

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian pertama dilakukan oleh (Essien & Ansa, 2023) dengan judul "*A Deep Learning-Based Face Recognition Attendance System*". Penelitian ini mengembangkan sistem presensi kelas otomatis yang menggunakan *Multitask Cascaded Convolutional Neural Network* (MTCNN) untuk deteksi dan penyelarasan wajah, serta model *pretrained* FaceNet-512 untuk menghasilkan *embedding* wajah berdimensi 512, dengan *cosine similarity* sebagai metrik pencocokan identitas. Dataset yang digunakan berjumlah tiga foto wajah untuk masing-masing 30 siswa, yang kemudian diperbanyak menjadi sepuluh gambar per siswa melalui augmentasi citra. Data kehadiran disimpan dalam basis data MySQL dan diakses melalui *API* yang dilengkapi mekanisme autentikasi pengguna. Hasil pengujian menunjukkan bahwa sebelum augmentasi diterapkan terdapat satu siswa yang gagal dikenali, sedangkan setelah augmentasi seluruh peserta kelas berhasil dikenali dan tercatat dalam laporan presensi. Kelebihan penelitian ini adalah kemampuannya memproses foto kelas berisi banyak wajah sekaligus dalam satu kali pengambilan gambar, namun pengujian masih terbatas pada satu sesi kelas dan sistem bergantung pada perangkat kamera geser (*sliding camera*) khusus yang dipasang di ruang kelas.

Penelitian kedua dilakukan oleh (Rutuja Bankar dkk, 2022) dengan judul "*Face Recognition Using Facenet Deep Learning Network for Attendance System*". Penelitian ini berfokus pada pengembangan sistem pengenalan wajah menggunakan model *FaceNet* yang berbasis algoritma *Deep Learning*. Sistem ini dirancang untuk mengenali wajah secara real-time dengan menghasilkan *embedding* wajah menggunakan *Triplet Loss Function* untuk memastikan *embedding* dari identitas yang sama lebih dekat dibandingkan dengan identitas yang berbeda. Dataset yang digunakan berjumlah 50 ribu gambar, dan model diuji dalam berbagai kondisi pencahayaan serta pose wajah. Hasil pengujian menunjukkan akurasi sebesar 99.63%. Kelebihan penelitian ini adalah akurasi

tinggi dan kecepatan proses pengenalan wajah, tetapi memerlukan perangkat keras dengan spesifikasi tinggi untuk performa optimal.

Penelitian ketiga dilakukan oleh (Dr M S Shashidhara, 2024) dengan judul *"Attendance System AI Based on Facial Recognition System"*. Penelitian ini menggunakan algoritma *Convolutional Neural Network (CNN)* untuk pengenalan wajah yang dikombinasikan dengan preprocessing berbasis *OpenCV*. Sistem ini dirancang untuk mencatat kehadiran siswa secara otomatis dengan tingkat akurasi sebesar 89%. Dataset yang digunakan terdiri dari ribuan gambar wajah siswa yang diuji dalam berbagai kondisi pencahayaan. Kelebihan penelitian ini adalah penerapan sistem berbasis AI yang mampu mencatat presensi secara real-time, tetapi terbatas pada skenario indoor dengan pencahayaan stabil.

Penelitian keempat dilakukan oleh (Bhargava dkk, 2023) dengan judul *"Integration of FaceNet and DeepFace Algorithms for Optimized Attendance Management in Educational Systems"*. Penelitian ini bertujuan untuk mengatasi keterbatasan sistem presensi manual dan biometrik yang memerlukan partisipasi aktif siswa serta rentan terhadap kecurangan presensi. Penelitian ini mengintegrasikan algoritma *FaceNet* dan *DeepFace* dalam sistem presensi berbasis video surveillance untuk pelacakan kehadiran secara otomatis. *FaceNet* mencapai akurasi sebesar 96%, dengan precision 95%, recall 94%, dan F1 score 94.5%, sementara *DeepFace* memiliki akurasi sedikit lebih rendah yaitu 94%. Hasil penelitian menunjukkan bahwa sistem ini mampu menyederhanakan proses presensi, meningkatkan akurasi, serta mendukung pelacakan kehadiran secara real-time. Kelebihan penelitian ini adalah kemampuannya dalam mendeteksi wajah dalam berbagai kondisi pencahayaan dan pose wajah, sementara tantangannya adalah kebutuhan perangkat keras dengan spesifikasi tinggi untuk performa optimal.

Tabel 2.1 Penelitian Terdahulu

Kategori	Essien & Ansa (2023)	Bankar dkk. (2022)	Shashidhara dkk. (2024)	Bhargava dkk. (2023)
Metode	MTCNN, FaceNet-512, cosine similarity	MTCNN, FaceNet 128D, SVM linear	dlib landmark, CNN ResNet, OpenCV	YOLO, FaceNet, DeepFace

Kategori	Essien & Ansa (2023)	Bankar dkk. (2022)	Shashidhara dkk. (2024)	Bhargava dkk. (2023)
Data	30 siswa, 3 foto/siswa, augmentasi jadi 10	Dataset standar (YALE, JAFFE, AT&T, Essex)	Tidak disebutkan	COCO, LFW, YouTube Faces
Sumber Gambar	Foto kelas, kamera geser	Foto kelas, kamera ruang kelas	Kamera perangkat IoT	Video surveillance
Pengujian	Satu sesi kelas, sebelum vs sesudah augmentasi	Dataset standar, split 80:20, <i>confusion matrix</i>	Satu individu, ruangan terang, sudut -20° – 20°	Dataset <i>benchmark</i> , FaceNet vs DeepFace
Hasil	Seluruh siswa dikenali setelah augmentasi	100% pada lima dataset, terendah 77,67%	Akurasi 99,63%	FaceNet 96%, DeepFace 94%
Keterbatasan	Bergantung kamera geser khusus, satu sesi kelas	Bukan foto kelas nyata, akurasi turun ke 77,67%	Satu individu, kondisi terang, data tidak disebutkan	Bukan pengujian kelas, butuh GPU kelas atas

Berdasarkan kajian terhadap beberapa penelitian terdahulu pada Tabel 2.1, diketahui bahwa penelitian-penelitian tersebut telah membahas topik yang relevan dengan penelitian ini, namun memiliki keterbatasan dari sisi pengujian yang sebagian besar dilakukan pada dataset citra wajah standar dan bukan pada foto kelas nyata, kondisi pengujian yang masih terbatas sehingga pengaruh jarak pengambilan gambar dan variasi pencahayaan belum terukur, serta ketergantungan pada perangkat atau infrastruktur khusus seperti kamera yang dipasang di ruang kelas, *video surveillance*, dan perangkat keras berspesifikasi tinggi.

Penelitian ini memiliki perbedaan dengan penelitian sebelumnya karena mengkaji permasalahan pencatatan kehadiran siswa dengan menyesuaikan pendekatan penelitian terhadap kebutuhan dan karakteristik objek penelitian, yaitu SMK Dharma Loka. Pengambilan citra dilakukan menggunakan kamera *smartphone* yang telah dimiliki guru tanpa memerlukan perangkat tambahan, deteksi wajah menggunakan MTCNN, ekstraksi *embedding* menggunakan FaceNet dengan pendekatan *transfer learning*, serta penentuan identitas dilakukan dengan menggabungkan *Support Vector Machine*, *Logistic Regression*, dan *Cosine Distance* melalui mekanisme *majority voting*. Pengujian juga diperluas dengan mengukur pengaruh jarak pengambilan gambar dan kondisi pencahayaan terhadap

akurasi pengenalan. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi pada penerapan sistem presensi otomatis berbasis *face recognition* di lingkungan sekolah menengah dalam menjawab perumusan masalah serta melengkapi hasil-hasil penelitian yang telah ada sebelumnya.

2.2 Landasan Teori

2.2.1 SMK Dharma Loka

SMK Dharma Loka adalah salah satu sekolah menengah kejuruan yang berlokasi di Jl. Soekarno Hatta, Gang Permata I No.99, Labuh Baru Barat, Kecamatan Payung Sekaki, Kota Pekanbaru, Riau. Sekolah ini didirikan pada 11 September 2000 berdasarkan Surat Keputusan Pendirian Nomor 08304/I09.8.4/MN-2000. Sebagai institusi pendidikan vokasi, SMK Dharma Loka menawarkan program pendidikan yang berorientasi pada keterampilan dan kesiapan kerja. Sekolah ini memiliki beberapa bidang keahlian utama, yaitu Akuntansi dan Keuangan Lembaga, Bisnis Daring dan Pemasaran, serta Multimedia, yang dirancang untuk membekali siswa dengan kompetensi sesuai kebutuhan industri.

Dalam upaya meningkatkan kualitas pendidikan, SMK Dharma Loka telah meraih akreditasi A (Amat Baik) berdasarkan Surat Keputusan Nomor 1453/BAN-SM/SK/2022. Sekolah ini memiliki visi untuk menjadi pusat pendidikan dan pelatihan profesional yang mampu mencetak lulusan yang mandiri, berkualitas, dan siap bersaing di tingkat global. Untuk mewujudkan visi tersebut, SMK Dharma Loka menjalin kerja sama dengan berbagai industri dan dunia usaha guna memberikan pengalaman praktik langsung kepada siswa. Selain itu, sekolah juga mengembangkan program pembelajaran berbasis teknologi untuk memastikan bahwa para siswa mendapatkan pendidikan yang relevan dengan perkembangan zaman.

Sebagai institusi yang berfokus pada pendidikan vokasi, SMK Dharma Loka tidak hanya menekankan pembelajaran di dalam kelas tetapi juga memberikan kesempatan kepada siswanya untuk mengikuti program magang, pelatihan keterampilan, serta kegiatan kewirausahaan. Dengan kombinasi antara teori dan praktik yang kuat, sekolah ini berupaya menciptakan lulusan yang tidak hanya

memiliki kompetensi teknis tetapi juga kemampuan soft skill yang dibutuhkan dalam dunia kerja. Dukungan fasilitas yang memadai serta tenaga pengajar yang berpengalaman menjadikan SMK Dharma Loka sebagai salah satu SMK unggulan di Kota Pekanbaru.



Gambar 2.1 Gedung SMK Dharma Loka
(Sumber: SMK Dharma Loka, 2016)

Visi dan Misi dari SMK Dharma Loka adalah :

Visi :

- 1) Menjadikan SMK Dharma Loka sebagai pusat pendidikan dan pelatihan secara profesional dalam rangka pengembangan sumber daya manusia yang mandiri, berkualitas, dan mampu menghadapi persaingan global
- 2) Terwujudnya lulusan SMK Dharma Loka yang mampu bekerja keras, percaya diri, tanggap lingkungan dan inovatif

Misi :

- 1) Memberikan layanan masyarakat dalam bidang pendidikan dan pelatihan yang profesional, unggul serta kompetitif dalam menghadapi globalisasi
- 2) Menjalin kerjasama dengan dunia usaha dan industri serta praktisi dalam penyediaan tenaga kerja yang terampil dan handal

2.2.2 Sistem Presensi

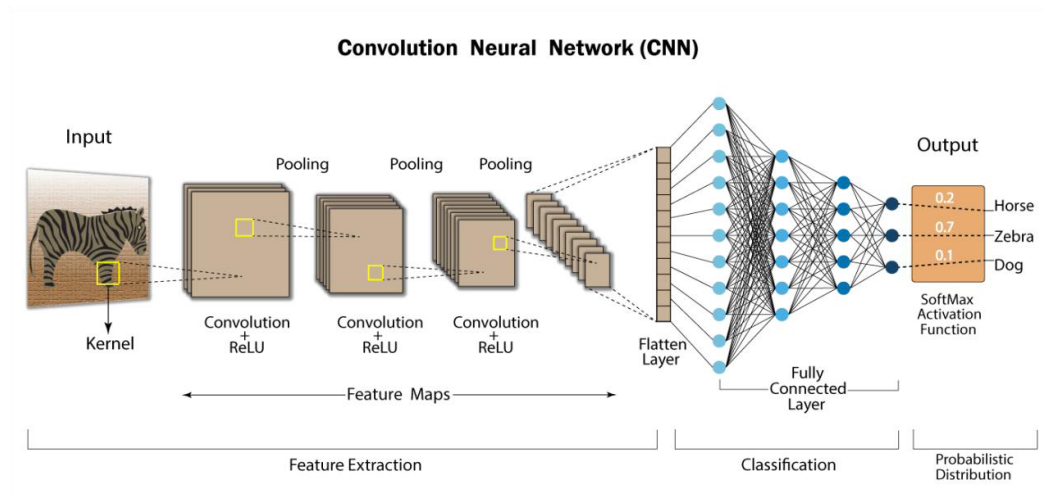
Sistem presensi adalah teknologi yang dirancang untuk mencatat kehadiran individu secara otomatis atau semi-otomatis, dengan tujuan meningkatkan efisiensi, akurasi, dan keamanan dibandingkan metode manual seperti tanda tangan atau absen kertas. Sistem ini memanfaatkan berbagai teknologi, seperti biometrik, pengenalan wajah, RFID, GPS, hingga kode QR. Teknologi tersebut tidak hanya mencatat kehadiran dengan lebih cepat tetapi juga menyediakan data yang lebih transparan dan sulit dimanipulasi (Ngo dkk, 2024).

Salah satu metode yang semakin populer adalah penggunaan *Face Recognition*. Teknologi ini memungkinkan identifikasi individu secara unik tanpa kontak fisik, sehingga lebih higienis dan praktis. Penerapan face recognition dinilai sangat bermanfaat di sektor pendidikan maupun organisasi karena mampu mendeteksi kehadiran dengan akurat dan mencegah kecurangan seperti kehadiran proxy (Rao, 2022).

Penelitian menunjukkan bahwa implementasi teknologi pengenalan wajah dalam sistem presensi dapat meningkatkan efisiensi dan akurasi pencatatan kehadiran. Misalnya, sebuah studi menemukan bahwa metode *Face Recognition* efektif dalam mendeteksi dan mengenali wajah dengan akurasi yang tinggi. Selain itu, integrasi teknologi ini dengan sistem informasi berbasis web atau aplikasi mobile memungkinkan pemantauan kehadiran secara real-time dan mempermudah proses rekapitulasi data, sehingga meningkatkan transparansi dan akuntabilitas dalam manajemen kehadiran (Fauzan & Andi, 2024).

2.2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang sangat populer dan efektif untuk pengolahan data berbasis gambar. CNN terinspirasi oleh cara kerja visual cortex pada otak manusia, di mana neuron merespons secara selektif terhadap bagian tertentu dari suatu rangsangan visual. CNN dirancang untuk bekerja dengan data yang memiliki struktur *grid*, seperti gambar. Keunggulan utama CNN adalah kemampuannya untuk secara otomatis mengekstrak fitur dari data input dengan cara yang sangat efisien dibandingkan dengan jaringan neural biasa (Gu dkk, 2015; O'Shea & Nash, 2015).



Gambar 2.2 Arsitektur *Convolution Neural Netowrk*

(Sumber: Dabass, 2025)

Komponen-komponen utama dalam CNN :

1) Convolutional Layer

Lapisan ini adalah inti dari CNN yang bertugas mengekstraksi fitur-fitur penting dari input, seperti pola, tepi, atau tekstur. Proses ini dilakukan dengan menggunakan filter kecil yang bergerak melintasi gambar untuk menghitung hasil konvolusi. Filter ini bertugas mengenali pola tertentu pada area kecil dari gambar, yang dikenal sebagai *receptive field*. Hasil dari proses ini adalah peta fitur (*feature map*) yang menunjukkan lokasi pola yang dikenali dalam gambar. Dengan mekanisme ini, convolutional layer mengurangi jumlah parameter yang harus dipelajari dibandingkan jaringan neural biasa.

2) Activation Function

Setelah operasi konvolusi, fungsi aktivasi diterapkan untuk memperkenalkan non-linearitas, sehingga model dapat menangkap hubungan yang lebih kompleks. Fungsi aktivasi yang sering digunakan adalah ReLU (Rectified Linear Unit), yang mengubah nilai negatif menjadi nol. Fungsi ini mempercepat proses pelatihan dan membantu jaringan belajar pola yang lebih kompleks.

3) Pooling Layer

Pooling layer digunakan untuk mengurangi ukuran peta fitur sambil mempertahankan informasi penting. Proses ini menurunkan jumlah data yang

perlu diproses, sehingga mengurangi beban komputasi dan risiko overfitting. Dua metode pooling yang umum adalah max pooling (memilih nilai maksimum dalam area tertentu) dan average pooling (mengambil nilai rata-rata). Dengan pooling, model juga menjadi lebih toleran terhadap perubahan posisi kecil dalam gambar.

4) Fully-Connected Layer

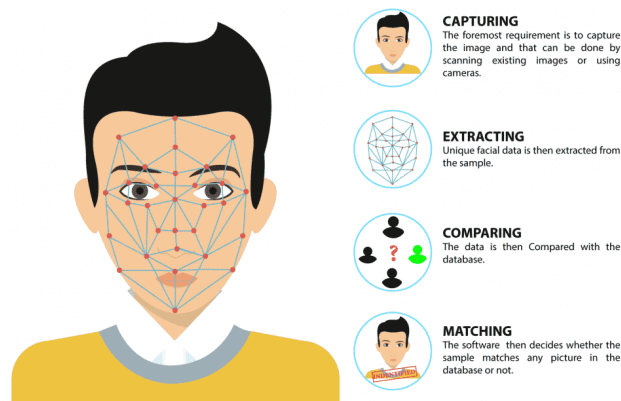
Fully-connected layer bertugas menggabungkan semua fitur yang telah diekstraksi untuk membuat keputusan akhir. Dalam lapisan ini, setiap neuron terhubung ke semua neuron di lapisan sebelumnya. Lapisan ini biasanya digunakan di bagian akhir jaringan untuk menghasilkan output berupa probabilitas kelas atau prediksi lainnya.

5) Output Layer

Lapisan terakhir dari CNN menghasilkan prediksi berdasarkan tugas yang sedang dilakukan. Untuk klasifikasi, biasanya digunakan fungsi softmax untuk menghasilkan probabilitas setiap kelas. Sedangkan untuk tugas regresi, fungsi linear sering digunakan. Lapisan ini mengubah fitur yang telah diringkas menjadi hasil akhir yang dapat diinterpretasikan.

2.2.4 Face Recognition

Face Recognition adalah teknologi biometrik yang digunakan untuk mengidentifikasi atau memverifikasi identitas seseorang dengan menganalisis karakteristik unik wajah mereka. Teknologi ini telah banyak diterapkan dalam berbagai bidang, seperti keamanan, presensi otomatis, dan pengawasan publik. Face Recognition bekerja dengan mendeteksi wajah dalam gambar atau video, mengekstraksi fitur wajah, dan mencocokkan fitur tersebut dengan data yang tersimpan dalam database.



Gambar 2.3 Ilustrasi Cara Kerja *Face Recognition*
(Sumber: Suyogbargule, 2026)

Teknik-teknik yang telah dikembangkan dan dapat digunakan untuk melakukan proses pengenalan wajah:

1) Pengenalan 2D

Menggunakan gambar dua dimensi untuk menganalisis fitur wajah. Metode ini rentan terhadap perubahan pencahayaan, sudut pandang, dan ekspresi wajah.

2) Pengenalan 3D

Memanfaatkan geometri tiga dimensi dari struktur wajah untuk mencapai akurasi lebih tinggi dibandingkan metode 2D. Metode ini mengukur fitur wajah yang kaku dan tidak terpengaruh oleh variasi pencahayaan atau sudut pandang.

3) Berbasis Fitur Lokal

Menangkap karakteristik tertentu dari area wajah, seperti bentuk bibir, sudut mata, atau bentuk hidung. Teknik seperti Local Binary Patterns (LBP) sering digunakan untuk memproses tekstur wajah.

4) Berbasis Template Global

Menggunakan seluruh wajah sebagai satu unit untuk pengenalan. Algoritma seperti Principal Component Analysis (PCA) atau Eigenfaces menganalisis pola keseluruhan wajah.

5) Neural Network

Sistem seperti FaceNet atau DeepFace menggunakan Convolutional Neural Networks (CNN) untuk mengekstraksi dan memetakan gambar wajah ke ruang

Euclidean berdimensi tinggi, menilai kesamaan berdasarkan jarak antara vektor yang dinormalisasi.

6) Pengenalan Berdasarkan Perilaku

Menggabungkan fitur wajah dengan analisis perilaku seperti gerakan kepala, ekspresi wajah, atau gerakan bibir untuk memberikan lapisan keamanan tambahan.

7) Hybrid Recognition

Mengombinasikan berbagai teknik, seperti 2D dan 3D atau fitur lokal dan global, untuk mengatasi kekurangan dari teknik individu dan meningkatkan akurasi.

2.2.5 Multi-task Cascaded Convolutional Networks (MTCNN)

Multi-task Cascaded Convolutional Networks (MTCNN) adalah metode deteksi wajah berbasis *deep learning* yang diperkenalkan oleh Zhang dkk pada tahun 2016. Metode ini menyelesaikan dua tugas sekaligus, yaitu menentukan posisi wajah pada gambar dan menentukan titik penanda wajah, melalui tiga jaringan konvolusi yang bekerja secara bertingkat. Pendekatan bertingkat ini membuat proses deteksi tetap efisien, karena jaringan yang lebih ringan menyaring sebagian besar kandidat terlebih dahulu sehingga jaringan yang lebih berat hanya memproses kandidat yang tersisa (Zhang dkk., 2016).

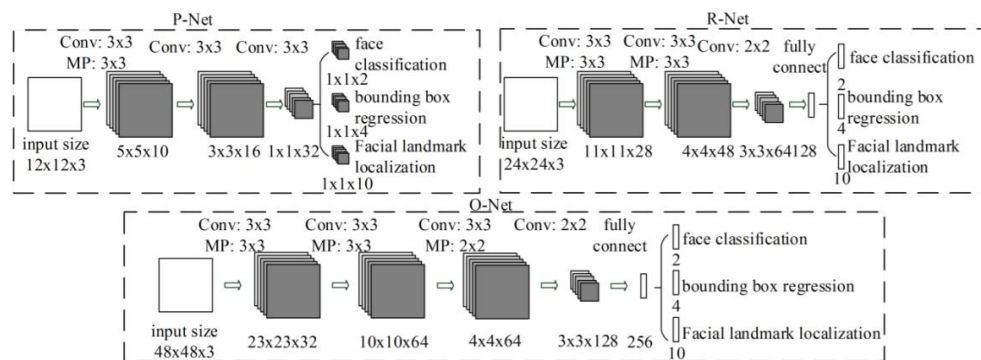
Sebelum masuk ke tahap pertama, gambar masukan terlebih dahulu diubah menjadi beberapa ukuran berbeda yang membentuk piramida citra (*image pyramid*). Mekanisme ini memungkinkan wajah dengan ukuran yang berbeda-beda dalam satu gambar tetap dapat terdeteksi, karena wajah yang berukuran kecil pada gambar asli akan menjadi cukup besar untuk dikenali pada tingkat piramida tertentu.

Tahap pertama dilakukan oleh *Proposal Network* (P-Net) yang menerima masukan berukuran 12×12 piksel. Jaringan ini memindai seluruh tingkat piramida citra dan menghasilkan sejumlah besar kandidat area yang diduga berisi wajah beserta perkiraan kotak pembatasnya. Kandidat yang saling bertumpang tindih kemudian disaring menggunakan *Non-Maximum Suppression* (NMS) agar satu wajah tidak menghasilkan banyak kandidat sekaligus.

Tahap kedua dilakukan oleh *Refine Network* (R-Net) yang menerima masukan berukuran 24×24 piksel. Jaringan ini menyaring kandidat dari tahap sebelumnya dengan membuang kandidat yang bukan wajah, sekaligus memperbaiki posisi kotak pembatas melalui regresi. Karena jumlah kandidat sudah jauh berkurang, jaringan pada tahap ini dapat dibuat lebih dalam tanpa membebani waktu pemrosesan secara berlebihan.

Tahap ketiga dilakukan oleh *Output Network* (O-Net) yang menerima masukan berukuran 48×48 piksel dan merupakan jaringan paling dalam di antara ketiganya. Jaringan ini menghasilkan keluaran akhir berupa kotak pembatas wajah, nilai keyakinan deteksi, serta lima titik penanda wajah. Nilai keyakinan tersebut dapat digunakan sebagai ambang penyaringan sehingga deteksi dengan tingkat keyakinan rendah dapat diabaikan.

Keunggulan utama MTCNN terletak pada kemampuannya mendeteksi wajah dengan ukuran dan posisi yang bervariasi dalam satu gambar, yang diperoleh dari penggunaan piramida citra sehingga proses deteksi tidak terikat pada satu skala tertentu. Karena itu MTCNN banyak digunakan sebagai tahap awal pada sistem pengenalan wajah, khususnya pada gambar yang memuat lebih dari satu wajah dengan jarak yang tidak seragam terhadap kamera (Zhang dkk., 2016).



Gambar 2. 4 Metode MTCNN untuk *Face Recognition*

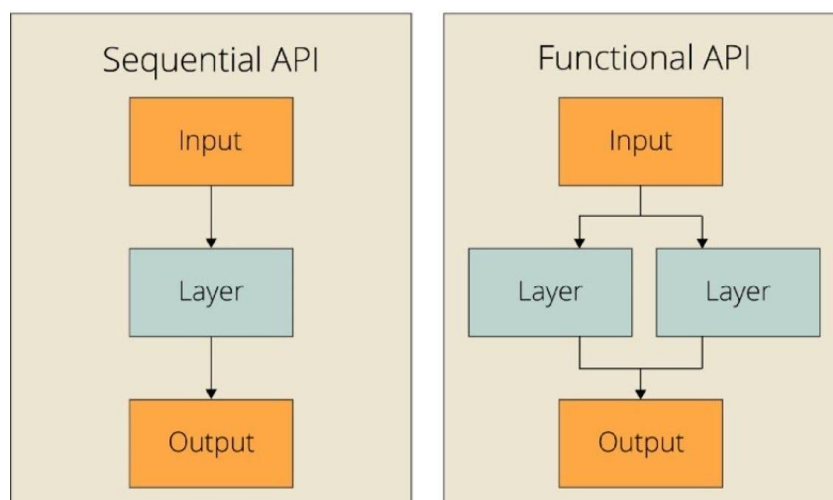
(Sumber: Fuad, 2020)

2.2.6 TensorFlow/Keras

TensorFlow adalah kerangka kerja open-source yang dikembangkan oleh Google Brain untuk membangun dan melatih model pembelajaran mesin dalam skala besar. Dengan menggunakan pendekatan grafik aliran data (*dataflow graph*),

TensorFlow memungkinkan representasi komputasi dalam bentuk graf, di mana setiap simpul mewakili operasi matematika, dan setiap sisi adalah tensor yang mengalir di antara operasi. TensorFlow sangat fleksibel dan mendukung berbagai perangkat keras, termasuk CPU, GPU, dan *Tensor Processing Units* (TPUs). Kemampuannya dalam menangani komputasi terdistribusi menjadikannya pilihan utama untuk tugas-tugas *deep learning*, seperti pengenalan gambar, pemrosesan bahasa alami, dan analisis prediktif (Abadi dkk, 2016).

Keras, di sisi lain, adalah antarmuka tingkat tinggi untuk membangun dan melatih model deep learning dengan cara yang lebih sederhana dan intuitif. Awalnya, Keras dapat berjalan di atas beberapa backend seperti TensorFlow, Theano, dan Microsoft CNTK, tetapi sejak versi 2.4, Keras terintegrasi sepenuhnya dengan TensorFlow sebagai modul default. Keras menawarkan dua API utama, yaitu *Sequential API*, yang digunakan untuk membangun model secara linear dari lapisan-lapisan, dan *Functional API*, yang memungkinkan desain model lebih kompleks, seperti jaringan dengan beberapa input atau output. Dengan Keras, pengguna dapat dengan mudah mendefinisikan, melatih, dan mengevaluasi model dengan hanya beberapa baris kode (Manaswi, 2018).



Gambar 2.5 API Utama yang Dimiliki Keras
(Sumber: Kevinnjagi, 2023)

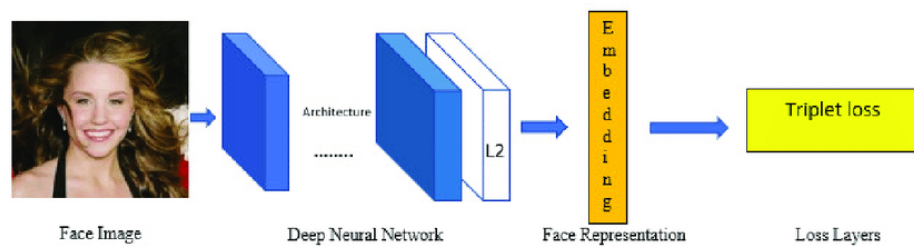
Integrasi antara TensorFlow dan Keras memberikan kombinasi yang ideal antara kekuatan dan kemudahan penggunaan. TensorFlow menangani komputasi berbasis tensor dan optimalisasi kinerja, sedangkan Keras menyediakan antarmuka

yang lebih ramah bagi pengguna untuk mendesain arsitektur model dengan lebih mudah. Dalam pengembangannya, Keras digunakan dalam berbagai aplikasi seperti pengenalan wajah, klasifikasi objek, dan analisis sentimen, berkat fiturnya yang mendukung lapisan konvolusi (CNN), jaringan saraf berulang (RNN), dan model transformer. Dengan ekosistem yang luas, dokumentasi lengkap, serta dukungan komunitas yang besar, TensorFlow dan Keras menjadi standar dalam penelitian dan aplikasi *deep learning* modern.

2.2.7 FaceNet

FaceNet adalah sistem pengenalan wajah berbasis *deep learning* yang dikembangkan oleh Florian Schroff, Dmitry Kalenichenko, dan James Philbin, yang diperkenalkan pada tahun 2015. Sistem ini memetakan wajah langsung ke ruang *Euclidean* berdimensi tertentu, di mana jarak antar titik mencerminkan kesamaan wajah. Dengan pendekatan ini, FaceNet dapat digunakan untuk berbagai tugas seperti verifikasi wajah (*face verification*), pengenalan wajah (*face recognition*), dan pengelompokan wajah (*face clustering*). Salah satu keunggulan utama FaceNet dibandingkan model sebelumnya adalah pendekatannya yang langsung mengoptimalkan *embedding* wajah, berbeda dari metode lain yang mengandalkan lapisan *bottleneck intermediate* untuk ekstraksi fitur (Schroff dkk, 2015).

FaceNet menggunakan *convolutional neural network* (CNN) yang dilatih dengan teknik *triplet loss*, yang terdiri dari triplet wajah (satu anchor, satu wajah positif, dan satu wajah negatif). Metode ini memastikan bahwa *embedding* dari wajah yang sama berada lebih dekat satu sama lain dibandingkan dengan *embedding* dari wajah yang berbeda. Untuk meningkatkan efisiensi pelatihan, FaceNet menerapkan *online triplet mining*, yang memilih triplet yang paling sulit untuk diidentifikasi selama proses training, sehingga sistem dapat belajar lebih efektif. Dibandingkan dengan pendekatan tradisional seperti *Eigenfaces* atau *Fisherfaces* yang berbasis analisis komponen utama, pendekatan FaceNet menghasilkan representasi fitur yang lebih kaya dan mampu mempertahankan informasi struktural wajah dengan lebih akurat (Parkhi dkk, 2015; Taigman dkk, 2014).



Gambar 2.6 Arsitektur Model FaceNet
(Sumber: Ammar dkk, 2022)

Dari segi performa, FaceNet mencapai akurasi 99,63% pada dataset *Labeled Faces in the Wild* (LFW) dan 95,12% pada *YouTube Faces DB*, yang merupakan peningkatan signifikan dibandingkan metode sebelumnya. Studi lebih lanjut menunjukkan bahwa embedding dari FaceNet dapat diintegrasikan dengan berbagai algoritma klasifikasi atau clustering, seperti *Support Vector Machine* (SVM) atau *k-Nearest Neighbors* (k-NN), untuk meningkatkan fleksibilitas penggunaannya dalam sistem keamanan dan otentikasi berbasis biometrik. Keunggulan lain dari FaceNet adalah efisiensi penyimpanan, sehingga memungkinkan aplikasi skala besar dengan konsumsi sumber daya minimal. Dengan keandalannya yang tinggi, FaceNet telah menjadi salah satu metode pengenalan wajah paling banyak digunakan di berbagai bidang, termasuk keamanan, verifikasi identitas, dan sistem akses berbasis biometrik (Liu dkk, 2017). Dimensi *embedding* yang dihasilkan bergantung pada arsitektur dan implementasi model yang digunakan.

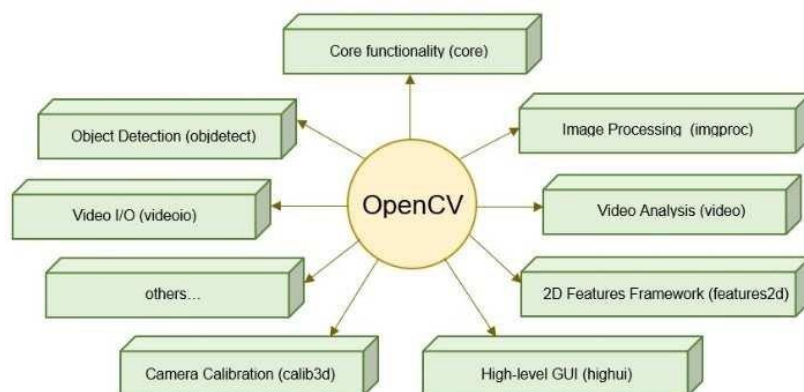
2.2.8 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah pustaka open-source yang dikembangkan oleh Intel pada tahun 1999 untuk mempermudah pengolahan citra dan visi komputer. Sejak diluncurkan, OpenCV telah berkembang menjadi salah satu alat utama dalam penelitian akademik dan aplikasi industri, dengan lebih dari 2,5 juta unduhan serta lebih dari 40.000 anggota komunitas pengguna aktif di seluruh dunia. OpenCV dirancang untuk memproses citra dan video secara efisien menggunakan berbagai algoritma yang dioptimalkan, termasuk deteksi wajah, pelacakan objek, dan pengenalan pola. Pustaka ini kompatibel dengan berbagai platform seperti Windows, Linux, macOS, Android, dan iOS serta

mendukung berbagai bahasa pemrograman seperti C++, Python, dan Java (Culjak dkk, 2012).

Dalam bidang pengenalan wajah, OpenCV memungkinkan penggunaan berbagai algoritma seperti *Eigenface* dan *Viola-Jones* untuk mendeteksi serta mencocokkan fitur wajah. Algoritma Eigenface, yang berbasis *Principal Component Analysis* (PCA), mampu mengidentifikasi seseorang yang terdapat dalam database, tetapi tidak dapat mengenali individu yang tidak terdaftar. Studi oleh Susim dan Darujati (2021) menunjukkan bahwa metode ini dapat mencapai tingkat akurasi hingga 85%, meskipun pencahayaan dan sudut wajah dapat mempengaruhi keakuratan sistem. Selain itu, OpenCV juga memiliki modul pembelajaran mesin (*Machine Learning*) yang mencakup metode seperti *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), dan Jaringan Saraf Tiruan (ANN) untuk meningkatkan akurasi dalam pengenalan wajah serta tugas klasifikasi lainnya (Susim dkk, 2021).

Keunggulan utama OpenCV terletak pada fleksibilitas dan efisiensinya dalam pemrosesan citra secara real-time. Dengan pustaka ini, pengembang dapat menerapkan teknik pemrosesan gambar seperti peningkatan kualitas citra, deteksi tepi, segmentasi objek, dan analisis fitur dengan lebih cepat dan akurat. OpenCV juga dapat diintegrasikan dengan pustaka lain seperti TensorFlow dan PyTorch untuk mendukung pengembangan sistem berbasis kecerdasan buatan (AI) yang lebih canggih (Culjak dkk, 2012).



Gambar 2.7 Diagram Struktur Modul OpenCV

(Sumber: Ralev, 2022)

2.2.9 Transfer Learning

Sebagian besar algoritma *machine learning* bekerja dengan asumsi bahwa data latih dan data yang akan diproses berada pada ruang fitur yang sama dan mengikuti distribusi yang sama. Pada penerapan nyata asumsi tersebut sering kali tidak terpenuhi, misalnya ketika terdapat permasalahan klasifikasi pada suatu domain tertentu namun data latih yang memadai hanya tersedia pada domain lain. *Transfer learning* merupakan pendekatan yang memungkinkan pengetahuan yang telah diperoleh dari satu domain dimanfaatkan untuk menyelesaikan permasalahan pada domain lain (Pan & Yang, 2010).

Dalam *transfer learning* dikenal istilah domain sumber dan domain target. Domain sumber adalah domain tempat pengetahuan diperoleh, biasanya memiliki jumlah data yang besar, sedangkan domain target adalah domain tempat pengetahuan tersebut diterapkan, yang umumnya memiliki keterbatasan data. Tujuan pendekatan ini adalah meningkatkan kemampuan model pada domain target dengan memanfaatkan pengetahuan dari domain sumber, sehingga proses pelatihan tidak perlu dimulai dari awal (Pan & Yang, 2010).

Pada penerapannya dalam *deep learning*, *transfer learning* umumnya dilakukan dengan memanfaatkan model yang telah dilatih sebelumnya pada dataset berskala besar. Model tersebut telah mempelajari ciri-ciri umum dari data selama proses pelatihan awal, sehingga kemampuan mengenali ciri tersebut dapat digunakan kembali untuk permasalahan yang serupa. Salah satu bentuk penerapannya adalah menggunakan model *pretrained* sebagai pengekstrak ciri, di mana bobot model dibiarkan tetap dan keluarannya digunakan sebagai masukan bagi metode klasifikasi yang dilatih secara terpisah.

Pendekatan ini banyak digunakan pada permasalahan dengan jumlah data terbatas, karena melatih jaringan berukuran besar dari awal memerlukan data dalam jumlah sangat banyak agar model dapat mempelajari ciri secara umum dan tidak sekadar menghafal data latih. Dengan memanfaatkan model yang telah dilatih pada dataset berskala besar, kebutuhan tersebut dapat dipenuhi tanpa harus menyediakan data sebanyak itu pada domain target.

2.2.10 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah metode klasifikasi yang diperkenalkan oleh Cortes dan Vapnik pada tahun 1995. Gagasan dasarnya adalah memetakan vektor masukan ke ruang fitur berdimensi tinggi, kemudian membangun bidang pemisah linear pada ruang tersebut. Metode ini pada dasarnya dirancang untuk permasalahan klasifikasi dua kelas (Cortes dkk., 1995).

Bidang pemisah yang dipilih adalah *optimal hyperplane*, yaitu fungsi keputusan linear dengan margin terbesar antara vektor dari kedua kelas. Margin merupakan jarak antara bidang pemisah dengan data terdekat dari masing-masing kelas. Untuk membentuk bidang tersebut tidak seluruh data latih diperlukan, melainkan hanya sebagian kecil data yang berada tepat pada batas margin, yang disebut *support vector*.

Pada kondisi nyata data sering kali tidak dapat dipisahkan secara sempurna. Untuk mengatasinya diperkenalkan konsep *soft margin* yang memperbolehkan sebagian data melanggar batas margin dengan memberikan penalti. Besarnya penalti dikendalikan oleh parameter C , yang mengatur keseimbangan antara kerumitan aturan keputusan dan banyaknya kesalahan yang ditoleransi pada data latih. Agar dapat menangani data yang tidak terpisah secara linear, SVM menggunakan fungsi *kernel*, salah satunya *Radial Basis Function* (RBF) dengan bentuk:

$$K(u, v) = \exp\left(-\frac{\|u - v\|^2}{\sigma^2}\right)$$

Untuk permasalahan dengan lebih dari dua kelas, SVM diperluas dengan membangun beberapa pemisah biner, yaitu satu pemisah untuk setiap kelas, kemudian klasifikasi ditentukan berdasarkan keluaran tertinggi di antara seluruh pemisah tersebut (Cortes dkk., 1995). Kemampuan generalisasi SVM tidak bergantung secara eksplisit pada dimensi ruang pemisahan, melainkan pada jumlah *support vector* yang terbentuk relatif terhadap jumlah data latih, sehingga metode ini sesuai untuk data berdimensi tinggi seperti *embedding* wajah.

2.2.11 Logistic Regression (LR)

Logistic Regression adalah metode klasifikasi yang memodelkan hubungan antara variabel masukan dengan keluaran yang bersifat kategorik. Berbeda dengan regresi linear yang keluarannya dapat bernilai berapa pun, *Logistic Regression* menghasilkan nilai yang selalu berada pada rentang 0 sampai 1 sehingga dapat ditafsirkan sebagai probabilitas keanggotaan suatu kelas (Hosmer & Lemeshow, 2000). Bentuk model tersebut dinyatakan sebagai:

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Fungsi tersebut menghasilkan kurva berbentuk S yang mendekati nilai 0 dan 1 secara bertahap, berbeda dengan garis lurus pada regresi linear yang dapat melampaui kedua batas tersebut. Sifat inilah yang membuat *Logistic Regression* sesuai untuk permasalahan klasifikasi, karena keluarannya selalu berada pada rentang yang wajar sebagai probabilitas.

Melalui transformasi *logit*, yaitu logaritma dari perbandingan antara probabilitas suatu kejadian terjadi dengan probabilitas tidak terjadi, model tersebut dapat dinyatakan kembali dalam bentuk yang linear terhadap parameternya. Dengan demikian prinsip-prinsip yang berlaku pada analisis regresi linear tetap dapat diterapkan pada *Logistic Regression* (Hosmer & Lemeshow, 2000).

Model dasar *Logistic Regression* dirancang untuk keluaran dengan dua kategori. Untuk permasalahan dengan lebih dari dua kelas digunakan perluasan berupa *multinomial logistic regression*, di mana model menghitung probabilitas untuk setiap kelas yang tersedia, kemudian kelas dengan nilai probabilitas tertinggi dipilih sebagai hasil klasifikasi.

2.2.12 Cosine Distance

Cosine distance adalah ukuran ketidakmiripan antara dua vektor yang diturunkan dari *cosine similarity*, yaitu ukuran kemiripan yang dihitung berdasarkan sudut di antara kedua vektor. Berbeda dengan ukuran jarak yang memperhitungkan panjang vektor, ukuran ini hanya memperhatikan arah, sehingga dua vektor yang mengarah ke arah yang sama akan dianggap mirip meskipun

panjangnya berbeda. Nilai *cosine similarity* diperoleh dari hasil kali titik kedua vektor dibagi dengan hasil kali panjang masing-masing vektor:

$$\text{sim}(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$$

Nilai yang dihasilkan berada pada rentang -1 hingga 1 . Nilai mendekati 1 menunjukkan kedua vektor mengarah ke arah yang hampir sama sehingga dianggap sangat mirip, nilai mendekati 0 menunjukkan kedua vektor saling tegak lurus atau tidak memiliki kemiripan, sedangkan nilai negatif menunjukkan arah yang berlawanan. *Cosine distance* kemudian diperoleh sebagai selisih antara satu dengan nilai kemiripan tersebut:

$$\text{dist}(a, b) = 1 - \text{sim}(a, b)$$

Dengan bentuk ini, nilai yang kecil menunjukkan kedua vektor semakin mirip dan nilai yang besar menunjukkan semakin berbeda, sehingga penafsirannya sejalan dengan ukuran jarak pada umumnya. Penggunaan ukuran ini sesuai dengan pendekatan FaceNet yang memetakan wajah ke suatu ruang vektor sedemikian rupa sehingga jarak antar titik pada ruang tersebut mencerminkan tingkat kemiripan wajah (Schroff dkk., 2015). Dengan pendekatan tersebut, identitas suatu wajah dapat ditentukan dengan menghitung jarak *embedding*-nya terhadap *embedding* acuan setiap individu, kemudian mengambil individu dengan jarak terkecil. Suatu ambang batas dapat ditetapkan agar wajah dengan jarak melebihi ambang tersebut ditolak dan tidak dipaksakan menjadi salah satu identitas yang terdaftar.

2.2.13 REST API dan Flask

Application Programming Interface (API) adalah antarmuka yang memungkinkan dua perangkat lunak saling bertukar data tanpa perlu mengetahui rincian implementasi masing-masing. Pada sistem yang terdiri atas beberapa aplikasi klien dan satu server, API berperan sebagai penghubung yang menentukan bagaimana permintaan dikirim dan bagaimana hasilnya dikembalikan.

REST (*Representational State Transfer*) adalah gaya arsitektur untuk sistem terdistribusi yang diperkenalkan oleh Fielding pada tahun 2000. Gaya arsitektur ini dibangun di atas beberapa prinsip utama. Prinsip pertama adalah pemisahan antara klien dan server, sehingga keduanya dapat dikembangkan secara terpisah selama

antarmuka di antara keduanya tetap sama. Prinsip kedua adalah sifat *stateless*, yaitu setiap permintaan dari klien harus memuat seluruh informasi yang diperlukan server untuk memprosesnya, tanpa bergantung pada permintaan sebelumnya. Prinsip ketiga adalah antarmuka yang seragam, di mana setiap sumber daya diidentifikasi melalui alamat tertentu dan diakses menggunakan operasi yang sudah baku (Fielding, 2000).

Dalam penerapannya melalui protokol HTTP, operasi terhadap sumber daya dinyatakan melalui *method* yang berbeda, yaitu GET untuk mengambil data, POST untuk menambahkan data, PUT untuk memperbarui data, dan DELETE untuk menghapus data. Setiap sumber daya diakses melalui alamat yang disebut *endpoint*, sedangkan pertukaran datanya umumnya menggunakan format JSON karena ringan dan mudah diproses oleh berbagai bahasa pemrograman.

Flask adalah *framework* pengembangan aplikasi web berbasis bahasa Python yang tergolong sebagai *micro-framework*. Istilah tersebut menunjukkan bahwa Flask hanya menyediakan fungsi inti dan tidak menyertakan komponen tambahan seperti lapisan abstraksi basis data maupun mekanisme validasi bawaan, sehingga pengembang memiliki keleluasaan untuk menambahkan komponen sesuai kebutuhan. Flask dibangun di atas dua pustaka utama, yaitu Werkzeug yang menangani hal-hal terkait protokol HTTP dan Jinja yang berfungsi sebagai mesin *template* (Ghimire, 2020).

Pada Flask, setiap *endpoint* didefinisikan menggunakan *decorator* yang menghubungkan suatu alamat dengan fungsi tertentu. Ketika permintaan masuk ke alamat tersebut, Flask akan menjalankan fungsi yang bersangkutan dan mengembalikan hasilnya kepada klien. Sifat Flask yang ringan dan tidak memaksakan struktur tertentu menjadikannya sesuai untuk membangun layanan API yang hanya menyediakan pemrosesan data tanpa menghasilkan tampilan antarmuka.

Flask menyediakan server bawaan yang ditujukan untuk keperluan pengembangan. Untuk penggunaan yang berkelanjutan, aplikasi Flask umumnya dijalankan melalui *server* WSGI (*Web Server Gateway Interface*) tersendiri yang lebih stabil dan mampu menangani beberapa permintaan secara bersamaan.

2.2.14 Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam machine learning. Tabel ini membandingkan hasil prediksi model dengan nilai sebenarnya, sehingga kita bisa melihat seberapa akurat model tersebut. Confusion matrix sering digunakan dalam berbagai aplikasi seperti pengenalan wajah, diagnosis medis, dan deteksi spam, karena membantu memahami kesalahan prediksi yang dibuat oleh model (Karimi, 2021).

Di dalam confusion matrix, terdapat empat kategori utama:

- 1) *True Positive* (TP) → Model benar memprediksi kelas positif.
- 2) *True Negative* (TN) → Model benar memprediksi kelas negatif.
- 3) *False Positive* (FP) → Model salah memprediksi kelas negatif sebagai positif (*Type I error*).
- 4) *False Negative* (FN) → Model salah memprediksi kelas positif sebagai negatif (*Type II error*).

Dari confusion matrix, kita bisa menghitung akurasi, presisi, recall, dan F1-score untuk mengukur performa model. Akurasi menunjukkan persentase prediksi yang benar, presisi mengukur seberapa andal model dalam mengidentifikasi kelas positif, recall menunjukkan seberapa baik model menangkap semua kasus positif, dan F1-score adalah keseimbangan antara presisi dan recall. Dengan memahami confusion matrix, kita bisa mengetahui apakah model lebih sering melakukan kesalahan False Positive atau False Negative dan melakukan perbaikan agar hasilnya lebih akurat (Claude Sammut, 2011).

Tabel 2.2 *Confusion Matrix*

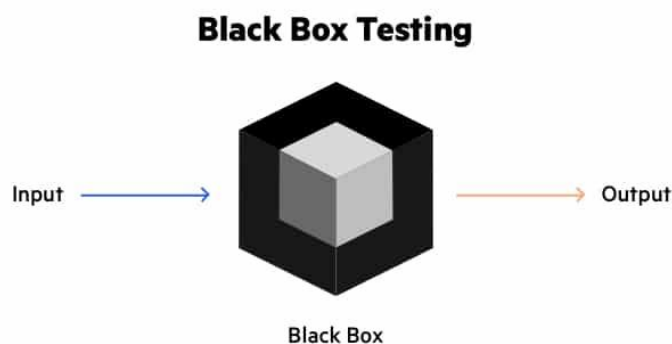
		Kelas Sebenarnya	
		Positive	Negative
Kelas Prediksi	Positive	TP	FN
	Negative	FP	TN

2.2.15 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsionalitas tanpa memeriksa struktur internal atau kode sumbernya. Teknik ini digunakan untuk menguji apakah sistem dapat beroperasi dengan benar berdasarkan serangkaian masukan yang diberikan. Pengujian ini bertujuan untuk menemukan kesalahan yang terdapat dalam komponen perangkat lunak dan memastikan bahwa perangkat lunak memenuhi spesifikasi yang diharapkan (Khan & Khan, 2012).

Black Box Testing digunakan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan, sehingga sering digunakan dalam tahap pengujian fungsional, pengujian integrasi, dan *User Acceptance Testing* (UAT). Pengujian ini dapat dilakukan oleh pengembang, penguji independen, atau bahkan pengguna akhir, karena tidak memerlukan pemahaman tentang kode sumber.

Keunggulan utama Black Box Testing adalah efisiensinya dalam menguji perangkat lunak tanpa memerlukan pemahaman teknis tentang kode program, sehingga bisa dilakukan oleh penguji independen dan pengguna akhir. Pengujian ini juga lebih berorientasi pada kebutuhan pengguna karena hanya berfokus pada fungsionalitas sistem (Khan & Khan, 2012).



Gambar 2.8 Black Box Testing
(Sumber: ITBOX, 2023)

2.2.16 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahap akhir dalam proses pengujian perangkat lunak yang dilakukan oleh pengguna akhir atau perwakilan bisnis untuk memastikan bahwa sistem telah memenuhi kebutuhan bisnis sebelum diimplementasikan secara penuh. UAT berfokus pada validasi bahwa perangkat lunak bekerja sebagaimana yang diharapkan dalam skenario dunia nyata, bukan hanya berdasarkan spesifikasi teknis yang diuji dalam tahap pengujian sebelumnya. Pengujian ini memastikan bahwa aplikasi dapat digunakan dengan lancar oleh pengguna dan telah memenuhi standar kualitas yang diperlukan sebelum peluncuran resmi (Hareton K.N. Leung & Peter W.L. Wong, 1997).

Dalam pelaksanaannya, UAT melibatkan berbagai pendekatan, seperti pengujian berbasis perilaku (*behavior-based testing*) dan pengujian kotak hitam (*black-box testing*). Pengujian berbasis perilaku berfokus pada pengujian sistem dari perspektif pengguna, dengan menggunakan skenario realistis untuk memastikan bahwa fitur utama bekerja dengan baik. Sementara itu, pengujian kotak hitam memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditetapkan tanpa melihat kode sumbernya. Leung & Wong (1997) memperkenalkan strategi *Operation-Based Testing (OBT)* yang menggunakan profil operasional untuk menguji perangkat lunak berdasarkan cara penggunaan aktualnya. Pendekatan ini dianggap lebih efektif dalam mengidentifikasi potensi masalah sebelum implementasi akhir (Hareton K.N. Leung & Peter W.L. Wong, 1997).

Keberhasilan UAT sangat bergantung pada *acceptance criteria* yang telah ditetapkan sebelumnya oleh tim pengembang dan pengguna bisnis. Jika perangkat lunak memenuhi kriteria ini, maka sistem dapat disetujui untuk digunakan secara operasional. Namun, jika ditemukan kekurangan atau bug kritis, perangkat lunak harus diperbaiki sebelum dapat diterima. Dengan mengikuti prosedur UAT yang ketat, organisasi dapat memastikan bahwa perangkat lunak yang akan digunakan telah diuji dengan baik, mengurangi risiko kegagalan setelah implementasi, dan meningkatkan kepercayaan pengguna terhadap sistem baru yang akan diterapkan (Hareton K.N. Leung & Peter W.L. Wong, 1997).

2.2.17 Pengujian Analisis

Pengujian analisis dilakukan untuk mengevaluasi pengaruh berbagai faktor yang dapat memengaruhi kinerja sistem dalam mendeteksi wajah siswa di dalam kelas. Faktor utama yang diuji meliputi jarak antara kamera dan siswa, variasi pencahayaan, serta sudut pandang wajah saat proses deteksi berlangsung. Pengujian ini bertujuan untuk memahami sejauh mana kondisi lingkungan dapat memengaruhi akurasi sistem, sehingga dapat dilakukan optimasi terhadap parameter yang digunakan dalam model pengenalan wajah.

Dalam proses pengujian, sistem diuji dalam berbagai skenario yang mencerminkan kondisi nyata di lingkungan kelas, seperti pencahayaan terang, redup, atau tidak merata, serta variasi posisi siswa terhadap kamera. Data hasil deteksi dianalisis berdasarkan tingkat akurasi, jumlah wajah yang berhasil dikenali, serta jumlah kesalahan deteksi. Dengan membandingkan hasil pengujian dari berbagai skenario, dapat diketahui kondisi optimal di mana sistem bekerja dengan akurasi tertinggi.

Hasil pengujian ini menjadi dasar untuk meningkatkan performa sistem, baik dengan penyesuaian parameter model, peningkatan kualitas pencahayaan di ruangan, maupun pengaturan posisi kamera agar dapat menangkap wajah siswa dengan lebih baik. Dengan memahami faktor-faktor yang memengaruhi deteksi wajah, sistem dapat lebih dioptimalkan untuk bekerja dengan baik dalam berbagai kondisi di lingkungan kelas.