor, aktuator, dan layar visual. Selain itu, sistem ini dapat dilengkapi dengan fitur tambahan seperti pembacaan kartu RFID, kamera ANPR (pengakuan nomor otomatis), dan sistem monitoring berbasis Internet of Things (IoT).

Alat dan Bahan

Arduino UNO

Merupakan papan mikrokontroler sumber terbuka yang dirancang untuk mempermudah proses pembuatan prototipe elektronik. Ditenagai oleh chip ATmega328P, papan ini menyediakan *input/output* digital dan analog, diprogram melalui Arduino IDE menggunakan varian bahasa C/C++. Desain hardware dan software-nya didistribusikan secara bebas (open-source), dengan dukungan komunitas global yang luas.

Menurut datasheet, Arduino Uno adalah board mikrokontroler berbasis ATmega328 yang memiliki 14 pin input digital, dengan 6 pin input analog yang dapat digunakan sebagai output PWM. Selain itu, memiliki jack daya, header ICSP, koneksi USB, osilator kristal 16 MHz, dan tombol reset.

Tabel 1. Spesifikasi AT-mega 328

No	Mikrokontroler	AT-mega 328
1	Tegangan pengoperasian	5V
2	Tegangan input	7-12V
3	Batas tegangan input 6-20V	Digital 14 (6 PWM)
4	Jumlah pin I/O analog	6
5	Jumlah pin input	40 Ma
6	Arus DC tiap pin I/O	50 Ma
7	Arus DC untuk pin 3.3V	32 KB (ATmega328)
8	SRAM	2KB (Atmega328)



Gambar 1. Arduino UNO

Liquid Crystal Display (LCD)

LCD adalah layar datar yang menggunakan kristal cair untuk menampilkan gambar. Kristal cair adalah bahan yang sifatnya berada di antara cair dan padat dan tidak memancarkan cahaya sendiri; oleh karena itu, biasanya LED, adalah sumber cahaya yang datang dari belakang layar untuk menampilkan gambar.

LCD terdiri dari beberapa lapisan, termasuk lapisan kristal cair di antara dua lapisan kaca atau plastik yang dilapisi elektroda transparan. Cahaya latar, atau backlight, dipolarisasi saat melewati lapisan pertama. Hal ini terjadi karena ketika tegangan listrik diterapkan pada elektroda, orientasi molekul kristal cair berubah, yang mengubah polarisasi cahaya yang melewatinya. Perubahan orientasi ini kemudian digabungkan dengan filter polarisasi kedua untuk mengontrol intensitas cahaya yang keluar, yang menghasilkan gambar yang dilihat oleh mata manusia.

Beberapa jenis LCD yang umum digunakan meliputi:

1. Twisted Nematic (TN): Jenis ini memiliki waktu respons cepat dan biaya produksi rendah, tetapi sudut pandang dan reproduksi warnanya terbatas.
2. In-Plane Switching (IPS): Menawarkan sudut pandang lebih luas dan reproduksi warna lebih baik dibandingkan TN, namun dengan waktu respons yang sedikit lebih lambat.
3. Vertical Alignment (VA): Menampilkan kontras tinggi dan reproduksi warna yang baik, tetapi sudut pandangnya lebih sempit dibandingkan IPS.



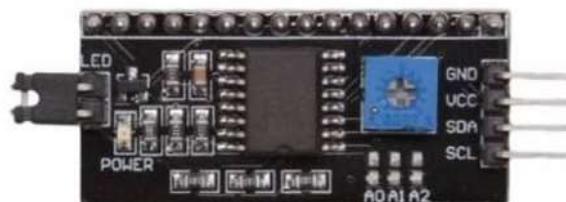
Gambar 2. Liquid Crystal Display (LCD)

I2C/TWI LCD 1602

Modul LCD (Liquid Crystal Display) 16x2 I2C/TWI LCD 1602 menggunakan protokol komunikasi I2C (Inter-Integrated Circuit) atau TWI (Two Wire Interface). Modul ini sering digunakan untuk menampilkan teks pada proyek mikrokontroler seperti Arduino.

Keuntungan Penggunaan I2C pada LCD 1602:

1. Penghematan Pin I/O: Dengan antarmuka I2C, hanya dua pin yang dibutuhkan untuk komunikasi, dibandingkan dengan antarmuka paralel yang memerlukan lebih banyak pin.
2. Kemudahan Integrasi: Memungkinkan koneksi beberapa perangkat I2C pada bus yang sama, asalkan setiap perangkat memiliki alamat unik.
3. Kontrol Backlight dan Kontras: Dilengkapi dengan potensiometer untuk menyesuaikan kontras dan jumper untuk mengontrol backlight, memberikan fleksibilitas dalam pengaturan tampilan.
4. Pin yang akan dihubungkan ke Arduino:
 1. GND : dihubungkan ke GND Arduino
 2. VCC : dihubungkan ke 5V Arduino
 3. SDA : Merupakan I2C data dan dihubungkan ke pin analog pada arduino.
 4. SCL : Merupakan I2C clock dan dihubungkan ke pin analog pada arduino.



Gambar 3. Modul I2C/TWI LCD 1602

Software Arduino IDE

Perangkat lunak resmi yang disebut Software Arduino IDE (Integrated Development Environment) digunakan untuk menulis, mengubah, mengkompilasi, dan mengunggah kode program (sketch) ke papan mikrokontroler Arduino.

Fungsi utama IDE Arduino:

1. Menulis kode program atau sketch menggunakan bahasa pemrograman C/C++ yang telah disederhanakan untuk Arduino.
2. Mengompilasi kode program ke dalam file biner yang dapat dibaca oleh mikrokontroler Arduino.
3. Mengunggah program ke board Arduino langsung melalui kabel USB.
4. Serial Monitor membantu melacak dan mendebugging data dari board secara real-time.

Dengan program Arduino Idea, kita dapat memodelkan parameter rangkaian analog dan digital. Program ini juga memungkinkan kita untuk membuat berbagai desain rangkaian dan menguji rangkaian dengan berbagai komponen yang mungkin, lakukan analisis AC/DC atau transient untuk memeriksa sifat rangkaian secara keseluruhan.

Motor Servo

Motor servo adalah jenis motor listrik yang sering digunakan dalam robotika, otomasi industri, kendaraan pengendali jauh, dan sistem kontrol presisi lainnya, seperti kamera, printer 3D, dan lengan robot.

Motor servo adalah jenis motor listrik yang dirancang untuk mengontrol posisi sudut secara presisi. Berbeda dengan motor DC biasa yang berputar terus menerus, motor servo memiliki kemampuan untuk berhenti pada sudut tertentu sesuai dengan sinyal kontrol yang diterimanya. Komponen utama dalam motor servo meliputi motor DC, rangkaian kontrol elektronik, gear, dan potensiometer yang berfungsi sebagai feedback untuk menentukan posisi sudut.



Gambar 4. Motor Servo

Sensor Inframerah (Infrared)

Sensor inframerah (IR) adalah perangkat elektronik yang mendeteksi radiasi inframerah yang dipancarkan atau dipantulkan oleh objek di sekitarnya. Sensor ini banyak digunakan dalam berbagai aplikasi, seperti deteksi gerakan, pengukuran suhu, dan analisis gas.

Jenis-Jenis Sensor Inframerah:

1. Sensor Inframerah Pasif (PIR): Sensor ini mendeteksi perubahan radiasi inframerah yang berasal dari objek yang bergerak, seperti manusia atau hewan. PIR sering digunakan dalam sistem alarm keamanan dan pencahayaan otomatis.
2. Sensor Inframerah Non-Dispersif (NDIR): Sensor ini digunakan untuk mendeteksi konsentrasi gas tertentu dengan mengukur penyerapan radiasi inframerah pada panjang gelombang spesifik yang sesuai dengan gas tersebut. NDIR banyak digunakan dalam analisis gas dan pemantauan kualitas udara.
3. Sensor Pyroelectric: Sensor ini mendeteksi perubahan suhu yang disebabkan oleh radiasi inframerah yang diterimanya. Sensor pyroelectric sering digunakan dalam aplikasi spektroskopi inframerah dan pencitraan termal.



Gambar 5. Sensor Inframerah

Software Sublime Text

Sublime Text adalah editor teks dan kode sumber yang ringan, cepat, dan serbaguna, yang dirancang untuk pengembang perangkat lunak. Dikembangkan oleh Jon Skinner pada tahun 2008, Sublime Text mendukung berbagai bahasa pemrograman, menyediakan fitur-fitur canggih untuk meningkatkan produktivitas programmer.

Bahasa Pemrograman Python

Python adalah bahasa pemrograman tingkat tinggi yang dirancang dengan fokus pada kemudahan penggunaan dan keterbacaan kode. Python bersifat open-source dan dapat digunakan secara gratis di berbagai sistem operasi seperti Windows, Linux, dan macOS

Command Prompt

Command Prompt (CMD) pada laptop adalah antarmuka berbasis teks (command-line interface) di system operasi Windows yang memungkinkan pengguna menjalankan perintah untuk mengelola sistem, file, program, dan jaringan. Command Prompt digunakan untuk melakukan tugas administratif, troubleshooting, atau menjalankan program yang tidak memiliki antarmuka grafis.

Training.xml

training.xml adalah sebuah file yang berisi model data hasil pelatihan (training) dari algoritma pembelajaran mesin (machine learning) yang digunakan untuk menyimpan informasi model yang dilatih menggunakan data wajah, seperti fitur-fitur wajah pada label yang sesuai.

Dataset internal

Dataset internal yang terdiri dari gambar wajah asli yang telah disimpan (foto).

Source Coding Python

Python adalah bahasa pemrograman tingkat tinggi, serbaguna, dan mudah dipelajari. Python dirancang oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Python dikenal dengan sintaksnya yang sederhana, bersih, dan mirip bahasa Inggris, sehingga sangat cocok untuk pemula maupun pengembang profesional. Python digunakan dalam berbagai aplikasi, seperti pengembangan web, kecerdasan buatan (AI), pembelajaran mesin (ML), analisis data, komputasi ilmiah, dan pengembangan perangkat lunak.

Library

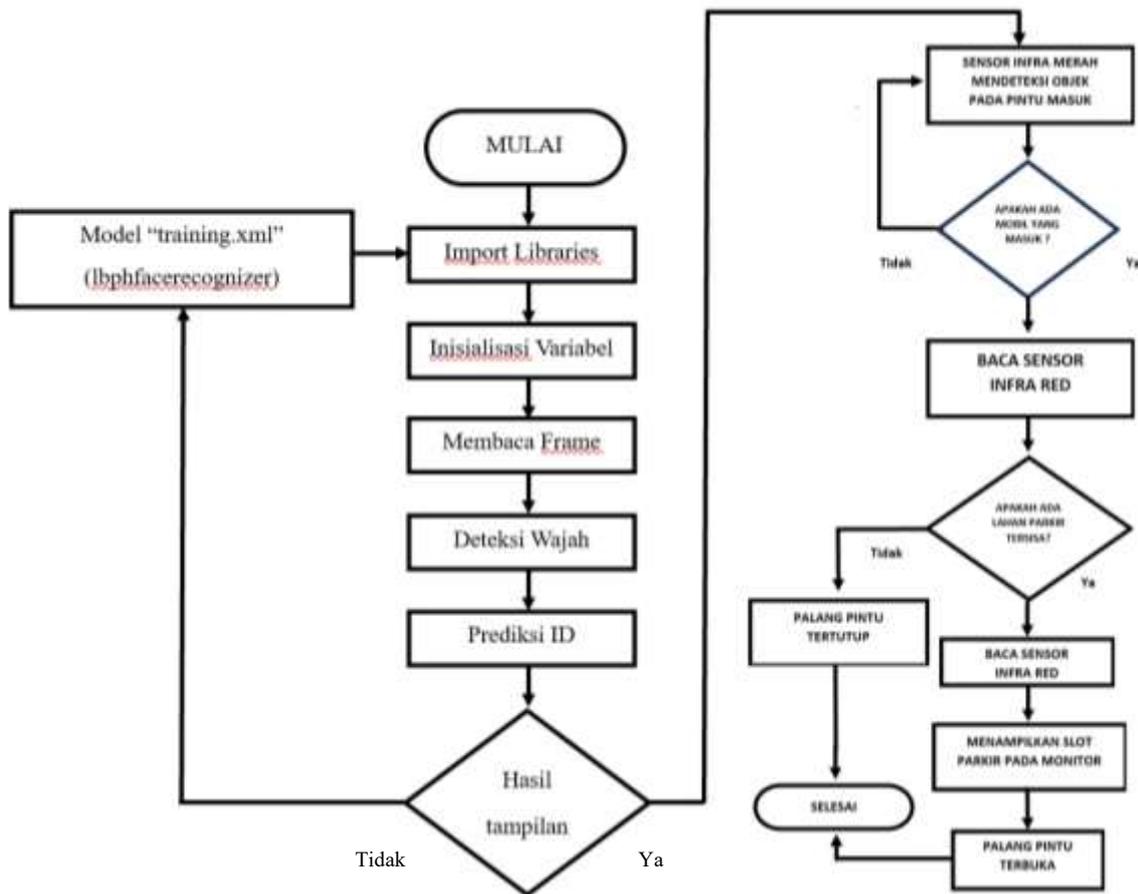
Ada beberapa library yang digunakan pada system:

1. Library os: dalam bahasa pemrograman Python digunakan untuk berinteraksi dengan sistem operasi. Library ini memungkinkan Anda untuk melakukan operasi seperti mengelola file dan direktori, menjalankan perintah sistem, dan mendapatkan informasi lingkungan sistem.
2. Library Numpy (Numerical Python): digunakan untuk komputasi numerik yang efisien, terutama untuk manipulasi array dan operasi matematis.
3. Library Pillow: pustaka Python untuk pengolahan gambar. Pillow merupakan penerus dari Pustaka Python Imaging Library (PIL).

Model Haar Cascade

Haar Cascade adalah metode berbasis machine learning untuk deteksi objek, termasuk wajah, mata, dan senyuman. Metode ini menggunakan serangkaian fitur sederhana (Haar-like features) yang diekstraksi dari gambar untuk mendeteksi objek tertentu.

Flowchart



Gambar 6. Flowchart

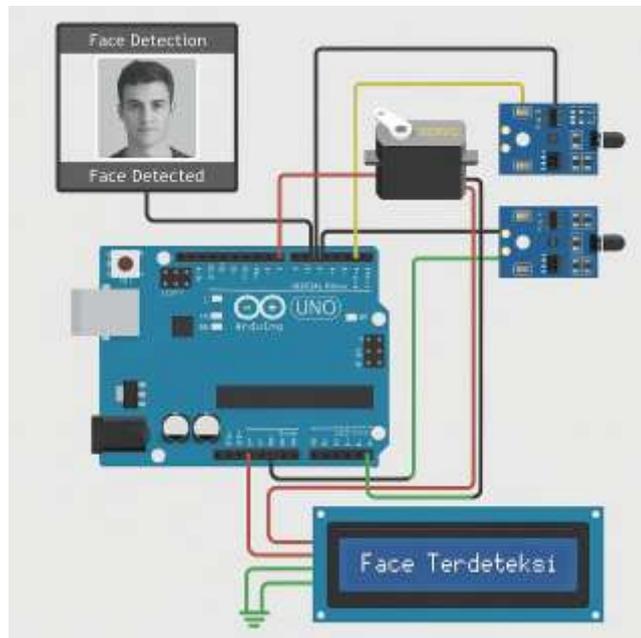
Flowchart di atas menggambarkan proses otomatisasi pada sistem parkir menggunakan sensor infra merah. Berikut adalah penjelasan hubungan antar bagian:

1. Mulai: Mengimport kode python.
2. Inisialisasi Variabel:
Inisialisasi kamera (video feed) dan model Haar Cascade untuk mendeteksi wajah. Juga, memuat model yang dilatih (training.xml) menggunakan LBPHFaceRecognizer.
3. Membaca Frame:
Kamera aktif dan setiap frame video dibaca untuk pemrosesan lebih lanjut.
4. Deteksi Wajah:
Menggunakan Haar Cascade untuk mendeteksi wajah dalam frame. Wajah yang terdeteksi direpresentasikan dalam bentuk koordinat persegi panjang.
5. Prediksi Id:
Model LBPHFaceRecognizer digunakan untuk memprediksi ID dari wajah yang terdeteksi berdasarkan data yang telah diinput
6. Hasil Tampilan:
Wajah yang dikenali diberikan label ID dan ditampilkan dilayar bersama frame video. Jika wajah tidak dikenali akan menampilkan (“Akses Ditolak”). Jika wajah dikenali, maka lanjut ke proses selanjutnya.
7. Baca Sensor Infra Merah:
Sistem membaca data untuk memeriksa apakah ada slot parkir kosong yang tersedia. LCD akan menampilkan teks “Selamat Datang”, dan akan menampilkan jumlah slot parkir tersisa. Dan sistem akan melanjutkan ke proses selanjutnya.

8. Sensor Infra Merah Mendeteksi Objek pada Pintu Masuk:
Sensor infra merah digunakan untuk mendeteksi keberadaan objek (mobil) yang mendekati atau berada di pintu masuk lahan parkir.
9. Apakah Ada Mobil yang Masuk?
Jika ya, sistem melanjutkan proses untuk membaca data dari sensor infra merah lebih lanjut. Tetapi jika tidak ada mobil yang masuk, sistem akan terus memantau kondisi menggunakan sensor infra merah.
10. Apakah Ada Lahan Parkir Tersisa?
Jika ya, sistem melanjutkan ke langkah berikutnya.
Jika tidak ada slot parkir kosong, palang pintu tertutup, dan sistem mengakhiri proses.
11. Menampilkan Slot Parkir pada Monitor: Slot parkir yang kosong ditampilkan di monitor untuk membantu pengemudi memilih lokasi parkir.
12. Palang Pintu Terbuka: Setelah slot parkir tersedia, sistem membuka palang pintu untuk memungkinkan mobil masuk ke area parkir.
13. Selesai: Proses selesai setelah mobil berhasil masuk atau sistem menolak kendaraan karena tidak ada slot parkir yang tersedia.

Setiap langkah terhubung secara logis untuk memastikan kendaraan hanya diperbolehkan masuk jika data wajah sesuai dengan data yang diinput sebelumnya dan jika ada slot parkir kosong. Sensor infra merah adalah komponen utama yang memonitor keberadaan kendaraan dan ketersediaan slot parkir, sementara monitor dan palang pintu bertindak sebagai alat bantu dan pengontrol untuk memfasilitasi proses parkir.

SKEMA RANGKAIAN



Gambar 7. Skema Rangkaian

Gambar rangkaian di atas menunjukkan sistem deteksi wajah otomatis menggunakan Arduino Uno, yang dikendalikan oleh input dari kamera atau modul deteksi wajah (berbasis komputer/AI), dan dilengkapi dengan servo motor serta LCD untuk menampilkan status.

Penjelasan Skema Rangkaian:

1. Deteksi Wajah (Face Detection): Sistem menggunakan kamera (terhubung ke komputer atau modul AI) untuk mendeteksi wajah. Jika wajah terdeteksi, sistem akan mengirim sinyal ke Arduino Uno.
2. Arduino Uno: Berfungsi sebagai otak dari sistem. Arduino menerima sinyal dari deteksi wajah, lalu memprosesnya untuk memberikan perintah ke komponen lain.

4. Sensor Ultrasonik (2 Unit): Terhubung ke Arduino, kemungkinan digunakan untuk mendeteksi keberadaan objek atau mengonfirmasi keberadaan orang di depan sensor secara fisik (jarak dekat), mendukung keputusan dari deteksi wajah.
5. Servo Motor: Servo motor akan bergerak (misalnya membuka pintu otomatis atau menggerakkan mekanisme tertentu) saat wajah berhasil dikenali. Digerakkan oleh sinyal dari pin output Arduino.
6. LCD 16x2: LCD menampilkan informasi status seperti "Face Terdeteksi" untuk memberitahu pengguna bahwa sistem telah mengenali wajah.
7. Alur Kerja Singkat:

Kamera mendeteksi wajah → Arduino menerima sinyal → Arduino mengaktifkan servo (misal: membuka pintu) → LCD menampilkan "Face Terdeteksi" → Sensor ultrasonik bisa memberikan konfirmasi tambahan untuk keamanan atau keakuratan.

Rangkaian ini adalah sistem akses otomatis berbasis pengenalan wajah, di mana Arduino mengendalikan perangkat output (servo & LCD) berdasarkan input dari deteksi visual dan sensor jarak. Sangat cocok digunakan dalam sistem keamanan modern seperti akses pintu otomatis atau kontrol masuk ruang tertentu.

HASIL DAN PEMBAHASAN

Source Code

i. Bagian pertama:

```
# Face Detection and Prediction
import cv2, time
camera = 0
video = cv2.VideoCapture(camera, cv2.CAP_DSHOW)
faceDeteksi = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
id = input('Masukan id : ')
a = 0
while True:
    a = a+1
    check, frame = video.read()
    abu = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    wajah = faceDeteksi.detectMultiScale(abu,1.3,5)
    for(x,y,w,h) in wajah:
        cv2.imwrite('Dataset/User.'+str(id)+'.'+str(a)+'.jpg',abu[y:y+h,x:x+w])
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
    cv2.imshow("Face Recognition", frame)
    if (a>29):
        break
video.release()
cv2.destroyAllWindows()
```

Penjelasan *Face Detection and Prediction*:

Proses ini digunakan untuk mendeteksi wajah dan memprediksi identitasnya.

1. Import Libraries: Mengimpor pustaka seperti OpenCV, time, os, dan Pillow untuk pengolahan citra dan manipulasi file.
2. Initialize Variables: Inisialisasi kamera (video feed) dan model Haar Cascade untuk deteksi wajah. Juga, memuat model yang telah dilatih (training.xml) menggunakan LBPHFaceRecognizer.
3. Capture Video Feed: Kamera aktif dan setiap frame video dibaca untuk pemrosesan lebih lanjut.
4. Detect Faces: Menggunakan Haar Cascade untuk mendeteksi wajah dalam frame. Wajah yang terdeteksi direpresentasikan dalam bentuk koordinat persegi panjang.

5. Predict IDEModel LBPHFaceRecognizer digunakan untuk memprediksi ID dari wajah yang terdeteksi berdasarkan data yang telah dilatih.
6. Display ResultWajah yang dikenali diberikan label ID dan ditampilkan di layar bersama frame video.
7. Exit LoopProgram akan berhenti jika pengguna menekan tombol tertentu (misalnya a).

b. Bagian kedua:

```
# Dataset Creation
import cv2, os
import numpy as np
from PIL import Image
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
def getImagesWithLabels(path):
    imagePath=[os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    Ids=[]
    for imagePath in imagePath:
        pilImage=Image.open(imagePath).convert('L')
        imageNp=np.array(pilImage,'uint8')
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        faces=detector.detectMultiScale(imageNp)
        for (x,y,w,h) in faces:
            faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples, Ids
faces, Ids = getImagesWithLabels('Dataset')
recognizer.train(faces, np.array(Ids))
recognizer.save('Dataset/training.xml')
```

Penjelasan program *Dataset Creation*:

Proses ini digunakan untuk membuat dataset dengan mengambil gambar wajah dari kamera dan menyimpannya.

1. Import Libraries: Mengimpor pustaka yang sama seperti di langkah sebelumnya.
2. Initialize Variables: Mengatur kamera (video feed) dan memuat model deteksi wajah menggunakan Haar Cascade.
3. Input IDE: Meminta pengguna untuk memasukkan ID unik yang akan digunakan untuk menyimpan dataset wajah.
4. Capture Video Feed: Kamera aktif, dan frame video dibaca untuk pengolahan.
5. Detect and Save Faces: Wajah yang terdeteksi dipotong dari frame berdasarkan koordinat yang diberikan oleh Haar Cascade. Gambar-gambar ini disimpan di folder Dataset dengan format User.<ID>.<nomor_urut>.jpg.
6. Exit Loop: Setelah sejumlah gambar (misalnya 30) dikumpulkan, proses perekaman berhenti.

c. Bagian ketiga:

```
#Training the Model
import cv2, time
import os
from PIL import Image
camera = 0
video = cv2.VideoCapture(camera, cv2.CAP_DSHOW)
faceDeteksi = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('Dataset/training.xml')
a = 0
while True:
    a = a+1
    check, frame = video.read()
    abu = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    wajah = faceDeteksi.detectMultiScale(abu,1.3,5)
    for(x,y,w,h) in wajah:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
        id, conf = recognizer.predict(abu[y:y+h,x:x+w])
        if (id==1):
            id = 'ADI'
        elif (id==2):
            id = 'HANIEL'
        elif (id==3):
            id = 'DATA TIDAK DIKENALI'

        cv2.putText(frame,str(id),(x+40,y-10),
cv2.FONT_HERSHEY_DUPLEX,1,(255,255,255))
        cv2.imshow("Face Recognition", frame)
        key = cv2.waitKey(1)
        if key == ord("a"):
            break
    video.release()
cv2.destroyAllWindows()
```

Penjelasan program *Training the Model*:

Proses ini digunakan untuk mencocokkan model pengenalan wajah berdasarkan dataset yang telah dibuat.

1. Import Libraries: Mengimpor pustaka OpenCV, os, numpy, dan Pillow untuk manipulasi gambar dan pelatihan model.
2. Load Dataset: Memuat semua gambar dari folder Dataset. File yang diambil memiliki format yang sesuai (User.<ID>.<nomor>.jpg).
3. Extract Features: Setiap gambar diubah ke skala abu-abu, dan wajah diidentifikasi menggunakan Haar Cascade. Koordinat ROI (Region of Interest) wajah digunakan untuk memotong area wajah. ID pengguna diambil dari nama file.
4. Train Recognizer: Data wajah dan ID digunakan untuk melatih model LBPHFaceRecognizer. Model ini menghasilkan fitur numerik untuk mengenali pola wajah.
5. Save Model: Model yang telah dilatih disimpan dalam file training.xml untuk digunakan dalam prediksi di masa mendatang.

Menjalankan Program

Dari source coding diatas, langkah selanjutnya adalah membuka Command Prompt (CMD) dan menginput file yang telah disimpan pada folder dengan menginput perintah pada command prompt: `cd C:\Windows\System32\Adi\jurnal\Koding`.

```
C:\Windows\System32>cd C:\Windows\System32\Adi\jurnal\Koding
C:\Windows\System32\Adi\jurnal\Koding>
```

Gambar 8. Command Prompt bagian pertama

Selanjutnya buat perintah “`rekam.py`” untuk merekam model yang akan disimpan ke Database “`training.xml`”. Selanjutnya user akan diminta memasukkan inputan id. Id : 1 untuk username “Adi”, Id : 2 untuk username “Haniel”. Input id berguna untuk menjalankan program agar merekam wajah lalu diproses dan dimasukkan ke dalam database.

```
C:\Windows\System32\Adi\jurnal\Koding>rekam.py
Masukan id : 1
C:\Windows\System32\Adi\jurnal\Koding>_
```

Gambar 9. Command Prompt bagian kedua

```
C:\Windows\System32\Adi\jurnal\Koding>rekam.py
Masukan id : 1
C:\Windows\System32\Adi\jurnal\Koding>training.py
C:\Windows\System32\Adi\jurnal\Koding>
```

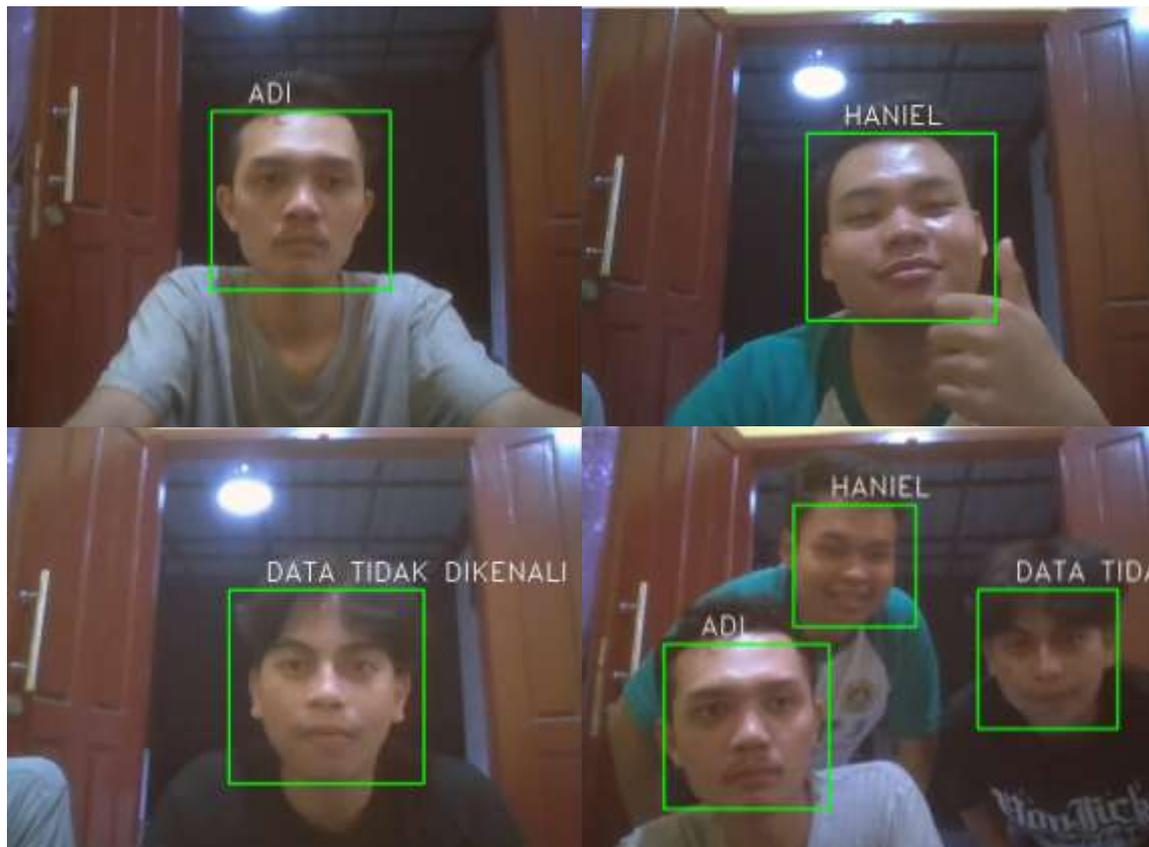
Gambar 10. Command Prompt bagian ketiga

Setelah selesai, masukkan perintah “`training.py`” untuk menyimpan data model yang telah diinputkan melalui perintah “`input id`” dan disimpan ke dalam data base.

```
C:\Windows\System32\Adi\jurnal\Koding>rekam.py
Masukan id : 1
C:\Windows\System32\Adi\jurnal\Koding>training.py
C:\Windows\System32\Adi\jurnal\Koding>main.py
C:\Windows\System32\Adi\jurnal\Koding>_
```

Gambar 11. Command Prompt bagian keempat

Setelah kedua langkah sebelumnya selesai, langkah terakhir mengetik perintah “main.py” dan akan menghasilkan output dibawah ini:



Gambar 12. Hasil Output Data Pendeteksi Wajah

Dari hasil pengolahan data dari sistem pengenalan wajah dapat dilihat bahwa program berhasil mendeteksi beberapa contoh wajah yang sudah diinputkan ke dalam library sebagai data utama dan satu lainnya belum diinput dalam library sistem pendeteksi. Saat data sudah sesuai dengan yang diinginkan, maka sistem akan melanjutkan prosesnya dengan mengirimkan sinyal pengoperasian ke LCD dan motor servo melalui sensor IR yang telah diprogram melalui Arduino UNO. Sensor akan mendeteksi ruang parkir yang tersedia dengan menjadikan kendaraan sebagai objek penghitungan jumlah kendaraan yang masuk dan keluar. Jika tempat masih tersisa maka motor servo sebagai media peralatan akan membuka palang parkir dan kendaraan bisa masuk untuk parkir. Sebaliknya, jika ruang parkir penuh, maka palang pintu tidak akan terbuka.

Jumlah kapasitas dari lahan parkir sama dengan jumlah data wajah yang tersimpan dalam library OpenCV, sehingga berapapun data yang diinputkan ke dalam library, maka itu juga jumlah yang digunakan dalam kapasitas lahan parkir yang disediakan.

KESIMPULAN

Teknologi pengenalan wajah dapat digunakan oleh sistem yang dirancang untuk menyediakan solusi otomatis untuk mengelola akses masuk kendaraan. Hasil implementasi dan pengujian menunjukkan bahwa sistem dapat mengenali wajah pengguna dengan akurasi yang cukup baik dalam kondisi pencahayaan yang mendukung dan sudut wajah yang sesuai. Selain itu, jika wajah yang dikenali terdaftar dalam database sistem, sistem dapat secara otomatis membuka pintu parkir. Seperti yang diharapkan, integrasi antara kamera, pemrosesan wajah dengan Python (menggunakan library seperti OpenCV dan face recognition), dan aktuator seperti servo atau relay yang terhubung ke pengendali gerbang (misalnya dengan Arduino atau Raspberry Pi). Selama diskusi, diketahui bahwa

elemen seperti kualitas kamera, pencahayaan lingkungan, dan kecepatan pemrosesan sangat memengaruhi kinerja sistem secara keseluruhan.

Sistem ini dapat ditingkatkan untuk pengembangan lebih lanjut dengan menambahkan fitur verifikasi ganda, seperti pengenalan nomor plat kendaraan untuk meningkatkan keamanan, dan menggunakan model deep learning yang lebih akurat untuk menangani berbagai variasi ekspresi dan kondisi wajah. Selain itu, ada potensi pengembangan lain yang dapat membuat sistem ini lebih adaptif dan komprehensif dalam penerapan di lingkungan bisnis yang lebih luas dan kompleks. Salah satu contohnya adalah integrasi sistem dengan log aktivitas parkir, database kehadiran karyawan, dan notifikasi real-time ke manajemen perusahaan atau pihak keamanan.

DAFTAR PUSTAKA

- Aditia, R. Ilham, and J. P. Sembiring, "Penetas Telur Otomatis Berbasis Arduino dengan Menggunakan Sensor DHT11," *J. Ilm. Mhs. Kendali dan List.*, vol. 3, no. 1, pp. 113-119, 2022, [Online].
- F. A. Husain, H. S. Nuryono, dan R. Rahmadani, "Sistem Absensi Berbasis Face Recognition Menggunakan Python dan OpenCV," *Jurnal INSTEK*, vol. 8, no. 2, 2023. [Online]. Tersedia: <https://journal.uin-alauddin.ac.id/index.php/instek/article/view/50354>
- Harry G. Walton dan David Dunmur. (2024). Britannica. (n.d.) "Liquid Crystal Display (LCD)". Diakses dari: <https://www.britannica.com/technology/liquid-crystal-display>
- H. R. P. Putra, A. S. Hutavea, dan Y. P. Y. Utama. 2025. PENGEMBANGAN SISTEM DETEKSI RUANG PARKIR OTOMATIS MENGGUNAKAN SENSOR INFRAMERAH DAN TEKNOLOGI ARDUINO. *Jurnal Teknik Elektro Universitas Tulungagung*, 1(1), 3-6. <https://journal.unita.ac.id/index.php/elijita/article/view/1513>
- Rahman, A., & Wahyuni, S. (n.d.). Implementasi aplikasi Arduino IDE pada mata kuliah sistem digital. *Jurnal Teknologi dan Pendidikan*, 2(1), 45–52. Diakses pada 13 Januari 2025, dari: <https://jurnal-fkip-uim.ac.id/index.php/teknos/article/download/40/42>.
- "Unlocking the Potential of Face Recognition in OpenCV: A Comprehensive Study of Algorithmic Approaches," *ACM Digital Library*, publikasi tahun 2024, DOI: 10.1145/3641032.3641056.