

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Teori dasar dalam penelitian merujuk pada landasan konseptual yang menjadi acuan untuk memahami, menganalisis, dan menjelaskan fenomena yang diteliti. Teori ini memberikan kerangka kerja ilmiah yang digunakan untuk mengidentifikasi hubungan antara variabel, memformulasikan hipotesis, dan menjawab rumusan masalah. Dalam penelitian, teori dasar berfungsi sebagai pijakan untuk mengembangkan argumen yang mendukung pentingnya penelitian serta relevansi metode yang digunakan. Teori ini dapat bersumber dari berbagai disiplin ilmu, tergantung pada topik penelitian yang diangkat, dan menjadi alat untuk menginterpretasikan hasil penelitian dengan cara yang logis, sistematis, dan sesuai dengan kerangka ilmiah yang telah ada.

2.1.1 Wajah

Wajah manusia merupakan bagian tubuh yang memiliki keunikan tersendiri sehingga dapat digunakan sebagai identitas untuk membedakan individu satu dengan lainnya. Keunikan ini disebabkan oleh kombinasi elemen morfologi yang khas, seperti bentuk dan ukuran mata, hidung, bibir, rahang, serta tekstur kulit. Elemen-elemen ini dipengaruhi oleh faktor genetik yang membentuk struktur dasar wajah, serta faktor lingkungan yang dapat menciptakan variasi tambahan, seperti pola kerutan, bekas luka, atau perubahan akibat usia. Keberagaman ini menjadikan

wajah sebagai salah satu elemen biometrik yang paling efektif dalam mengidentifikasi individu (Dang et al., 2023).

Proses pengenalan wajah bekerja dengan menganalisis pola-pola unik pada wajah, seperti jarak antar fitur, bentuk lekukan, dan proporsi wajah secara keseluruhan. Algoritme pengenalan wajah modern juga memanfaatkan teknologi kecerdasan buatan dan pemrosesan citra untuk mengidentifikasi ciri-ciri wajah dalam bentuk data digital. Data ini kemudian dibandingkan dengan data dalam basis data untuk menemukan kecocokan. Keakuratan pengenalan wajah semakin tinggi berkat kemampuan algoritme untuk mengenali pola yang sangat kompleks, bahkan di bawah kondisi pencahayaan atau posisi wajah yang berbeda.

Selain elemen morfologi statis, wajah juga memiliki fitur dinamis, seperti ekspresi wajah, yang dapat menambah dimensi dalam membedakan individu. Meskipun ekspresi bersifat sementara, pergerakan otot wajah mengikuti pola tertentu yang unik pada setiap orang. Teknologi pengenalan wajah dapat memanfaatkan kombinasi antara ciri statis dan dinamis untuk meningkatkan keakuratan identifikasi. Oleh karena itu, wajah tidak hanya mencerminkan identitas visual, tetapi juga memberikan informasi perilaku yang berguna dalam berbagai aplikasi (Pramono et al., 2020).

Keandalan wajah sebagai alat identifikasi juga didukung oleh fakta bahwa sulit bagi individu untuk mengubah atau menyembunyikan ciri-ciri wajah secara permanen. Tidak seperti sidik jari atau tanda tangan, wajah secara alami terlihat dan mudah diakses oleh teknologi, tanpa memerlukan tindakan khusus dari individu yang diidentifikasi. Hal ini menjadikan pengenalan wajah sebagai metode yang

lebih nyaman, terutama dalam konteks pengawasan dan akses kontrol otomatis, seperti sistem keamanan, kehadiran elektronik, dan manajemen identitas (Sriyati et al., 2020).

Dengan perkembangan teknologi pengenalan wajah yang semakin canggih, metode ini kini diterapkan di berbagai bidang, termasuk pendidikan, kesehatan, dan keamanan. Di bidang pendidikan, misalnya, teknologi ini digunakan untuk sistem absensi otomatis yang mampu mencatat kehadiran siswa secara efisien dan akurat. Dengan memanfaatkan keunikan wajah sebagai elemen utama, sistem pengenalan wajah tidak hanya mempercepat proses identifikasi, tetapi juga meningkatkan keandalan dalam mencatat data dan memastikan validitasnya. Hal ini menunjukkan bahwa wajah merupakan elemen biometrik yang sangat potensial untuk mendukung inovasi teknologi di berbagai sektor (Jaini et al., 2021).

2.1.2 Pendeteksian Wajah

Proses deteksi wajah memiliki peran krusial sebagai langkah awal yang fundamental sebelum tahapan pengenalan wajah dapat dilaksanakan. Secara umum, pengenalan wajah mencakup dua langkah utama, yaitu mendeteksi keberadaan wajah dan mencocokkannya dengan data yang tersimpan dalam basis data (Mekala et al., 2019). Pentingnya pengenalan wajah otomatis untuk sistem absensi daring terus meningkat seiring dengan pesatnya perkembangan di bidang kecerdasan buatan, pembelajaran mesin, serta jaringan saraf tiruan. Kemajuan dalam teknologi ini telah memicu ketertarikan yang besar di kalangan peneliti, mendorong

eksplorasi lebih lanjut di bidang tersebut (Sulistiyo et al., 2014). Teknologi pengenalan wajah memiliki berbagai fungsi, yang meliputi:

1. Dalam konteks keamanan *smartphone*, pengenalan wajah berfungsi sebagai sistem yang memerlukan pengguna untuk menampilkan seluruh wajah mereka di depan sensor kamera agar gambar dapat diambil dan diperiksa untuk mengidentifikasi apakah mereka adalah pengguna *smartphone* yang sah atau tidak.
2. Dalam konteks pengenalan wajah untuk keperluan absensi, teknologi ini digunakan untuk mengidentifikasi karyawan atau siswa berdasarkan data biometrik mereka, sehingga mereka dapat mengakses data-data yang diperlukan.
3. Sebagai sarana untuk memerangi tindak kriminal, teknologi pengenalan wajah digunakan untuk mengidentifikasi dan menganalisis wajah pelaku kejahatan yang terekam dalam rekaman CCTV.



Gambar 2. 1 Pendeteksian Wajah
Sumber: (Sulistiyo et al., 2014)

2.1.3 Absensi

Absensi merupakan proses pencatatan kehadiran individu dalam suatu kegiatan, baik di lingkungan pendidikan, kerja, maupun organisasi lainnya. Absensi memiliki peran penting dalam memastikan partisipasi individu dalam suatu aktivitas, memantau kedisiplinan, serta menyediakan data yang relevan untuk berbagai keperluan administrasi. Sistem absensi umumnya dirancang untuk mencatat informasi mengenai waktu, tanggal, serta status kehadiran seseorang, yang nantinya dapat digunakan sebagai dasar untuk pengambilan keputusan atau evaluasi (Sugeng & Mulyana, 2022).

Isputrawan & Suriyanti (2023) menyatakan bahwa terdapat berbagai variasi dalam sistem absensi yang dapat dikategorikan berdasarkan tujuan penggunaannya serta manfaat yang dihasilkan. Secara garis besar, sistem absensi dapat dikelompokkan ke dalam dua kategori utama..

1. Absensi manual adalah metode tradisional yang dilakukan secara langsung oleh individu atau pengelola untuk mencatat kehadiran. Proses ini biasanya melibatkan pengisian daftar kehadiran, pemanggilan nama, atau pencatatan secara tertulis. Kelebihan dari absensi manual adalah kesederhanaan penerapannya dan tidak memerlukan perangkat teknologi. Namun, metode ini memiliki beberapa kekurangan, seperti kerentanan terhadap manipulasi data, kesalahan pencatatan, serta kurang efisien dalam pengelolaan data, terutama jika jumlah peserta yang dicatat sangat besar. Oleh karena itu, absensi manual dianggap kurang efektif untuk kebutuhan pengelolaan data dalam skala besar.

2. Absensi non-manual, juga dikenal sebagai absensi digital atau otomatis, merupakan metode pencatatan kehadiran yang memanfaatkan teknologi untuk memproses dan menyimpan data kehadiran secara lebih efisien. Contoh dari absensi non-manual meliputi penggunaan kartu identitas elektronik, perangkat pemindai sidik jari, atau teknologi pengenalan wajah. Sistem ini menawarkan berbagai keunggulan, seperti akurasi tinggi, kecepatan pencatatan, serta kemudahan dalam pengelolaan dan pengolahan data. Selain itu, absensi non-manual juga mampu mengurangi risiko kesalahan manusia dan manipulasi data, sehingga lebih andal dibandingkan metode manual.

2.1.4 Machine Learning

Machine Learning (ML) adalah salah satu cabang dari Kecerdasan Buatan (AI) yang saat ini berkembang dengan pesat, bertujuan untuk memungkinkan sistem mempelajari pola dan menghasilkan keputusan secara otomatis tanpa memerlukan instruksi eksplisit dari pengguna. Penerapan teknologi ini mencakup berbagai bidang, seperti kesehatan, pendidikan, ekonomi, teknologi, dan sosial. ML merupakan disiplin yang berfokus pada pengembangan serta analisis algoritme yang mampu mengolah berbagai jenis data untuk menghasilkan informasi yang bermanfaat tanpa memerlukan pengkodean secara manual. Dengan menggunakan dataset sebagai input, algoritme dalam ML dapat membangun model atau aturan yang merepresentasikan pola data tersebut. Contoh implementasi ML dapat dilihat pada sistem pengenalan tulisan tangan atau filter pesan spam dalam layanan email

(Gonzalez & Woods, 2018). Di dalam *Machine Learning*, terdapat beberapa pendekatan utama yang sering digunakan:

1. Teknik Pembelajaran Terawasi (*supervised learning*)

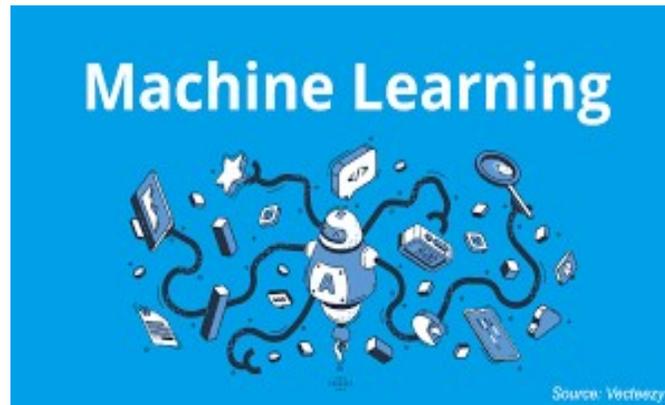
Pendekatan ini menggunakan data yang telah diberi label untuk melatih model sehingga mampu memprediksi hasil tertentu. Algoritme belajar dari data historis dengan pasangan input dan output, sehingga mampu membuat keputusan berdasarkan pola yang telah dipelajari sebelumnya.

2. Teknik Pembelajaran Tidak Terawasi (*unsupervised learning*)

Metodologi ini diterapkan pada dataset yang tidak memiliki label atau anotasi. Tujuannya adalah untuk mengidentifikasi pola tersembunyi, struktur, atau hubungan antar data tanpa panduan eksplisit. Teknik ini sering digunakan dalam pengelompokan data (*clustering*) dan analisis eksploratif.

3. Pembelajaran Penguatan (*Reinforcement Learning*)

Reinforcement Learning adalah pendekatan yang memfokuskan pada pembelajaran melalui interaksi dengan lingkungan. Sistem belajar berdasarkan umpan balik berupa penghargaan (*reward*) atau hukuman (*penalty*) untuk setiap tindakan yang diambil. Tujuannya adalah untuk memaksimalkan akumulasi penghargaan jangka panjang melalui eksplorasi dan eksploitasi keputusan. Metode ini banyak digunakan dalam robotika, permainan komputer, dan pengendalian sistem otonom.



Gambar 2. 2 *Machine Learning*
Sumber: (Gonzalez & Woods, 2018)

2.1.5 Pengolahan Citra Digital

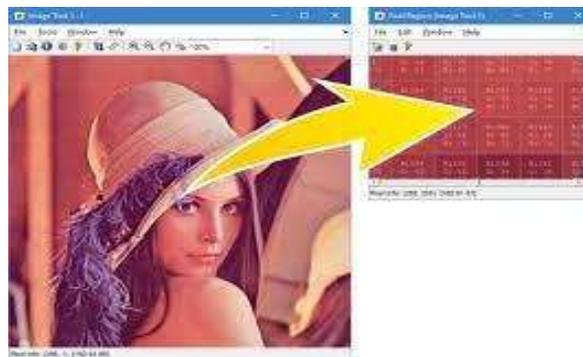
Pengolahan citra digital adalah cabang ilmu komputer yang berfokus pada manipulasi dan analisis gambar digital untuk memperoleh informasi, meningkatkan kualitas visual, atau menghasilkan representasi data yang bermanfaat. Dalam proses ini, gambar digital direpresentasikan sebagai matriks yang terdiri atas elemen-elemen gambar atau piksel yang tersusun dalam baris dan kolom. Setiap piksel memiliki nilai tingkat keabuan atau intensitas tertentu yang merepresentasikan cahaya pada titik tersebut. Nilai intensitas ini dapat diidentifikasi menggunakan koordinat khusus dalam matriks, yaitu (x, y) , di mana setiap titik memiliki nilai intensitas yang diwakili oleh fungsi tertentu (Gonzalez & Woods, 2018).

Tahapan dalam pengolahan citra digital melibatkan beberapa proses, seperti akuisisi citra, yang dilakukan untuk menangkap gambar menggunakan perangkat seperti kamera atau sensor digital; praproses citra, yang mencakup peningkatan kualitas gambar melalui penghilangan noise atau penyesuaian kontras; serta segmentasi citra, yang memisahkan objek dari latar belakang untuk dianalisis lebih

lanjut. Selanjutnya, ekstraksi fitur digunakan untuk memperoleh informasi tertentu dari gambar, seperti pola, bentuk, atau warna, yang sering dimanfaatkan dalam aplikasi pengenalan pola dan objek.

Teknologi pengolahan citra digital telah diterapkan secara luas di berbagai bidang. Dalam dunia medis, pengolahan citra digunakan untuk analisis radiologi dan pencitraan diagnostik. Di sektor keamanan, teknologi ini memainkan peran penting dalam pengenalan wajah dan sistem pengawasan. Di industri, pengolahan citra diterapkan untuk mengontrol kualitas produk. Dengan integrasi teknologi kecerdasan buatan dan pembelajaran mesin, pengolahan citra digital menjadi lebih canggih, memungkinkan analisis data visual dengan tingkat akurasi yang tinggi.

Keunggulan utama dari pengolahan citra digital adalah kemampuannya untuk memproses data visual dengan presisi tinggi serta fleksibilitas untuk diterapkan di berbagai sektor. Seiring dengan perkembangan algoritme dan perangkat keras, teknologi ini terus menjadi salah satu fondasi penting dalam inovasi di era modern.



Gambar 2. 3 *Digital Image Processing*
Sumber: (Gonzalez & Woods, 2018)

2.2 *Software Pendukung*

Software pendukung merujuk pada perangkat lunak yang digunakan oleh peneliti sebagai alat untuk mempermudah dan mempercepat proses penelitian. Perangkat lunak ini memungkinkan periset untuk menjalankan berbagai tugas, seperti pemrosesan data, simulasi, dan pengembangan sistem, dengan lebih efisien. Dalam konteks penelitian ini, beberapa perangkat lunak utama yang digunakan meliputi:

2.2.1 Bahasa Pemrograman *Python*

Python adalah salah satu bahasa pemrograman tingkat tinggi yang dirancang dengan sintaksis yang sederhana, mudah dipahami, dan fleksibel, sehingga cocok untuk digunakan oleh pemula maupun pengembang yang berpengalaman. Bahasa ini pertama kali diperkenalkan oleh Guido van Rossum pada tahun 1991 dan sejak itu terus berkembang menjadi salah satu bahasa pemrograman yang paling populer di dunia. Python mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, imperatif, dan fungsional, yang membuatnya serbaguna untuk berbagai kebutuhan pengembangan perangkat lunak. (Hidayat et al., 2018).

Python menjadi pilihan utama dalam pengembangan *Machine Learning* (ML) dan *Artificial Intelligence* (AI) dikarenakan beberapa alasan utama yaitu (Susim & Darujati, 2021a):

1. Ekosistem Pustaka dan Kerangka Kerja yang Kaya

Python memiliki banyak pustaka dan framework yang dirancang khusus untuk pengembangan ML dan AI, seperti TensorFlow, PyTorch, Scikit-learn, Keras, dan Pandas. Pustaka-pustaka ini menyediakan fungsi dan algoritme yang siap digunakan, sehingga pengembang tidak perlu membangun semuanya dari awal. Selain itu, pustaka seperti NumPy dan SciPy mendukung komputasi numerik dan pemrosesan data secara efisien, sedangkan Matplotlib dan Seaborn mempermudah visualisasi data.

2. Komunitas yang Aktif dan Dukungan yang Kuat

Python memiliki komunitas pengembang yang sangat besar dan aktif. Komunitas ini menyediakan dokumentasi, tutorial, dan forum diskusi yang membantu pengembang dalam memecahkan masalah teknis. Dukungan ini sangat penting, terutama bagi peneliti dan pengembang yang baru mulai bekerja dengan ML dan AI.

3. Kemudahan Penggunaan dan Pembelajaran

Sintaksis Python yang sederhana dan mirip dengan bahasa alami membuatnya lebih mudah dipelajari dibandingkan bahasa pemrograman lainnya. Hal ini memungkinkan pengembang untuk fokus pada logika dan algoritme daripada terjebak dalam kerumitan sintaksis. Python juga memungkinkan pengembangan prototipe yang cepat, yang sangat berguna dalam eksperimen dan penelitian AI.

4. Portabilitas dan Integrasi yang Baik

Python bersifat lintas *platform*, artinya kode yang ditulis dalam Python dapat dijalankan pada berbagai sistem operasi tanpa banyak modifikasi. Selain itu,

Python dapat diintegrasikan dengan bahasa lain, seperti C, C++, dan Java, untuk meningkatkan kinerja atau memanfaatkan fungsi-fungsi spesifik dari bahasa lain.

5. Ketersediaan Platform Pengembangan Berbasis *Cloud*

Banyak *platform cloud* seperti Google Colab, AWS SageMaker, dan Microsoft Azure mendukung pengembangan berbasis Python untuk ML dan AI. *Platform* ini memungkinkan pengembang untuk menjalankan kode Python pada perangkat keras canggih seperti GPU tanpa perlu pengaturan yang rumit.



Gambar 2. 4 Python
Sumber: (Hidayat et al., 2018).

2.2.2 Visual Studio

Visual Studio adalah *Integrated Development Environment* (IDE) yang dikembangkan oleh Microsoft untuk mendukung pengembangan aplikasi perangkat lunak secara menyeluruh. IDE ini menawarkan alat dan fitur lengkap untuk menulis, mengedit, menganalisis, dan melakukan debugging kode. Visual Studio mendukung berbagai bahasa pemrograman, seperti C++, C#, Python, JavaScript, dan Visual Basic, serta dirancang untuk pengembangan aplikasi desktop, web,

cloud, hingga aplikasi lintas platform. Salah satu keunggulan utama Visual Studio adalah integrasi yang kuat dengan ekosistem Microsoft, seperti Azure, .NET *Framework*, dan SQL Server, yang membuatnya menjadi pilihan populer di kalangan pengembang profesional.

Dengan antarmuka yang intuitif dan kemampuan *debugging* yang canggih, Visual Studio memungkinkan pengembang untuk memonitor proses eksekusi kode, menemukan bug, dan memperbaiki masalah dengan lebih efisien. Fitur seperti IntelliSense menyediakan saran penulisan kode secara otomatis, membantu pengguna menulis kode lebih cepat dan akurat. Selain itu, Visual Studio juga memiliki kemampuan untuk mengelola proyek skala besar dengan alat seperti pengelolaan kontrol versi melalui Git dan integrasi langsung dengan layanan CI/CD (*Continuous Integration/Continuous Deployment*).



Gambar 2. 5 Perangkat Lunak Visual Studio
Sumber: (Susim & Darujati, 2021a)

2.2.3 StarUML

StarUML adalah sebuah alat pemodelan perangkat lunak berbasis *Unified Modeling Language* (UML) yang dirancang untuk mendukung pengembangan perangkat lunak dengan pendekatan berorientasi objek. Alat ini digunakan oleh

pengembang dan perancang sistem untuk membuat diagram yang merepresentasikan struktur dan perilaku sebuah sistem secara visual. Dengan StarUML, pengguna dapat memodelkan berbagai aspek dalam siklus hidup pengembangan perangkat lunak, seperti analisis kebutuhan, desain sistem, dan dokumentasi teknis. StarUML mendukung berbagai standar UML, seperti Class Diagram, Sequence Diagram, Use Case Diagram, Activity Diagram, dan lain-lain, yang menjadikannya sangat fleksibel untuk berbagai jenis proyek (Hidayat et al., 2018).

Salah satu keunggulan utama StarUML adalah antarmuka yang ramah pengguna dan mendukung berbagai fitur yang mempermudah proses pemodelan. StarUML mendukung penyesuaian model dengan menyediakan template dan kemampuan untuk menambahkan elemen-elemen tambahan sesuai kebutuhan. Selain itu, alat ini mendukung integrasi dengan teknologi terkini, seperti ekspor model ke berbagai format dokumen dan integrasi dengan alat lain untuk mendukung pengembangan perangkat lunak secara kolaboratif.

StarUML juga dirancang untuk kompatibilitas lintas platform, sehingga dapat digunakan di sistem operasi Windows, macOS, dan Linux. Aplikasi ini mendukung pengembangan berbasis bahasa pemrograman yang beragam dengan fitur seperti generasi kode otomatis (code generation) dari diagram UML. Fitur ini memungkinkan pengembang untuk mempercepat proses implementasi sistem dengan langsung menghasilkan kerangka kode dari model yang dibuat.



Gambar 2. 6 StarUML
Sumber: (Susim & Darujati, 2021a)

Di bawah ini ialah beberapa atribut yang tersedia di *StarUML*:

1. Diagram *use case* (*Usecase Diagram*)

Diagram *use case* merupakan salah satu elemen penting dalam proses analisis dan perancangan perangkat lunak. Diagram ini berfungsi untuk memberikan gambaran visual yang jelas mengenai cara pengguna atau aktor berinteraksi dengan sistem yang akan dikembangkan. Sebagai alat grafis, diagram *use case* sangat efektif dalam menyajikan informasi secara ringkas dan mudah dipahami. Dengan menggunakan simbol-simbol standar, diagram ini memfasilitasi komunikasi yang lebih baik antara pengembang perangkat lunak dan pemangku kepentingan lainnya, sehingga mendukung kolaborasi yang lebih efisien dalam proses perancangan sistem. Dalam diagram ini, aktor berperan sebagai individu yang berinteraksi langsung dengan sistem, yang umumnya merupakan pengguna utama. Interaksi antara berbagai elemen sistem diilustrasikan melalui asosiasi, yang mencerminkan hubungan atau komunikasi antar komponen yang ada. Selain itu, konsep ekstensi digunakan untuk menunjukkan maksud atau tujuan yang hendak dicapai oleh sistem dalam kondisi operasional tertentu. Generalisasi dalam diagram ini

menggambarkan elemen yang memiliki perbedaan makna, namun tetap memiliki hubungan tertentu dalam struktur sistem. Sementara itu, inklusi menyoroti perilaku atau fungsi penting yang wajib ada dalam sistem agar dapat beroperasi sesuai kebutuhan.

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan alur kerja atau proses dalam sebuah sistem. Diagram ini menggambarkan urutan aktivitas, tindakan, atau operasi yang terjadi, baik dalam perangkat lunak maupun proses bisnis. Setiap aktivitas direpresentasikan dengan simbol elips, aliran kerja ditunjukkan dengan panah, dan keputusan atau percabangan digambarkan menggunakan simbol berlian. *Activity diagram* membantu memvisualisasikan alur proses secara menyeluruh, mulai dari langkah awal hingga penyelesaian, sehingga memudahkan analisis dan pemahaman proses. Setiap aktivitas dimulai dengan simbol yang merepresentasikan titik awal, menandakan permulaan suatu proses atau kejadian dalam sistem. Selama proses berlangsung, terdapat aktivitas-aktivitas yang saling bergantung, menunjukkan keterkaitan antara satu langkah dengan langkah berikutnya. Selain itu, diagram ini juga mencakup keputusan, yaitu eksekusi dapat bercabang berdasarkan kondisi atau kriteria tertentu yang telah ditentukan.

3. Diagram Urutan (*Sequence Diagram*)

Diagram sekuensial digunakan untuk menggambarkan interaksi antara berbagai entitas dalam sistem secara berurutan. Aktor dalam diagram ini mewakili individu yang berperan dalam komunikasi atau interaksi dengan sistem, biasanya

merupakan pengguna utama. Elemen *lifeline* berfungsi sebagai penghubung antara aktor dengan kasus penggunaan dalam sistem, menggambarkan eksistensi suatu objek selama proses berlangsung. Dalam komunikasi antara dua entitas, objek berperan sebagai media yang memungkinkan pertukaran informasi atau perintah. Selanjutnya, konsep *uptime* menyoroti keberlanjutan operasi dalam sistem, memastikan hubungan antar-entitas tetap berjalan dengan baik. Selain itu, terdapat mekanisme di mana suatu entitas dapat menerima pernyataan atau pesan yang kemudian diteruskan ke entitas lain untuk memproses instruksi lebih lanjut.

4. Diagram Kelas (*Class Diagram*)

Diagram kelas digunakan untuk merepresentasikan struktur sistem dalam pemrograman berbasis objek. Elemen utama dalam diagram ini adalah kelas, yang menggambarkan susunan entitas yang terdapat dalam sistem. Antarmuka (*interface*) berfungsi sebagai mekanisme utama dalam pemrograman berbasis objek, memungkinkan komunikasi antar kelas tanpa harus bergantung pada implementasi spesifik. Asosiasi menunjukkan atribut yang mencerminkan kesamaan atau kesetaraan antar elemen dalam sistem. Selain itu, asosiasi terarah (*directed associations*) digunakan untuk menegaskan hubungan antara dua elemen dengan multiplikasi yang memberikan makna lebih spesifik.

Konsep generalisasi menggambarkan hubungan dalam sistem yang dapat berlaku dalam konteks yang lebih spesifik maupun lebih umum. Dependensi menunjukkan keterkaitan antar kelas, di mana satu kelas dapat bergantung pada kelas lainnya dalam menjalankan fungsinya. Sementara itu, agregasi merepresentasikan hubungan antara satu elemen dengan kumpulan elemen lainnya yang membentuk suatu kesatuan dalam sistem.

2.2.4 Library OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah pustaka *open-source* yang dirancang untuk aplikasi pemrosesan citra digital dan visi komputer. Pustaka ini menawarkan berbagai fungsi dan algoritma canggih yang mempermudah pengolahan gambar dan video, seperti deteksi wajah, pengenalan objek, analisis gerakan, serta manipulasi citra. OpenCV mendukung berbagai bahasa pemrograman seperti Python, C++, dan Java, sehingga fleksibel untuk digunakan di berbagai platform. Salah satu alasan utama OpenCV populer adalah efisiensinya dalam menangani operasi berbasis citra dan dukungannya terhadap akselerasi perangkat keras untuk meningkatkan kinerja (Hidayat et al., 2018).

Keunggulan OpenCV meliputi sifatnya yang gratis dan *open-source*, dukungan *multiplatform*, dan ketersediaan lebih dari 2500 fungsi yang mencakup berbagai kebutuhan pemrosesan citra, seperti peningkatan kualitas gambar, analisis video, dan pengenalan objek. Selain itu, OpenCV dirancang untuk efisiensi tinggi, sehingga mampu mendukung aplikasi pemrosesan citra secara real-time, seperti sistem pengawasan keamanan atau kendaraan otonom. Pustaka ini juga dapat diintegrasikan dengan teknologi lain, seperti TensorFlow atau PyTorch, memungkinkan pengguna untuk menggabungkan fungsi pemrosesan citra dengan kecerdasan buatan atau pembelajaran mesin

OpenCV telah diterapkan secara luas di berbagai bidang, termasuk keamanan dan pengawasan, teknologi ini digunakan untuk deteksi wajah dan pelacakan objek. Dalam bidang kesehatan, OpenCV mendukung analisis gambar medis, seperti

deteksi tumor atau pembacaan pencitraan diagnostik. Di industri otomotif, pustaka ini memainkan peran penting dalam pengembangan sistem kendaraan otonom dan pengenalan rambu lalu lintas. Selain itu, OpenCV digunakan dalam e-commerce untuk pengenalan gambar produk, serta dalam robotika untuk navigasi dan pengenalan objek. Dengan fleksibilitas dan kapabilitasnya yang luas, OpenCV menjadi salah satu pustaka utama dalam pengembangan aplikasi berbasis visi komputer dan pemrosesan citra (Susim & Darujati, 2021).



Gambar 2. 7 Open CV
Sumber: (Susim & Darujati, 2021a)

2.3 Metode *Eigenface*

Metode Eigenface adalah salah satu pendekatan yang paling dikenal dalam sistem pengenalan wajah berbasis visi komputer. Dikembangkan oleh Turk dan Pentland pada tahun 1991, metode ini memanfaatkan prinsip analisis komponen utama atau *Principal Component Analysis* (PCA) untuk mengurangi dimensi data dan mengekstraksi fitur utama dari gambar wajah. Dalam metode ini, setiap wajah direpresentasikan sebagai kombinasi linier dari sejumlah kecil vektor basis, yang disebut eigenfaces. Eigenfaces merupakan vektor eigen yang dihasilkan dari dekomposisi matriks kovarians kumpulan gambar wajah (Hidayat et al., 2018).

Pendekatan ini bekerja dengan cara mengonversi gambar wajah ke dalam ruang vektor berdimensi rendah, di mana setiap dimensi mewakili karakteristik utama wajah tersebut. Gambar wajah yang direpresentasikan dalam ruang eigen ini kemudian dapat dibandingkan berdasarkan kedekatan jarak vektornya untuk mengidentifikasi atau memverifikasi identitas individu. Salah satu keunggulan metode Eigenface adalah kemampuannya untuk mengurangi ukuran data secara signifikan tanpa kehilangan informasi penting yang diperlukan untuk pengenalan wajah.

Dalam prosesnya, metode Eigenface terdiri dari beberapa langkah utama. Pertama, kumpulan gambar wajah dilatih untuk membentuk basis eigen dengan menghitung nilai rata-rata dari semua gambar wajah dan menormalisasinya. Selanjutnya, matriks kovarians dihitung dari perbedaan setiap gambar terhadap rata-rata, dan eigenvalue serta eigenvector diekstraksi dari matriks tersebut. Eigenvector yang memiliki eigenvalue terbesar digunakan untuk membentuk ruang eigen (eigenspace). Gambar wajah baru yang ingin dikenali kemudian diproyeksikan ke dalam eigenspace, dan jarak antara vektor gambar baru dengan basis eigen dalam ruang tersebut dihitung untuk mengidentifikasi atau memverifikasi identitasnya.

Metode Eigenface memiliki beberapa keunggulan, seperti efisiensi dalam komputasi, karena hanya memanfaatkan fitur utama dari gambar wajah. Selain itu, metode ini relatif sederhana dan mudah diimplementasikan dibandingkan metode lain dalam pengenalan wajah. Namun, Eigenface memiliki keterbatasan, terutama dalam menangani perubahan pencahayaan, ekspresi wajah, dan orientasi wajah

yang dapat memengaruhi akurasi sistem. Meskipun demikian, metode ini tetap menjadi dasar bagi banyak penelitian dan pengembangan sistem pengenalan wajah modern (Susim & Darujati, 2021).



Gambar 2. 8 Metode Eigenface
Sumber: (Hidayat et al., 2018)

Berikut adalah langkah-langkah metode *eigenface* yang digunakan dalam penelitian ini untuk melakukan pengenalan wajah..

1. Pengambilan Gambar Wajah

```
import cv2

# Ambil gambar wajah dari kamera

cap = cv2. VideoCapture (0)

ret, frame = cap. read ()

cv2. imwrite( 'wajah.jpg', frame)

cap. release ()

cv2.destroyAllWindows ()
```

- Mengumpulkan dataset gambar wajah dengan variasi pencahayaan, ekspresi, dan sudut pandang.

- Mengkonversi gambar ke format *grayscale*.

2. Pengolahan Gambar

```
import cv2

# Baca gambar wajah

img = cv2.imread ('wajah.jpg')

# Ubah gambar menjadi grayscale

gray = cv2.cvtColor(img,
CV2.COLOR_BR2GRAY)

# Simpan gambar grayscale

cv2.imwrite('wajah_gray.jpg',
gray)
```

- Mengcrop gambar wajah untuk menghilangkan latar belakang.
- Mengubah ukuran gambar ke ukuran yang seragam
- Mengaplikasikan filter untuk mengurangi noise

3. Ekstraksi Fitur Wajah

```
import cv2

import numpy as np

# Baca gambar wajah grayscale

img = cv2.imread ('wajah_gray.jpg')

# Deteksi wajah menggunakan Haar

Cascade face_cascade =
cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
```

```

faces =
face_cascade.detectMultiScale(img, scaleFactor=1.1,
minNeighbors=5)
# Ekstrak fitur wajah
for (x, y, w, h) in faces:
roi = img[y:y+h, x:x+w]
roi = cv2.resize(roi, (100,
100))
roi = roi.flatten()
print(roi)

```

- Mengubah gambar wajah menjadi vektor kolom
- Membuat matriks data vektor-vektor kolom tersebut

4. Pembuatan *Eigenface*

```

import numpy as np
# Baca fitur wajah yang telah
diekstrak
fitur_wajah = np.loadtxt('fitur_wajah.txt')
# Hitung matriks kovarians
cov = np.cov(fitur_wajah.T)
# Hitung nilai eigen dan vektor
eigen
eigen_values, eigen_vectors = np.linalg.eig(cov)
# Buat matriks eigenface

```

```
eigenface = np.dot (fitur_wajah,
eigen_vectors)
print (eigenface)
```

- menghitung *eigenface* dengan mengalikan matriks transformasi dengan matriks data.
- Eigenface merupakan representasi wajah dalam bentuk vektor.

5. Pengenalan Wajah

```
import numpy as np
# Baca matriks eigenface yang
telah dibuat elgen face = np. loadtxt( 'eigenface.txt')
# Baca fitur wajah yang ingin
dikenali
fitur_wajah_baru = np. loadtxt( 'fitur_wajah_baru.txt')
# Hitung nilai kesamaan antara
fitur wajah baru dan matriks eigenface kesamaan =
np.dot(fitur_wajah_baru, eigenface) print (kesamaan)
```

- Menghitung jarak antara *egenface* input dengan *eiganface* yang sudah ada di database.
- Menggunakan metode jarak seperti euclidean atau mahalanovis.

Mengidentifikasi wajah dengan jarak terkecil.

2.4 Penelitian Terdahulu

Penelitian tentang perancangan sistem absensi otomatis di SMK Tunas Muda Berkarya dengan metode *Eigenface* untuk otomatisasi sistem absensi di sekolah tersebut menggunakan beberapa penelitian sebelumnya yang dijadikan sebagai dasar pengembangan, penelitian-penelitiannya adalah sebagai berikut.

1. Menurut penelitian yang dilakukan oleh (Pramono et al., 2020) berjudul "Pembangunan Sistem Presensi Magang Berbasis *Mobile* Menggunakan *Face Recognition*" yang diterbitkan dalam Jurnal Informatika Atma Jogja pada tahun 2020, hasil penelitian menunjukkan bahwa sistem presensi berbasis pengenalan wajah mencapai tingkat akurasi sebesar 95%.
2. Berdasarkan penelitian yang dilakukan oleh Hidayat et al. (2018) dalam artikel berjudul "*Rancang Bangun Sistem Presensi Menggunakan Face Recognition dengan Metode Eigenface*" yang diterbitkan di Jurnal Ilmiah Setrum (P-ISSN 2301-4652, Volume 07, No. 02, Desember 2018), algoritma *Eigenface* digunakan untuk implementasi pengenalan wajah. Algoritma ini memanfaatkan pendekatan *Principal Component Analysis* (PCA) yang pertama kali dikembangkan di Massachusetts Institute of Technology (MIT). Penelitian tersebut memberikan penjelasan mendalam tentang penerapan dan fungsi algoritma *Eigenface* dalam sistem presensi. Metode ini sering dianggap sebagai salah satu pendekatan yang paling sederhana sekaligus efektif dalam pengenalan wajah. Pendekatan *Eigenface* terdiri atas tiga tahap utama, yaitu pengambilan gambar, pelatihan data gambar, dan proses pengenalan wajah.

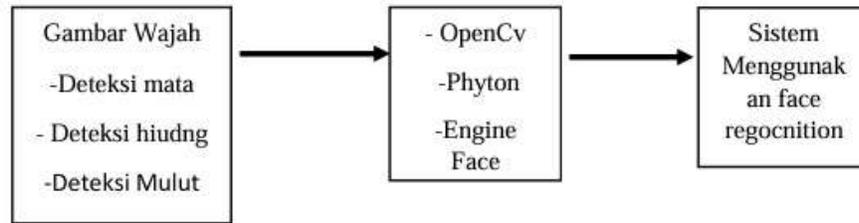
3. Penelitian yang dilakukan oleh Putra et al. (2022) dengan judul "*Perancangan dan Pengembangan Sistem Absensi Online Berbasis Android Menggunakan Pengenalan Wajah*" yang dipublikasikan dalam *Jurnal Sistem Informasi dan Teknologi Islam* (Volume 03, No. 01, Februari 2022, ISSN 2721-0901) menyimpulkan bahwa sistem absensi yang dirancang mampu melakukan proses absensi secara real-time. Sistem ini menggunakan teknologi deteksi wajah berbasis metode pengenalan wajah Eigenface yang didukung oleh pendekatan Principal Component Analysis (PCA).
4. Berdasarkan artikel ilmiah yang ditulis oleh (Susim & Darujati, 2021) berjudul "*Pengolahan Citra untuk Pengenalan Wajah dengan Menggunakan OpenCV*", yang dipublikasikan dalam *Jurnal Syntax Admiration* (Volume 2, Nomor 3, Maret 2021), penelitian tersebut menyoroti tiga faktor utama yang memengaruhi hasil deteksi wajah. Faktor-faktor tersebut meliputi kondisi pencahayaan, orientasi wajah, dan variabel lain yang memengaruhi tingkat akurasi sistem.
5. Menurut Putra (2015) dalam artikel berjudul "*Perancangan dan Pengembangan Sistem Absensi Realtime Melalui Metode Pengenalan Wajah*", yang diterbitkan di *Jurnal Sains dan Teknologi* (Volume 3, Issue 2, ISSN 2303-3142), penelitian ini menunjukkan bahwa sistem absensi yang dirancang berhasil melakukan pencatatan kehadiran secara real-time dengan memanfaatkan teknologi pengenalan wajah. Pengujian yang dilakukan menggunakan metode *blackbox* membuktikan bahwa sistem tersebut dapat berjalan secara efektif dan memenuhi tujuan yang dirancang.

6. Menurut Jaini et al. (2021) dalam artikel berjudul "Sistem Manajemen Kehadiran Menggunakan Metode Pengenalan Wajah Berbasis Web", yang dipublikasikan dalam Jurnal Ilmiah Teknologi Sistem Informasi (Volume 02, Issue 02, ISSN 2722-4619), penelitian ini menyimpulkan bahwa sistem yang dikembangkan mampu secara efektif membedakan mahasiswa yang perlu dicatat kehadirannya dan yang tidak. Selain itu, sistem tersebut juga dapat mengidentifikasi serta membedakan fitur wajah mahasiswa yang telah terdaftar, sehingga mendukung pengelolaan kehadiran dengan lebih akurat dan efisien.
7. Menurut Prayoga dan Simanjuntak (2020) dalam artikel berjudul "*Rancang Bangun Prototipe dan Sistem Parkir Mobile QR Code Berbasis Android dengan Arduino*", yang dipublikasikan dalam *Jurnal Pengembangan Sistem Informasi* (Volume 5, Nomor 2, Juli 2020), penelitian ini menyimpulkan bahwa sistem berbasis Android dirancang menggunakan Android Studio dan memanfaatkan MySQL sebagai basis data untuk menyimpan data. Sistem ini dirancang untuk mendukung pengelolaan parkir secara efisien dengan teknologi QR Code.
8. Menurut Renaldi dan Sitohang (2023) dalam artikel berjudul "*Rancangan Sistem Absensi Siswa Otomatis Berbasis Smart QR Card Menggunakan Algoritma BCRYPT*", yang dipublikasikan dalam *Jurnal COMASIE* (Vol. 9, No. 1, 2023), penelitian ini menyimpulkan bahwa sistem absensi yang dirancang mampu mengatasi berbagai permasalahan dalam proses absensi.

Sistem ini memanfaatkan teknologi Smart QR Card dan algoritma BCRYPT untuk meningkatkan efisiensi dan keamanan dalam pencatatan kehadiran.

2.5 Kerangka Pemikiran

Kerangka pemikiran adalah landasan konseptual yang digunakan untuk menjelaskan hubungan antara berbagai variabel atau elemen yang terlibat dalam sebuah penelitian. Kerangka ini berfungsi sebagai panduan untuk memahami alur logis penelitian, mulai dari identifikasi masalah, analisis kebutuhan, hingga implementasi solusi. Dalam penelitian perancangan sistem absensi berbasis pengenalan wajah, kerangka pemikiran berperan penting untuk menggambarkan bagaimana teknologi pengenalan wajah dapat diterapkan untuk mengatasi keterbatasan pada sistem absensi manual. Kerangka ini menghubungkan identifikasi masalah, seperti ketidakakuratan pencatatan atau waktu yang tidak efisien, dengan solusi berbasis teknologi, seperti algoritma pengenalan wajah yang mampu mendeteksi dan mencatat kehadiran secara otomatis dan real-time. Dengan kerangka pemikiran ini, penelitian dapat menunjukkan tahapan proses, mulai dari pengumpulan data, pengolahan citra wajah, hingga integrasi ke dalam sistem absensi. Hal ini memberikan gambaran yang jelas tentang bagaimana penelitian akan mencapai tujuannya, yakni merancang sistem absensi yang lebih efisien, akurat, dan modern dengan memanfaatkan teknologi berbasis kecerdasan buatan dan visi komputer.



Gambar 2. 9 Kerangka Pemikiran
Sumber: (Data Penelitian, 2024)