

BAB II **TINJAUAN LITERATUR**

2.1. TINJAUAN PUSTAKA

Penelitian mengenai sistem absensi berbasis deteksi dan pengenalan wajah telah banyak dilakukan sebagai solusi atas keterbatasan metode absensi manual. Kenda dan Witanti (2022) melakukan penelitian tentang sistem presensi berbasis wajah menggunakan metode Haar Cascade pada OpenCV. Penelitian ini bertujuan menggantikan sistem absensi manual yang dinilai tidak efektif dan rawan kecurangan. Hasil penelitian menunjukkan bahwa sistem mampu mengenali wajah dengan tingkat akurasi mencapai 93% pada kondisi pencahayaan yang baik. Penelitian ini membuktikan bahwa algoritma Haar Cascade efektif digunakan untuk mendeteksi wajah secara real-time dalam sistem presensi berbasis komputer.

Penelitian lain dilakukan oleh Zein (2023) yang mengembangkan sistem absensi cerdas menggunakan OpenCV berbasis pengenalan wajah dengan algoritma LBPH. Sistem ini mengintegrasikan proses pengambilan citra melalui kamera, deteksi wajah, ekstraksi fitur, hingga pencatatan absensi secara otomatis dalam bentuk laporan digital. Hasil penelitian menunjukkan bahwa algoritma LBPH memiliki performa yang stabil dan cukup akurat dalam mengenali wajah siswa, serta unggul dibandingkan metode lain dalam kondisi variasi pencahayaan. Penelitian ini menegaskan bahwa kombinasi metode deteksi wajah dan pengenalan wajah berbasis OpenCV sangat sesuai untuk sistem absensi di lingkungan pendidikan .

Selanjutnya, Vernanda, Metandi, dan Syafrizal (2025) melakukan penelitian rancang bangun sistem absensi siswa menggunakan deteksi wajah pada SMP Negeri

006 Long Ikis. Sistem dikembangkan berbasis web dengan pemanfaatan kamera webcam untuk mendeteksi dan mengenali wajah siswa secara real-time. Pengujian menunjukkan bahwa sistem mampu mendeteksi wajah dengan tingkat akurasi tinggi pada berbagai kondisi pencahayaan dan posisi wajah. Penelitian ini menekankan bahwa penerapan deteksi wajah dapat meningkatkan efisiensi, akurasi, dan keamanan data presensi dibandingkan metode konvensional.

Novianti, Priyangan, dan Pramestiana (2024) melakukan penelitian mengenai pembangunan sistem aplikasi absensi berbasis deteksi wajah menggunakan OpenCV di Kampus STMIK Kalirejo Lampung. Sistem yang dikembangkan memanfaatkan webcam untuk menangkap citra wajah dan mencocokkannya dengan data yang tersimpan pada database sehingga dapat mengidentifikasi kehadiran dosen dan staf secara otomatis. Hasil penelitian menunjukkan bahwa sistem absensi berbasis pengenalan wajah mampu mempercepat proses pencatatan kehadiran, meminimalkan manipulasi data, serta mempermudah proses pemantauan dan pembuatan laporan kehadiran dibandingkan dengan sistem absensi manual.

Putra, Opitasari, dan Parwati (2025) melakukan penelitian mengenai pengembangan sistem absensi berbasis web menggunakan metode deteksi wajah dengan OpenCV di TK Az-Zahra. Sistem dirancang dengan bahasa pemrograman Python dan platform web sehingga dapat diakses oleh administrator maupun pengguna untuk mengelola data kehadiran secara real-time. Hasil penelitian menunjukkan bahwa sistem ini mampu meningkatkan efisiensi pengelolaan absensi, mempermudah proses pencatatan dan pelaporan kehadiran, serta mengurangi potensi kecurangan dalam absensi melalui identifikasi wajah otomatis

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa teknologi deteksi dan pengenalan wajah menggunakan OpenCV terbukti efektif untuk diterapkan pada sistem absensi. Sistem yang dikembangkan mampu mengurangi kecurangan, mempercepat proses pencatatan kehadiran, serta meningkatkan objektivitas data absensi. Namun, sebagian besar penelitian masih mengimplementasikan sistem dalam bentuk monolitik dan belum menekankan aspek arsitektur sistem modern.

Perbedaan utama penelitian yang akan dilakukan terletak pada pendekatan arsitektur microservice, di mana proses deteksi dan pengenalan wajah dikembangkan menggunakan Python sebagai backend, sedangkan manajemen absensi dan layanan API dibangun menggunakan Laravel. Selain itu, sistem juga dilengkapi dengan integrasi Telegram Gateway yang berfungsi untuk mengirimkan notifikasi kehadiran siswa kepada orang tua secara otomatis setelah proses absensi selesai. Notifikasi dikirimkan secara real-time melalui aplikasi Telegram sehingga informasi kehadiran dapat diterima dengan cepat dan praktis. Pendekatan ini diharapkan mampu menghasilkan sistem absensi berbasis deteksi wajah yang lebih modular, skalabel, dan mudah dikembangkan, sekaligus meningkatkan transparansi informasi dan komunikasi antara sekolah dan orang tua siswa untuk kebutuhan institusi pendidikan di masa mendatang.

Berikut tabel perbandingan penelitian terdahulu dengan penelitian yang akan dilakukan:

Tabel 2. 1 Tinjauan Pustaka

Nama Peneliti	Judul Penelitian	Hasil Penelitian	Perbedaan dengan Penelitian yang Akan Dilakukan
Zein (2023)	Sistem Absensi Cerdas Menggunakan OpenCV Berbasis Pengenalan Wajah dengan Algoritma LBPH	Sistem berhasil mengintegrasikan proses pengambilan citra, deteksi wajah, ekstraksi fitur, hingga pencatatan absensi secara otomatis. Algoritma LBPH memiliki performa yang stabil dan akurat, terutama pada kondisi pencahayaan yang bervariasi sehingga sesuai diterapkan pada sistem absensi di lingkungan pendidikan.	Penelitian yang akan dilakukan tidak hanya menerapkan OpenCV untuk deteksi dan pengenalan wajah, tetapi juga menggunakan arsitektur microservice dengan Python sebagai layanan deteksi wajah, Laravel sebagai backend manajemen absensi, serta integrasi Telegram Gateway untuk mengirim notifikasi kehadiran secara real-time kepada orang tua siswa.
Vernanda, Metandi, dan Syafrizal (2025)	Rancang Bangun Sistem Absensi Siswa Menggunakan Deteksi Wajah pada SMP Negeri 006 Long Ikis	Sistem berbasis web yang memanfaatkan webcam mampu mendeteksi dan mengenali wajah siswa secara real-time dengan tingkat akurasi tinggi pada berbagai kondisi pencahayaan dan posisi wajah. Sistem juga meningkatkan efisiensi, akurasi, dan keamanan data presensi.	Penelitian yang akan dilakukan menerapkan arsitektur microservice yang memisahkan layanan deteksi wajah dan layanan manajemen absensi, serta menambahkan integrasi Telegram Gateway untuk penyampaian notifikasi kehadiran kepada orang tua secara otomatis.
Novianti, Priyangan, dan Pramestiana (2024)	Pembangunan Sistem Aplikasi Absensi Berbasis Deteksi Wajah Menggunakan OpenCV di Kampus STMIK Kalirejo Lampung	Sistem mampu mengidentifikasi kehadiran dosen dan staf secara otomatis menggunakan webcam dan basis data wajah. Sistem mempercepat proses pencatatan kehadiran, meminimalkan manipulasi data, serta mempermudah pemantauan dan pelaporan kehadiran.	Penelitian yang akan dilakukan berfokus pada monitoring kehadiran siswa dengan pendekatan arsitektur microservice menggunakan Python dan Laravel, serta menyediakan notifikasi kehadiran melalui Telegram Gateway secara real-time kepada orang tua siswa.

Putra, Opitasari, dan Parwati (2025)	Pengembangan Sistem Absensi Berbasis Web Menggunakan Metode Deteksi Wajah dengan OpenCV di TK Az-Zahra	Sistem berbasis web yang dibangun menggunakan Python mampu meningkatkan efisiensi pengelolaan absensi, mempermudah pencatatan dan pelaporan kehadiran, serta mengurangi potensi kecurangan melalui identifikasi wajah otomatis.	Penelitian yang akan dilakukan tidak hanya membangun sistem berbasis web, tetapi juga menerapkan arsitektur microservice, memisahkan layanan deteksi wajah dengan backend menggunakan Python dan Laravel, serta menambahkan fitur notifikasi Telegram kepada orang tua setelah proses absensi selesai.
---	--	---	--

2.2 LANDASAN TEORI

2.2.1 Rancang Bangun Sistem

Rancang bangun sistem merupakan suatu proses yang dilakukan untuk menghasilkan sebuah sistem informasi yang mampu memenuhi kebutuhan pengguna melalui tahapan analisis, perancangan, implementasi, dan pengujian (Jhonny & Hadiwinata, 2024). Proses ini bertujuan untuk menerjemahkan kebutuhan pengguna menjadi suatu solusi yang terstruktur sehingga sistem yang dibangun dapat berfungsi secara efektif, efisien, dan sesuai dengan tujuan yang telah ditetapkan. Dalam pengembangan sistem, tahap perancangan menjadi bagian yang sangat penting karena menentukan struktur, arsitektur, alur proses, serta hubungan antar komponen yang akan diimplementasikan pada tahap Pembangunan (Sinuraya et al., 2024).

Menurut konsep rekayasa perangkat lunak, rancang bangun sistem tidak hanya berfokus pada pembuatan aplikasi, tetapi juga mencakup penyusunan model sistem, desain basis data, antarmuka pengguna (*user interface*), serta mekanisme pengolahan data agar sistem dapat berjalan secara optimal. Perancangan yang baik akan

mempermudah proses implementasi, pemeliharaan, dan pengembangan sistem di masa mendatang sehingga dapat mengurangi risiko terjadinya kesalahan maupun perubahan yang signifikan selama proses pengembangan (Saputra et al., 2024).

2.2.2 Monitoring Kehadiran Siswa

Monitoring kehadiran siswa adalah proses pemantauan dan pencatatan kehadiran siswa dalam kegiatan pembelajaran di sekolah. Monitoring ini dilakukan untuk mengetahui apakah siswa hadir, terlambat, izin, sakit, atau tidak hadir tanpa keterangan. Kehadiran siswa menjadi salah satu indikator penting dalam menilai kedisiplinan dan keterlibatan siswa dalam proses belajar mengajar.

Monitoring kehadiran siswa memiliki peran penting dalam kegiatan administrasi sekolah. Data kehadiran yang tercatat dengan baik dapat digunakan oleh pihak sekolah untuk membuat laporan absensi, mengevaluasi tingkat kedisiplinan siswa, serta menjadi bahan pertimbangan dalam proses pembinaan siswa. Selain itu, data kehadiran juga dapat membantu pihak sekolah dalam mengetahui pola ketidakhadiran siswa secara lebih terstruktur.

Dalam pelaksanaannya, monitoring kehadiran siswa dapat dilakukan secara manual maupun menggunakan sistem berbasis teknologi. Monitoring secara manual biasanya dilakukan melalui pencatatan pada buku absensi atau lembar kehadiran, sedangkan monitoring berbasis teknologi dilakukan dengan bantuan sistem komputer, aplikasi, atau perangkat digital. Penggunaan teknologi dalam monitoring kehadiran dapat membantu mempercepat proses pencatatan, mengurangi kesalahan data, serta memudahkan penyajian laporan kehadiran.

2.2.3 Deteksi wajah (*Face Recognition*)

Deteksi wajah merupakan sistem biometrik yang digunakan untuk mengidentifikasi atau memverifikasi seseorang berdasarkan citra digital wajah. Sistem ini bekerja dengan cara mendeteksi wajah pada gambar, mengekstraksi ciri atau fitur wajah, kemudian melakukan proses pengenalan terhadap wajah tersebut (Parmar & Mehta., 2014). Deteksi wajah termasuk bagian dari bidang pengolahan citra dan computer vision karena memanfaatkan gambar atau video sebagai objek utama yang dianalisis.

Deteksi wajah atau pengenalan wajah adalah teknologi yang digunakan untuk mengenali identitas seseorang berdasarkan karakteristik wajah. Teknologi ini bekerja dengan cara menangkap citra wajah melalui kamera, kemudian sistem akan menganalisis pola atau fitur wajah untuk mencocokkannya dengan data wajah yang telah tersimpan sebelumnya.

Face recognition banyak diterapkan pada sistem keamanan, absensi, autentikasi perangkat, hingga sistem pengawasan. Dibandingkan metode biometrik lainnya, seperti sidik jari atau iris mata, pengenalan wajah memiliki keunggulan karena proses identifikasi dapat dilakukan secara non-kontak menggunakan kamera sehingga lebih praktis dan efisien. Secara umum, proses *face recognition* terdiri dari beberapa tahapan seperti berikut:

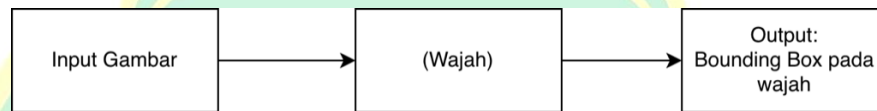
1. Akuisisi Citra

Tahap pertama adalah memperoleh citra wajah menggunakan kamera. Citra yang diperoleh harus memiliki pencahayaan yang cukup serta posisi wajah

menghadap kamera agar proses pengenalan menghasilkan tingkat akurasi yang baik.

2. Deteksi Wajah

Pada tahap ini sistem mendeteksi lokasi wajah pada gambar. Area selain wajah akan diabaikan sehingga hanya bagian wajah yang diproses pada tahap berikutnya.



Gambar 2. 1 Flow Deteksi Wajah

3. Pra – Pemrosesan

Setelah wajah ditemukan, citra akan diproses agar memiliki kualitas yang seragam. Tahapan ini meliputi:

- Resize ukuran gambar
- Konversi warna (RGB menjadi Grayscale apabila diperlukan)
- Normalisasi pencahayaan
- Alignment posisi wajah

Tahapan ini bertujuan meningkatkan kualitas data sehingga proses ekstraksi fitur menjadi lebih akurat.

4. Ekstraksi Fitur (*Feature Extraction*)

Tahap ekstraksi fitur bertujuan memperoleh representasi numerik dari karakteristik wajah. Pada metode modern, wajah diubah menjadi sebuah *embedding vector* yang mewakili ciri khas seseorang.

Misalnya hasil ekstraksi menjadi:

$$F = (f_1, f_2, f_3, \dots, f_n)$$

Dengan:

- F = vector fitur wajah
- f_i = vector ke- i

5. Pencocokan Wajah (*Face Matching*)

Tahap terakhir adalah membandingkan vektor fitur wajah baru dengan data wajah yang tersimpan di dalam database. Salah satu metode yang umum digunakan adalah *Euclidean Distance*:

$$d = \sqrt{\sum_{f=1}^n (x_f - y_f)^2}$$

Keterangan:

- d = jarak antar dua wajah
- x_i = fitur wajah masukan
- y_i = fitur wajah pada database
- n = jumlah fitur

Semakin kecil nilai *Euclidean Distance*, maka kedua wajah semakin mirip

2.2.4 Biometrik

Menurut Jain, Ross, dan Prabhakar, biometrik adalah pengenalan individu secara otomatis berdasarkan karakteristik fisiologis atau perilaku yang dimiliki oleh seseorang. Karakteristik fisiologis dapat berupa wajah, sidik jari, iris mata, retina, dan bentuk tangan, sedangkan karakteristik perilaku dapat berupa suara, tanda tangan, atau pola berjalan. Biometrik digunakan karena setiap individu memiliki ciri

khas yang berbeda sehingga dapat dijadikan sebagai dasar untuk proses identifikasi atau verifikasi identitas.

Menurut Wayman, Jain, Maltoni, dan Maio, sistem biometrik tidak hanya bergantung pada teknologi pengenalan, tetapi juga melibatkan perangkat keras, perangkat lunak, jaringan, prosedur, serta kebijakan yang mendukung proses identifikasi. Dengan demikian, biometrik tidak berdiri sendiri sebagai teknologi, melainkan menjadi bagian dari suatu sistem yang dirancang untuk meningkatkan proses pengenalan identitas manusia secara lebih efektif.

2.2.5 Pengolahan Citra Digital

Menurut Gonzalez dan Woods, pengolahan citra digital merupakan bidang yang mempelajari proses pengolahan gambar menggunakan komputer. Citra digital dapat berupa gambar dua dimensi yang tersusun dari elemen-elemen kecil yang disebut piksel. Setiap piksel memiliki nilai tertentu yang menunjukkan tingkat warna atau intensitas cahaya pada bagian gambar tersebut. Pengolahan citra digital digunakan untuk memperbaiki kualitas gambar, mengekstraksi informasi, serta membantu komputer dalam memahami isi dari suatu gambar.

Pengolahan citra digital banyak digunakan dalam berbagai bidang, seperti kedokteran, keamanan, industri, pendidikan, robotika, dan sistem biometrik. Dalam sistem biometrik, pengolahan citra digital berperan penting dalam proses pengenalan wajah karena citra wajah harus diproses terlebih dahulu sebelum dapat dikenali oleh sistem. Faktor seperti pencahayaan, ketajaman gambar, posisi objek, dan kualitas kamera dapat memengaruhi hasil pengolahan citra digital.

2.2.6 Laravel

Laravel adalah framework berbasis bahasa pemrograman PHP yang digunakan untuk membangun aplikasi web secara lebih cepat, terstruktur, dan mudah dikembangkan. Laravel menyediakan berbagai fitur siap pakai yang membantu pengembang dalam membuat sistem, seperti routing, controller, model, middleware, authentication, migration, validation, dan template engine. Framework ini menggunakan konsep Model View Controller atau MVC yang memisahkan bagian pengolahan data, tampilan, dan logika sistem agar kode program lebih rapi dan mudah dikelola.

Konsep MVC pada Laravel terdiri dari tiga bagian utama, yaitu Model, View, dan Controller. Model berfungsi untuk mengelola data dan berhubungan langsung dengan database. View berfungsi sebagai tampilan antarmuka yang dilihat oleh pengguna. Controller berfungsi sebagai penghubung antara Model dan View, serta mengatur alur proses dari permintaan pengguna. Dengan penerapan MVC, pengembangan aplikasi menjadi lebih terorganisir karena setiap bagian memiliki tugas masing-masing.

2.2.7 Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat open source dan banyak digunakan dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan, machine learning, serta pengolahan citra digital. Python dikenal memiliki sintaks yang sederhana dan mudah dipahami, sehingga memudahkan pengembang dalam menulis, membaca, dan memelihara kode program. Menurut Van Rossum, Python dirancang sebagai bahasa pemrograman yang menekankan

keterbacaan kode serta mendukung berbagai paradigma pemrograman, seperti pemrograman prosedural, pemrograman berorientasi objek, dan pemrograman fungsional.

Menurut Lutz, Python merupakan bahasa pemrograman yang fleksibel dan dapat digunakan untuk berbagai jenis pengembangan aplikasi, mulai dari aplikasi desktop, aplikasi web, otomasi sistem, hingga pemrosesan data. Python memiliki banyak pustaka atau library yang dapat membantu pengembang dalam membangun sistem secara lebih cepat. Beberapa pustaka yang sering digunakan antara lain NumPy untuk komputasi numerik, Pandas untuk pengolahan data, Matplotlib untuk visualisasi data, TensorFlow untuk machine learning, serta OpenCV untuk pengolahan citra dan computer vision.

2.2.8 Library Pendukung Sistem

Library Python merupakan kumpulan modul atau pustaka yang menyediakan berbagai fungsi sehingga memudahkan pengembang dalam membangun suatu aplikasi tanpa harus membuat seluruh kode dari awal. Berbagai library Python telah dikembangkan untuk mendukung kebutuhan pengolahan data, kecerdasan buatan, hingga *computer vision*.

Salah satu library yang banyak digunakan dalam bidang *computer vision* adalah OpenCV (*Open Source Computer Vision Library*). OpenCV atau *Open Source Computer Vision Library* merupakan pustaka perangkat lunak bersifat open source yang digunakan untuk kebutuhan pengolahan citra digital dan computer vision. Menurut Bradski dan Kaehler, OpenCV dirancang untuk menyediakan infrastruktur umum dalam membangun aplikasi computer vision secara cepat,

efisien, dan fleksibel. OpenCV banyak digunakan untuk berbagai kebutuhan seperti deteksi objek, pengenalan wajah, pelacakan gerakan, pengolahan video, kalibrasi kamera, segmentasi citra, hingga analisis bentuk dan pola pada gambar digital.

Salah satu proses dasar dalam OpenCV adalah konversi citra berwarna menjadi grayscale. Konversi grayscale dilakukan agar proses pengolahan citra menjadi lebih sederhana karena citra hanya memiliki satu kanal intensitas. Rumus umum konversi RGB ke grayscale adalah:

$$Y = 0,299R + 0,587G + 0,114B$$

Keterangan dari rumus tersebut adalah Y sebagai nilai intensitas grayscale, R sebagai nilai warna merah, G sebagai nilai warna hijau, dan B sebagai nilai warna biru. Nilai grayscale ini digunakan karena mata manusia lebih sensitif terhadap warna hijau dibandingkan warna merah dan biru, sehingga bobot warna hijau lebih besar dalam perhitungan.

Pada proses deteksi wajah, OpenCV dapat menggunakan metode Haar Cascade Classifier. Metode ini bekerja dengan cara mendeteksi pola tertentu pada wajah, seperti area mata, hidung, mulut, dan perbedaan intensitas antara bagian terang dan gelap pada wajah. Haar Cascade menggunakan fitur Haar-like yang menghitung perbedaan nilai intensitas piksel pada beberapa area persegi panjang. Rumus sederhana fitur Haar dapat dituliskan sebagai:

$$F = \Sigma \text{ area putih} - \Sigma \text{ area hitam}$$

Nilai F menunjukkan hasil perbedaan antara jumlah piksel pada area putih dan area hitam. Jika nilai fitur sesuai dengan pola wajah yang telah dilatih sebelumnya, maka sistem dapat mendeteksi adanya wajah pada citra tersebut.

Untuk mempercepat proses perhitungan fitur Haar, OpenCV menggunakan konsep integral image. Integral image digunakan agar jumlah nilai piksel pada suatu area dapat dihitung dengan cepat. Rumus integral image adalah:

$$\Pi(x, y) = \sum I(x', y')$$

Dengan ketentuan $x' \leq x$ dan $y' \leq y$. $\Pi(x, y)$ merupakan nilai integral image pada titik koordinat x dan y , sedangkan $I(x', y')$ adalah nilai intensitas piksel asli. Dengan teknik ini, OpenCV dapat menghitung jumlah piksel pada area tertentu tanpa harus menjumlahkan semua piksel satu per satu secara berulang.

Setelah wajah terdeteksi, proses berikutnya dalam sistem pengenalan wajah adalah ekstraksi fitur wajah. Pada beberapa metode deteksi wajah, OpenCV dapat menggunakan algoritma seperti Eigenfaces, Fisherfaces, dan Local Binary Pattern Histogram atau LBPH. Algoritma LBPH cukup sering digunakan karena relatif sederhana dan dapat bekerja pada kondisi pencahayaan yang bervariasi. LBPH bekerja dengan membandingkan nilai piksel pusat dengan piksel di sekitarnya. Jika nilai piksel tetangga lebih besar atau sama dengan piksel pusat, maka diberi nilai 1, sedangkan jika lebih kecil diberi nilai 0.

Rumus dasar Local Binary Pattern dapat dituliskan sebagai:

$$LBP(xc, yc) = \sum s(ip - ic) 2^p$$

dengan keterangan ic adalah nilai piksel pusat, ip adalah nilai piksel tetangga, p adalah indeks piksel tetangga, dan s adalah fungsi pembanding. Fungsi s dapat dituliskan sebagai:

$$s(x) = 1, \text{ jika } x \geq 0$$

$$s(x) = 0, \text{ jika } x < 0$$

Hasil dari proses LBP kemudian dibuat dalam bentuk histogram. Histogram tersebut digunakan sebagai ciri atau fitur wajah yang akan dibandingkan dengan data wajah yang telah tersimpan. Semakin kecil jarak perbedaan antara histogram wajah yang diuji dengan histogram wajah pada database, maka semakin besar kemungkinan wajah tersebut dikenali sebagai orang yang sama.

Pada tahap pencocokan wajah, OpenCV membandingkan fitur wajah baru dengan fitur wajah yang telah dilatih sebelumnya. Salah satu pendekatan perhitungan jarak yang dapat digunakan adalah Euclidean Distance. Rumus Euclidean Distance adalah:

$$d = \sqrt{\sum (x_i - y_i)^2}$$

Keterangan dari rumus tersebut adalah d sebagai nilai jarak, x_i sebagai nilai fitur citra wajah yang diuji, dan y_i sebagai nilai fitur citra wajah yang tersimpan pada database. Jika nilai jarak semakin kecil, maka tingkat kemiripan wajah semakin tinggi. Sebaliknya, jika nilai jarak semakin besar, maka wajah dianggap semakin berbeda.

2.2.9 Telegram Gateway

Telegram Gateway adalah layanan penghubung antara sistem aplikasi dengan platform Telegram yang digunakan untuk mengirimkan pesan atau notifikasi secara otomatis. Telegram Gateway memanfaatkan Telegram Gateway sebagai media komunikasi antara sistem dan pengguna. Dengan adanya gateway, aplikasi dapat mengirimkan informasi tertentu ke akun atau grup Telegram tanpa harus dilakukan secara manual oleh pengguna.

Telegram Gateway merupakan akun khusus pada Telegram yang dapat dikendalikan oleh sistem melalui Bot API. Bot API memungkinkan aplikasi untuk mengirim pesan, menerima perintah, membaca respons pengguna, serta melakukan interaksi otomatis lainnya. Bot Telegram biasanya dibuat melalui BotFather, yaitu layanan resmi Telegram yang digunakan untuk membuat bot dan memperoleh token sebagai kunci akses agar sistem dapat berkomunikasi dengan bot tersebut.

Dalam proses kerjanya, Telegram Gateway membutuhkan beberapa komponen utama, yaitu token bot, chat ID, koneksi internet, serta perintah pengiriman pesan dari sistem. Token bot berfungsi sebagai identitas dan akses autentikasi bot, sedangkan chat ID digunakan untuk menentukan tujuan pengiriman pesan, baik kepada pengguna pribadi maupun grup. Setelah sistem menghasilkan data atau informasi tertentu, sistem akan mengirimkan request ke Telegram Gateway API, kemudian Telegram akan meneruskan pesan tersebut kepada penerima sesuai chat ID yang dituju.

2.2.10 Database MySQL

Database adalah kumpulan data yang disimpan secara terstruktur di dalam sistem komputer sehingga dapat diakses, dikelola, dan diperbarui dengan mudah. Database digunakan untuk menyimpan berbagai informasi yang dibutuhkan oleh suatu sistem, seperti data pengguna, data transaksi, data laporan, maupun data lain yang berhubungan dengan proses kerja aplikasi. Menurut Connolly dan Begg, database merupakan kumpulan data yang saling berhubungan dan dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi.

MySQL adalah salah satu sistem manajemen basis data atau Database Management System yang menggunakan bahasa SQL atau Structured Query Language untuk mengelola data. MySQL bersifat open source dan banyak digunakan dalam pengembangan aplikasi web karena memiliki performa yang baik, mudah digunakan, serta dapat berjalan pada berbagai sistem operasi. Dengan MySQL, pengguna dapat membuat database, membuat tabel, menambahkan data, mengubah data, menghapus data, dan mengambil data sesuai kebutuhan sistem.

Dalam MySQL, data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom. Baris menunjukkan satu record data, sedangkan kolom menunjukkan atribut dari data tersebut. Setiap tabel biasanya memiliki primary key sebagai identitas unik untuk membedakan satu data dengan data lainnya. Selain itu, MySQL juga mendukung penggunaan foreign key untuk menghubungkan data antar tabel sehingga relasi data dapat dibuat lebih terstruktur.

MySQL menggunakan perintah SQL untuk melakukan pengelolaan data. Beberapa perintah dasar SQL antara lain CREATE untuk membuat database atau tabel, INSERT untuk menambahkan data, SELECT untuk menampilkan data, UPDATE untuk mengubah data, dan DELETE untuk menghapus data. Selain itu, MySQL juga mendukung penggunaan query yang lebih kompleks seperti JOIN, GROUP BY, ORDER BY, dan WHERE untuk melakukan pencarian dan pengolahan data secara lebih spesifik.

MySQL banyak digunakan pada aplikasi berbasis web karena dapat diintegrasikan dengan berbagai bahasa pemrograman, seperti PHP, Python, Java, dan JavaScript. Dalam pengembangan sistem, MySQL berperan sebagai media

penyimpanan data yang dapat diakses oleh aplikasi sesuai dengan kebutuhan pengguna. Keunggulan MySQL antara lain mudah dipelajari, memiliki dokumentasi yang luas, mendukung banyak pengguna, serta mampu mengelola data dalam jumlah besar dengan baik.

2.2.11 Unified Modeling Language

Menurut Rosa dan Shalahuddin (2021), *Unified Modeling Language* (UML) merupakan bahasa pemodelan visual yang digunakan untuk membantu proses analisis, perancangan, dan dokumentasi sistem perangkat lunak berbasis objek. Pada penelitian ini, UML digunakan untuk membantu proses perancangan aplikasi Monitoring Siswa berbasis web agar alur sistem dapat digambarkan secara lebih jelas dan terstruktur.

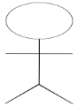
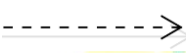
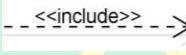
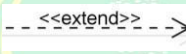
1. Use Case Diagram

Menurut Rosa dan Shalahuddin (2021), *Use Case Diagram* merupakan diagram yang digunakan untuk menggambarkan hubungan antara pengguna (*Aktor*) dengan fungsi-fungsi yang tersedia pada sistem.

Use Case Diagram ini berfungsi untuk memetakan batasan sistem dan interaksi pengguna, sehingga alur fungsionalitas aplikasi dapat dipahami dengan jelas oleh pengembang maupun pengguna. *Use Case Diagram* adalah suatu pemodelan yang menggambarkan kelakuan (*behavior*) yang akan di buat, *Use Case diagram* juga mendeskripsikan interaksi satu dengan lainnya (A.S. & M. Shalahuddin, 2018).

Simbol simbol *Use Case diagram* terlihat pada table 2.2

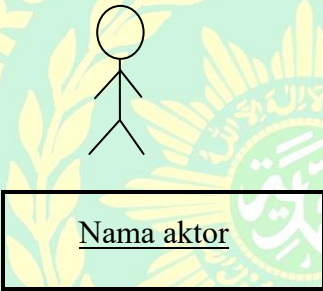
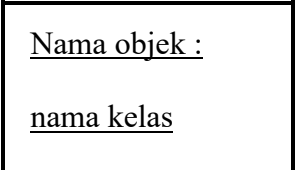
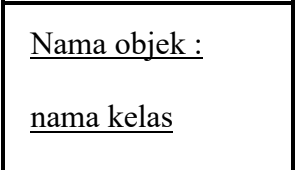
Tabel 2. 2 Keterangan Simbol Use Case Diagram

SIMBOL	DESKRIPSI
Aktor 	Menjelaskan peran user saat berinteraksi dengan <i>Use Case</i>
Dependency 	Mendeskripsikan perubahan yang terjadi pada elemen mandiri dengan elemen yang bergantung
Generalization 	Mendeskripsikan hubungan perilaku objek anak dan <i>structure objek</i> induk
Include 	Spesifikasi <i>Use Case</i> sumber <i>eksplisit</i>
Extend 	Spesifikasi target <i>Use Case</i> memperluas perilaku dari <i>Use Case</i> sumber
Association 	Komunikasi <i>Aktor</i> dan <i>Use Case</i>
Use Case 	Mendeskripsikan aksi yang ditampilkan oleh <i>system</i>

2. Sequence Diagram

Menurut Rosa & Shalahuddin (2021), *Sequence Diagram* merupakan diagram yang menggambarkan interaksi antar objek di dalam sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana pesan dikirim antar objek untuk menjalankan suatu fungsi. Berikut merupakan simbol pada diagram *sequence* terlihat pada table 2.3 (Dewi & Voutama, 2024):

Tabel 2. 3 Keterangan Simbol.Diagram Sequence

No	Simbol	Deskripsi
1	<i>Aktor</i> 	Mendeskripsikan orang atau proses berinteraksi pada system yang di buat diluar <i>system</i>
2.	<i>lifeline</i> 	Mendeskripsikan kehidupan pada suatu objek
3.	Objek 	Merupakan objek yang menyatakan berinteraksi pesan

No	Simbol	Deskripsi
4.	Waktu aktif 	Merupakan objek yang berinteraksi dalam keadaan aktif

		1. <i>Login</i> 2. <i>cekStatusLogin</i> 3. <i>open</i>
5.	Pesan tipe <i>create</i>  <<Create>>	Merupakan objek yang membuat objek lain

3. Activity Diagram

Menurut Rosa dan Shalahuddin (2021), *Activity Diagram* merupakan pemodelan visual yang berfungsi untuk mendeskripsikan rangkaian alur kerja, urutan aktivitas, maupun proses prosedural di dalam suatu sistem. Diagram ini menekankan pada aliran kontrol dari satu aktivitas ke aktivitas lainnya,

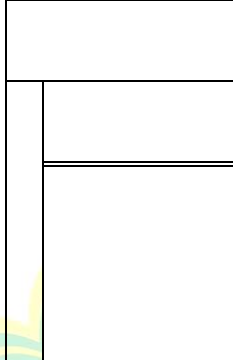
termasuk percabangan keputusan (*decision node*) yang memengaruhi alur proses.

Berikut adalah simbol yang ada pada *activity diagram* terlihat pada table

2.4 (Narulita & Abdillah,2024):

Tabel 2. 4 Keterangan Simbol Diagram Activity

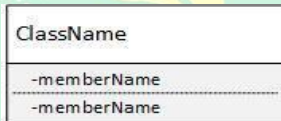
No	Simbol	Deskripsi
1	Status awal 	Merupakan aktivitas awal yang dilakukan oleh <i>system</i>
2	Aktivitas 	Merupakan aktivitas yang dilakukan oleh sistem
3	Percabangan 	Menyatakan aktivitas lebih dari satu
4.	Penggabungan 	Menyatakan menggabungkan aktivitas ke yang lain atau lebih dari satu
5.	Selesai 	Menyatakan selesai nya aktivitas <i>system</i>

No	Simbol	Deskripsi
6.	<i>Swimlane</i>	Memisahkan organisasi bisnis pada aktivitas yang terjadi
		

4. Class Diagram

Menurut Rosa & Shalahuddin (2021), *Class Diagram* merupakan diagram yang digunakan untuk menggambarkan struktur *class* dalam sistem beserta hubungan antar *class* tersebut. Berikut merupakan simbol pada class diagram terlihat pada table 2.5 (Pangestu & Voutama, 2024)

Tabel 2. 5 Keterangan Simbol Diagram Class

No	Simbol	Deskripsi
1.	Kelas	Kelas adalah struktur <i>system</i>
		
2.	Antarmuka	Perancangan pada pemrograman berorientasi objek
		
3.	Asosiasi	Merupakan relasi kelas
		

No	Simbol	Deskripsi
4.	Asosiasi berarah	Merupakan relasi kelas satu dengan yang lainnya
5.	Generalisasi	Merupakan relasi kelas umum
6.	Kebergantungan	Relasi dengan kebergantungan
7.	Agregasi	Relasi dengan semua bagian

2.2.12 Blackbox Testing

Black Box Testing adalah metode pengujian perangkat lunak yang dilakukan dengan cara menguji fungsi sistem berdasarkan input dan output tanpa melihat struktur kode program di dalamnya. Pengujian ini berfokus pada apakah sistem dapat berjalan sesuai dengan kebutuhan pengguna atau tidak. Dalam metode ini,

penguji hanya memberikan data masukan ke dalam sistem, kemudian mengamati apakah hasil keluaran yang dihasilkan sudah sesuai dengan hasil yang diharapkan.

Menurut Pressman, Black Box Testing merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini bertujuan untuk menemukan kesalahan pada fungsi sistem, kesalahan antarmuka, kesalahan pada struktur data, kesalahan akses database, serta kesalahan pada proses input dan output. Dengan menggunakan metode ini, penguji dapat mengetahui apakah setiap fitur dalam sistem telah bekerja sesuai dengan rancangan yang telah ditentukan.

Kelebihan Black Box Testing adalah pengujian dapat dilakukan tanpa harus memahami kode program secara detail. Metode ini juga sesuai digunakan untuk menguji sistem dari sudut pandang pengguna, karena hasil pengujian didasarkan pada fungsi yang terlihat pada antarmuka aplikasi. Namun, kelemahannya adalah pengujian ini tidak dapat mengetahui kesalahan yang terjadi di dalam struktur kode program, sehingga biasanya perlu dilengkapi dengan metode pengujian lain apabila diperlukan.