

BAB I

PENDAHULUAN

1.1 Latar Belakang

Wajah merupakan salah satu fitur biometrik manusia yang memiliki keunggulan dibandingkan fitur biometrik lainnya. Keunggulan utama pengenalan wajah adalah sifatnya yang alami, mudah dikenali baik oleh manusia maupun mesin, serta dapat dideteksi dari jarak jauh tanpa perlu kontak fisik. Teknologi pengenalan wajah saat ini banyak diterapkan dalam berbagai sistem keamanan dan pengawasan, seperti di bandara, pusat perbelanjaan, dan area parkir. Selain itu, teknologi ini juga digunakan dalam sistem kontrol akses dan absensi kehadiran (Alhanaee *et al.*, 2021). Namun demikian, akurasi sistem pengenalan wajah sangat dipengaruhi oleh kondisi lingkungan, terutama pencahayaan.

Deteksi wajah dalam kondisi gelap merupakan tantangan besar dalam berbagai bidang, termasuk keamanan publik, penegakan hukum, dan sistem pengawasan otomatis. Sistem CCTV konvensional yang bekerja pada spektrum cahaya tampak (*Visible Spectrum*) umumnya gagal menangkap detail wajah dengan jelas dalam kondisi minim cahaya, sehingga proses deteksi maupun pengenalan wajah menjadi sulit dilakukan (Bhargava *et al.*, 2023). Hal ini berdampak langsung pada efektivitas sistem keamanan berbasis visual yang diandalkan di banyak tempat umum.

Metode pengenalan wajah tradisional berbasis citra Visible Spectrum (VIS) sangat bergantung pada sumber cahaya eksternal, sehingga rentan terhadap perubahan intensitas dan arah pencahayaan. Pada lingkungan dengan cahaya redup atau gelap, kualitas citra menurun secara signifikan dan berimbas pada akurasi sistem dalam mengenali wajah (Rana *et al.*, 2022). Meskipun pengenalan wajah memiliki keunggulan dibandingkan metode biometrik lain karena bersifat non-kontak dan dapat digunakan dari jarak jauh, namun keterbatasannya pada kondisi pencahayaan rendah menjadi hambatan utama dalam penerapannya di dunia nyata.

Teknologi pencitraan *Near-Infrared* (NIR) muncul sebagai solusi terhadap keterbatasan pencahayaan pada sistem pengenalan wajah. NIR bekerja dengan panjang gelombang elektromagnetik antara 780–2500 nm, lebih panjang dari cahaya tampak (VIS) namun lebih pendek dari inframerah tengah (*Short wave IR*). Citra NIR mampu merekam detail wajah dalam kondisi minim cahaya tanpa ketergantungan terhadap sumber cahaya tampak (VIS) (Wu *et al.*, 2023). Oleh karena itu, NIR menjadi alternatif ideal untuk sistem pengenalan wajah di malam hari atau pada area dengan pencahayaan terbatas.

Namun, tantangan berikutnya terletak pada jarak akuisisi citra. Banyak sistem pengawasan berbasis CCTV gagal menangkap wajah secara jelas ketika subjek berada pada jarak jauh karena degradasi resolusi dan kontras citra. Untuk

mengatasi hal tersebut, peningkatan kualitas citra dan optimalisasi algoritma deteksi wajah menjadi langkah krusial.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan metode pengenalan wajah jarak jauh dalam kondisi gelap berbasis citra NIR dengan menggabungkan teknik *Selective Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *Multi-task Cascaded Convolutional Neural Network* (MTCNN). Teknik Selective CLAHE diterapkan untuk meningkatkan kontras lokal citra agar detail wajah pada jarak jauh tetap terlihat jelas, sedangkan MTCNN digunakan untuk mendeteksi dan mengekstraksi area wajah dengan lebih presisi. Selanjutnya, hasil deteksi wajah diproses menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *InceptionResNetV2* guna menghasilkan sistem pengenalan wajah yang akurat dan andal dalam kondisi gelap dan jarak jauh.

1.2 Landasan Teori

1.2.1 Deteksi Wajah

Face Detection atau deteksi wajah adalah tugas penting dalam visi komputer dalam berbagai bidang seperti interaksi manusia-komputer, pengawasan, dan sistem biometric. Face detection adalah metode untuk mendeteksi wajah dari sebuah citra gambar yang memiliki beberapa fitur pada gambar tersebut. Hal ini melibatkan identifikasi dan lokalisasi wajah dalam gambar yang dapat menjadi tantangan karena variasi ukuran, bentuk, warna dan pencahayaan. Deteksi wajah menjadi tugas yang menantang karena karakteristik wajah yang tidak kaku dan dapat berubah dalam ukuran, bentuk, warna, dll. Deteksi wajah menjadi tugas yang lebih menantang ketika gambar yang diberikan tidak jelas, fitur wajah terhalang oleh benda lain, faktor pencahayaan yang tidak tepat dan wajah yang tidak menghadap kamera (Meena & Sharan, 2016).

Metode-metode ini menganalisis piksel-piksel citra dengan menggunakan teknik pengklasifikasi berbasis fitur, jaringan saraf tiruan, atau kombinasi metode berbasis aturan. Deteksi wajah melibatkan serangkaian tahapan, termasuk ekstraksi fitur, pelatihan model dan proses pengklasifikasi untuk mengenali wajah manusia dengan tingkat akurasi yang tinggi. Aplikasi deteksi wajah meliputi pengenalan wajah, identifikasi keamanan, pengolahan video, dan pengenalan emosi (Viola & Jones, 2001).

1.2.2 Face Recognition

Face recognition atau pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam sistem keamanan selain pengenalan retina mata, pengenalan sidik jari dan iris mata (Bah dan Ming, 2020). Pengenalan wajah merupakan sebuah tantangan yang memberikan variasi data tertentu karena beragam variasi data dari orang yang berbeda, termasuk variasi sistematis dari beberapa faktor seperti kondisi kecerahan dan pose seseorang (Barnouti, Hazim *et al.*, 2016).

Dalam beberapa kasus, kondisi di sekitar orang tersebut juga hal yang penting dalam sistem rekognisi wajah manusia. Menangani ukuran data yang

besar dan menggabungkannya merupakan hal yang sulit untuk sebuah sistem rekognisi. Bagaimanapun, mengingat beberapa wajah juga hal yang sulit. Kunci utama dari sebuah sistem pembelajaran adalah kapasitas memori. Fitur wajah yang mungkin bisa digunakan untuk rekognisi wajah masih terus dipelajari. Dengan kata lain, diperlukan fitur umum dan spesifik untuk rekognisi wajah (Hazim Barnouti dkk., 2016).

1.2.3 Near Infrared (NIR)

Near Infrared (NIR) atau inframerah dekat adalah radiasi elektromagnetik dengan panjang gelombang yang lebih panjang dari cahaya tampak (VIS) tetapi lebih pendek dari radiasi inframerah tengah. Panjang gelombang NIR umumnya berada di antara 780 nanometer (nm) dan 2500 nm seperti pada gambar 13 (Farokhi, Sajad *et al.*, 2016). Meskipun tidak terlihat oleh mata manusia, NIR memiliki beberapa sifat yang membuatnya berguna dalam berbagai aplikasi.

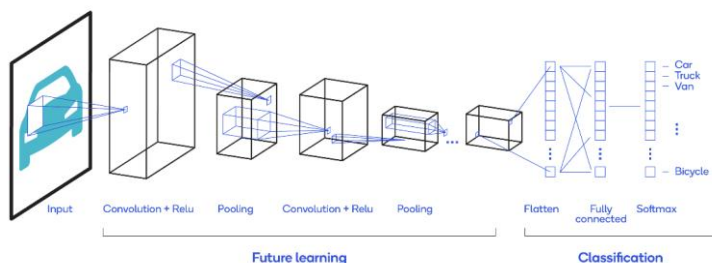
Visible	Infrared				
	Near IR	Short wave IR	Medium wave IR	Long wave IR	
0.4μm	0.75μm	1.4μm	3μm	8μm	15μm

Gambar 1 Rentang Spektrum Radiasi (Farokhi, Sajad *et al.*, 2016)

1.2.4 Convolutional Neural Network (CNN)

Convolutional network atau yang dikenal dengan *Convolutional Neural Network* (CNN) adalah tipe khusus dari *neural network* untuk memproses data yang mempunyai topologi jala atau *grid-like topology*. Pemberian nama *convolutional neural network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linear. Jadi *convolutional network* adalah *neural network* yang menggunakan konvolusi minimal pada salah satu lapisannya (LeCun *et al.*, 2015). *Convolutional neural network* merupakan *special case* dari *Artificial Neural Network* (ANN) yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah *object recognition* dan *detection*.

Arsitektur CNN terdiri atas beberapa layer yaitu *convolution layer*, fungsi *activation layer*, *pooling layer*, dan *fully connected layer*. Ilustrasi jaringan arsitektur *Convolutional Neural Network* dapat dilihat pada gambar 1.



Gambar 2 Arsitektur CNN

(Sumber: <https://developer.qualcomm.com/blog/exploring-different-types-neural-networks>)

Tahap pertama dalam arsitektur CNN adalah tahap konvolusi. Kemudian dilanjutkan fungsi aktivasi menggunakan fungsi aktivasi ReLu (*Rectifier Linear Unit*), selanjutnya dilanjutkan dengan proses *pooling*. Proses ini diulang terus menerus sampai didapatkan *feature map* yang cukup untuk dilanjutkan ke *fully connected neural network*, sehingga dapat dihasilkan *output class*. Penjelasan secara detail dari tahap arsitektur CNN tersebut dijelaskan sebagai berikut:

a. *Convolutional Layer*

Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua offset yang memungkinkan seperti yang ditunjukkan pada gambar 2 Kotak hijau secara keseluruhan adalah citra yang akan di konvolusi. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua offset yang memungkinkan seperti yang ditunjukkan pada gambar 2 Kotak hijau secara keseluruhan adalah citra yang akan di konvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar di sebelah kanannya.



Gambar 3 Proses *convolution* (Khan *et al.*, 2018)

b. *Rectified Linear Unit* (ReLU)

ReLU bertujuan untuk menjaga hasil citra proses konvolusi berada pada domain definit positif. ReLU merupakan salah satu fungsi aktivasi populer dalam *deep neural network*. Pada fungsi aktivasi dapat mengubah jumlah angka pembobotan dari masukan yang masuk ke dalam *neuron* buatan. Fungsi ini harus bersifat *non-linear* untuk pengkodean pola yang kompleks dari data. Aktivasi memiliki dua jenis yaitu *Sigmoid* dan *Tanh*. Persamaan (1) yang biasa digunakan dalam fungsi ini adalah:

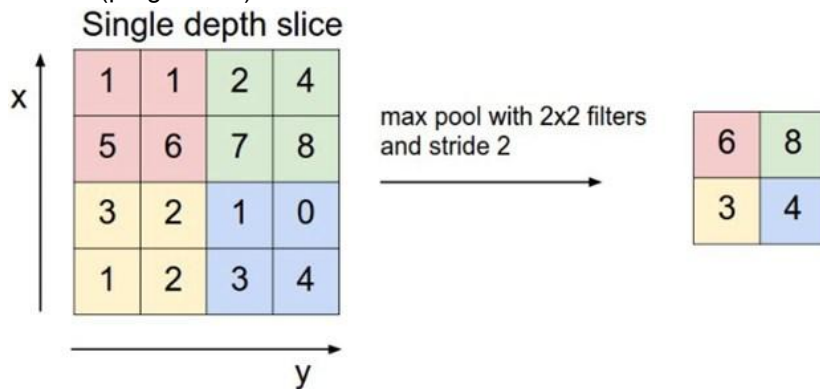
$$f(x) = \max(x, 0) \quad (1)$$

Di mana:

x = nilai pada *feature map*

c. *Subsampling Layer* atau *Pooling Layer*

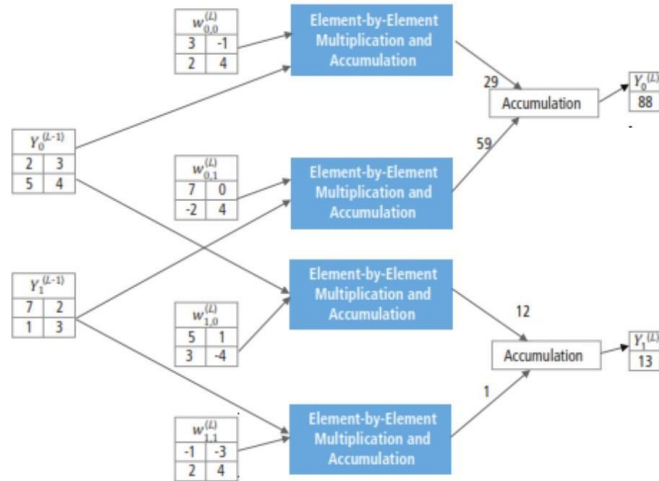
Proses ini bertujuan untuk mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *subsampling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Pada sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada gambar 3. *Grid* yang berwarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan *grid* disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran).



Gambar 4 Operasi max pooling (Khan *et al.*, 2018)

d. *Fully Connected Layer*

Fully connected layer merupakan sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung dengan neuron lapisan selanjutnya. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Perbedaan *fully connected layer* dengan konvolusi biasa adalah neuron pada lapisan konvolusi terhubung hanya ke daerah tertentu sedangkan *fully connected* memiliki neuron yang semuanya terhubung (Khan *et al.*, 2018). Namun keduanya masih menggunakan operasi dot, sehingga fungsinya tidak begitu berbeda. Berikut adalah proses *fully connected*:



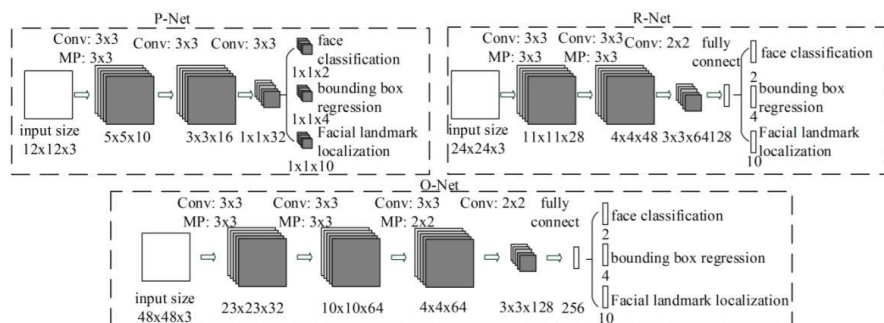
Gambar 5 Proses *fully connected layer* (Khan *et al.*, 2018)

1.2.5 Multi-task Cascaded Convolutional Neural Network (MTCNN)

Multi-task Cascaded Convolutional Neural Network (MTCNN) merupakan salah satu variasi dari metode *Deep Convolutional Neural Network* yang biasa digunakan untuk mengenali atau mendeteksi wajah, MTCNN terdiri dari 3 jaringan terpisah: P-Net, R-Net, dan O-Net.

a. Jaringan P-Net

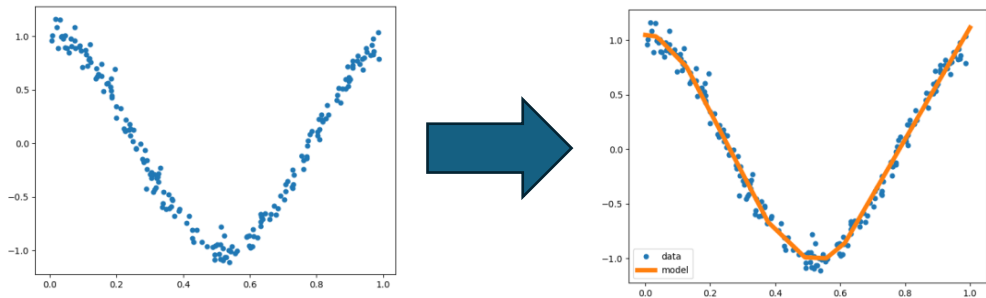
Pada jaringan P-Net, untuk setiap gambar, kernel 12x12 menelusuri gambar, mencari wajah. Kernel 12x12, perlahan-lahan bergerak melintas secara horizontal sampai sudut bawah gambar, mencari wajah. Begitupun di jaringan R-Net dan Onet melakukan penelusuran mendeteksi wajah yang membedakan di setiap jaringan ada pada bagian kernel, yaitu P-Net menggunakan kernel 12x12, R-Net menggunakan kernel 24x24 dan O-Net menggunakan kernel 48x48. Seperti yang direpresentasikan pada gambar 6



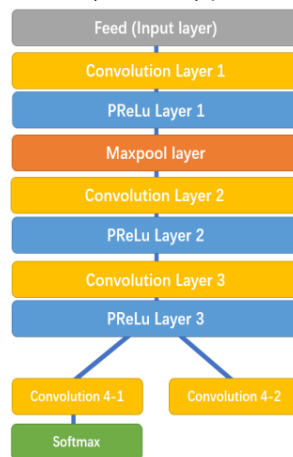
Gambar 6 Arsitektur MTCNN (Zhang, Kaipeng *et al.*, 2016)

Dalam masing-masing 12x12 kernel ini, dibagi menjadi tiga tahapan konvolusi dijalankan dengan 3x3 kernel. Setelah setiap lapisan konvolusi

selesai dijalankan, lapisan Parametric Rectified Linear Unit (pReLU) akan menghilangkan vanishing gradient dengan cara menerapkan fungsi aktivasi element. Contoh implementasi pReLU dapat dilihat pada gambar 7



Gambar 7 Ilustrasi penggunaan fungsi pReLU (medium.com) setelah lapisan konvolusi ketiga, jaringan terbagi menjadi dua lapisan. Aktivasi dari lapisan ketiga dibagi ke dua lapisan konvolusi yang terpisah, dan lapisan *softmax* memberikan probabilitas desimal untuk setiap hasil, dan probabilitas bertambah hingga satu. Dalam kasus ini, ia menghasilkan dua probabilitas, yaitu probabilitas bahwa ada wajah di daerah tersebut dan probabilitas bahwa tidak ada wajah. Pada gambar 8 merupakan representasi dari cara kerja *Multi-task Cascaded Convolutional Neural Network* (MTCNN) pada bagian jaringan P-Net

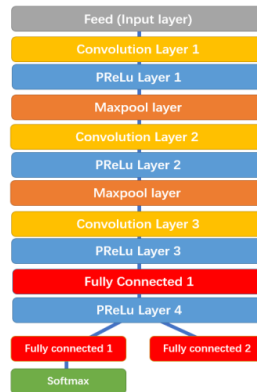


Gambar 8 Representasi cara kerja MTCNN pada jaringan P-Net (Paula, Thomas da Silva, 2017)

Konvolusi 4-1 menghasilkan probabilitas suatu wajah berada di setiap kotak pembatas (*bounding box*), dan konvolusi 4-2 menghasilkan koordinat dari *bounding box*.

b. Jaringan R-Net

R-Net memiliki struktur yang serupa, tetapi dengan lebih banyak lapisan. Menggunakan vektor *bounding box* P-Net sebagai inputnya, dan menyempurnakan koordinatnya. Gambar 9 menunjukkan bagaimana cara kerja MTCNN pada jaringan R-Net



Gambar 9 Representasi cara kerja MTCNN pada jaringan R-Net (Paula, Thomas da Silva, 2017)

Demikian pula, R-Net terbagi menjadi dua lapisan pada ujungnya, memberikan dua output : yaitu *Fully Connected* 1 berisi probabilitas suatu wajah berada di setiap kotak pembatas (*bounding box*) dan *Fully Connected* 2 berisi koordinat kotak pembatas (*bounding box*) yang baru dan menambah tingkat kepastian deteksi pada setiap kotak pembatas (*bounding box*).

c. Jaringan O-Net

Jaringan terakhir yaitu, O-Net mengambil kotak pembatas (*bounding box*) RNet sebagai input dan bertugas menandai koordinat landmark wajah. Pada gambar 10 merupakan representasi dari cara kerja *Multi-task Cascaded Convolutional Neural Network* (MTCNN) pada bagian jaringan O-Net

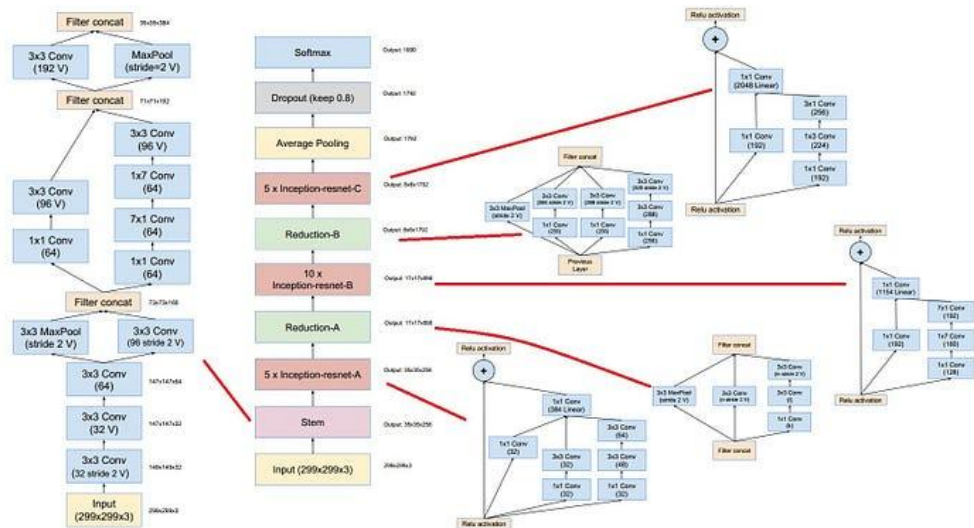


Gambar 10 Representasi cara kerja MTCNN pada jaringan O-Net
(Paula, Thomas da Silva, 2017)

O-Net terbagi menjadi tiga lapisan diujungnya, memberikan tiga output berbeda: *Fully Connected 1* yaitu probabilitas wajah berada di dalam *bounding box*, *Fully Connected 2* yaitu koordinat kotak pembatas (*bounding box*), dan *Fully Connected 3* yaitu koordinat landmark wajah (lokasi mata, hidung, dan mulut).

1.2.6 InceptionResNetV2

InceptionResNetV2 adalah arsitektur jaringan saraf konvolusional lanjutan yang menggabungkan efisiensi blok Inception dengan stabilitas residual connection dari ResNet. Gabungan ini memungkinkan pembelajaran fitur yang lebih dalam dan kompleks, namun tetap menjaga efisiensi komputasi dan kestabilan proses pelatihan (Szegedy *et al.*, 2016).



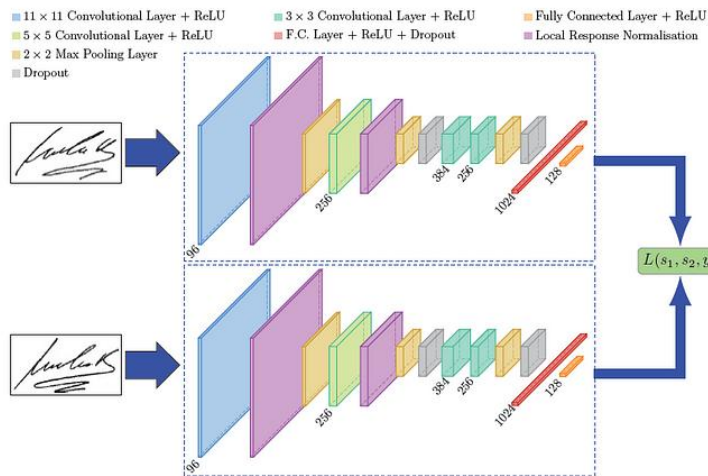
Gambar 11 Struktur *InceptionResNetV2*

Berbeda dengan struktur CNN secara umum yang hanya menumpuk layer konvolusi secara linear, *InceptionResNetV2* memanfaatkan berbagai ukuran kernel dalam satu blok untuk menangkap informasi spasial dari berbagai skala, sekaligus menggunakan koneksi pintasan untuk mempercepat konvergensi dan menghindari degradasi performa saat jaringan semakin dalam. Pendekatan ini terbukti efektif dalam menangani tugas klasifikasi dan embedding, termasuk pengenalan wajah (He *et al.*, 2016).

Dalam penelitian ini, *InceptionResNetV2* dimanfaatkan sebagai bagian dari feature extractor dalam sistem untuk menghasilkan vektor representasi (embedding) dari citra wajah.

1.2.7 Siamese Network

Siamese Network adalah arsitektur jaringan saraf yang terdiri dari dua atau lebih subnet identik yang berbagi bobot (shared weights) dan digunakan untuk mempelajari kesamaan antara dua input. Arsitektur ini dirancang untuk menyelesaikan masalah klasifikasi berbasis perbandingan (comparison-based classification), dengan memfokuskan pembelajaran pada seberapa mirip atau berbeda dua data input satu sama lain (Bromley *et al.*, 1993).



Gambar 12 Implementasi Siamese Network

Pada implementasinya, masing-masing cabang dari Siamese Network menerima satu input dan memprosesnya melalui jaringan yang sama hingga menghasilkan embedding atau representasi fitur. Representasi ini kemudian dibandingkan menggunakan fungsi jarak seperti Euclidean distance atau cosine similarity. Hasil perbandingan inilah yang menentukan apakah kedua input berasal dari kelas yang sama atau tidak (Schroff *et al.*, 2015).

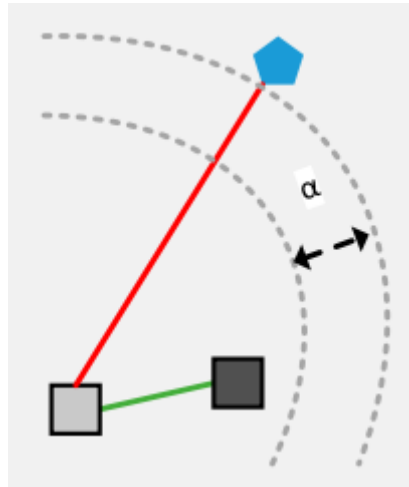
Salah satu keunggulan utama dari Siamese Network adalah kemampuannya untuk bekerja dengan baik pada situasi data terbatas, seperti pada pengenalan wajah berbasis Near-Infrared (NIR), di mana jumlah data pelatihan bisa sangat terbatas. Karena model ini tidak memerlukan pelatihan ulang untuk setiap kelas baru, ia sangat cocok untuk sistem identifikasi terbuka (open-set recognition) atau verifikasi wajah (Schroff *et al.*, 2015).

Dalam konteks pengenalan wajah, Siamese Network sering dipadukan dengan triplet loss untuk mengoptimalkan jarak antar embedding. Strategi ini memungkinkan model untuk secara eksplisit belajar merepresentasikan wajah-wajah dari orang berbeda agar sejauh mungkin dalam ruang vektor, sambil menjaga jarak antar embedding dari wajah yang sama tetap dekat (Schroff *et al.*, 2015).

1.2.8 Triplet Loss

Triplet Loss adalah fungsi kerugian (loss function) yang banyak digunakan dalam pembelajaran mesin, khususnya pada tugas-tugas yang melibatkan pembelajaran representasi seperti pengenalan wajah, verifikasi gambar, atau pencarian informasi berbasis kesamaan. Konsep ini pertama kali diperkenalkan dalam konteks deep metric learning untuk menghasilkan representasi vektor (embedding) yang optimal, di mana data dari kelas yang sama berada lebih dekat

dalam ruang vektor, sedangkan data dari kelas yang berbeda berada lebih jauh (Schroff *et al.*, 2015).



Gambar 13 Ilustrasi Triplet Loss

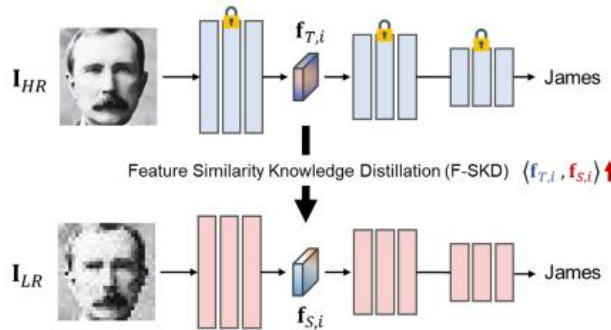
Triplet Loss bekerja dengan menggunakan tiga sampel data dalam satu waktu, yang dikenal sebagai triplet: anchor (A), positive (P), dan negative (N). Anchor adalah sampel acuan dari kelas tertentu, positive adalah sampel dari kelas yang sama dengan anchor, dan negative adalah sampel dari kelas yang berbeda. Tujuan Triplet Loss adalah meminimalkan jarak antara anchor dan positive ($d(A, P)$) serta memaksimalkan jarak antara anchor dan negative ($d(A, N)$) dalam ruang embedding. Secara matematis, Triplet Loss didefinisikan sebagai berikut:

$$Loss = \max(d(A, P) - d(A, N) + \alpha, 0) \quad (2)$$

di mana d adalah fungsi jarak (biasanya jarak Euclidean atau Cosine Distance), α adalah margin atau parameter yang menentukan jarak minimum antara anchor-negative dibandingkan anchor-positive, dan fungsi $\max(\dots, 0)$ memastikan kerugian hanya dihitung jika jarak anchor-positive lebih besar dari jarak anchor-negative ditambah margin (Schroff *et al.*, 2015).

1.2.9 Knowledge Distillation

Konsep *knowledge distillation* (KD) pertama kali diperkenalkan sebagai metode untuk mentransfer pengetahuan dari model jaringan saraf yang besar dan kompleks (disebut "guru" atau *teacher*) ke model yang lebih kecil dan lebih efisien secara komputasi (disebut "murid" atau *student*). Idennya adalah bahwa model guru yang besar, setelah dilatih, memiliki "pengetahuan" yang lebih kaya tentang data daripada sekadar label yang benar (*hard targets*). Pengetahuan ini terkandung dalam output probabilitasnya sebelum fungsi aktivasi akhir (logits).



Gambar 14 Ilustrasi Konsep *Knowledge Distillation* (Shin *et al.*, 2023)

Dengan melatih model murid untuk meniru logits dari model guru (yang dihaluskan menggunakan parameter "suhu" atau temperature untuk mengungkap relasi antar kelas), model murid dapat belajar untuk menggeneralisasi dengan cara yang sama seperti model guru, sehingga mencapai akurasi yang jauh lebih tinggi daripada jika dilatih hanya menggunakan hard targets (Shin *et al.*, 2023)

1.2.10 Cosine Similarity

Cosine Similarity adalah metrik yang digunakan untuk mengukur kesamaan antara dua vektor dalam ruang multidimensi berdasarkan sudut di antara mereka. Metrik ini menghitung kosinus sudut antara dua vektor, menghasilkan nilai dalam rentang $[-1, 1]$, dimana nilai 1 menunjukkan kesamaan sempurna, 0 menunjukkan tidak ada korelasi, dan -1 menunjukkan perbedaan sempurna (Jurafsky & Martin, 2023).

Secara matematis, cosine similarity antara dua vektor A dan B didefinisikan sebagai:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (3)$$

Dalam konteks pembelajaran metrik (metric learning), triplet loss memanfaatkan cosine similarity untuk mempelajari representasi fitur yang efektif dengan memaksimalkan jarak (margin) antara sampel negatif dan meminimalkan jarak antara sampel positif (Schroff *et al.*, 2015). Triplet loss bekerja dengan tiga

elemen: anchor (a), positive (p), dan negative (n), di mana tujuannya adalah memastikan bahwa:

$$\text{cosine similarity}(a,p) > \text{cosine similarity}(a,n) + \text{margin}$$

Dengan demikian, cosine similarity berperan sebagai fungsi pengukuran jarak yang membantu model membedakan antara pasangan yang mirip dan tidak mirip .

1.2.11 Tolak Ukur Evaluasi

Evaluasi kinerja model dilakukan berdasarkan tolak ukur (metrik) berikut:

1. Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk melihat bagaimana model mengklasifikasikan pasangan wajah yang sama dan berbeda. Dalam konteks pengenalan wajah, confusion matrix menampilkan jumlah pasangan wajah yang sebenarnya sama namun diidentifikasi dengan benar sebagai sama (true similar), jumlah pasangan wajah yang sebenarnya berbeda dan diidentifikasi dengan benar sebagai berbeda (true different), serta kesalahan-kesalahan prediksi seperti pasangan wajah yang sebenarnya sama tetapi diklasifikasikan sebagai berbeda (false negative), dan pasangan wajah yang sebenarnya berbeda namun diklasifikasikan sebagai sama (false positive). Matriks ini membantu menganalisis performa model secara lebih mendetail dan menjadi dasar untuk menghitung metrik lainnya seperti akurasi, precision, recall, dan F1-score (Kaya and Bilge, 2019).

Pada studi kasus ini, confusion matrix diatur sebagai berikut:

Tabel 1. Pemetaan Confusion Matrix

	Diprediksi Sama	Diprediksi Berbeda
Sebenarnya Sama	True Positive	False Negative
Sebenarnya Berbeda	False Positive	True Negative

Dari table ini dapat dihitung:

a) True Acceptance Rate (TAR):

$$TAR = \frac{TP}{TP+FN} \quad (4)$$

b) False Acceptance Rate (FAR):

$$FAR = \frac{FP}{FP+TN} \quad (5)$$

c) False Rejection Rate (FRR):

$$FRR = \frac{FN}{TP+FN} \quad (6)$$

2. TAR@FAR

TAR@FAR adalah salah satu metrik evaluasi yang mengukur seberapa baik model mengenali pasangan wajah yang sama, pada tingkat kesalahan tertentu. TAR (True Accept Rate) menunjukkan persentase pasangan

wajah yang benar-benar sama yang berhasil diidentifikasi dengan benar sebagai wajah yang sama. Sementara FAR (False Accept Rate) menunjukkan persentase pasangan wajah yang sebenarnya berbeda namun secara keliru dikenali sebagai wajah yang sama. Metrik TAR@FAR menampilkan kinerja model pada tingkat kesalahan yang diizinkan. Misalnya, TAR@FAR=0.01 menunjukkan seberapa banyak pasangan wajah yang benar-benar sama dapat dikenali ketika model dibatasi hanya menerima 1% pasangan wajah yang berbeda secara salah (Kaya and Bilge, 2019).

$$TAR@FAR = \frac{\text{Jumlah TP pada tingkatan FP tertentu}}{TP+FN} \quad (7)$$

3. Equal Error Rate (EER)

EER adalah salah satu metrik evaluasi yang banyak digunakan dalam sistem verifikasi biometrik, termasuk pengenalan wajah. EER diperoleh pada titik di mana False Acceptance Rate (FAR) sama dengan False Rejection Rate (FRR), yang artinya pada saat itu tingkat kesalahan sistem dalam menerima sampel yang salah (false acceptance) sama dengan tingkat kesalahan dalam menolak sampel yang benar (false rejection). Nilai EER yang lebih rendah menunjukkan kinerja sistem yang lebih baik, karena menandakan sistem mampu memisahkan dengan baik antara data yang benar-benar mirip (genuine) dan data yang tidak mirip (impostor). Perhitungan EER dilakukan dengan memplot kurva FAR dan FRR terhadap threshold, kemudian mencari titik potong antara keduanya. Metrik ini sering digunakan karena memberikan ukuran tunggal yang ringkas mengenai trade-off antara kesalahan false acceptance dan false rejection dalam sistem verifikasi wajah (Kaya and Bilge, 2019).

4. Area Under the Receiver Operating Characteristic Curve (ROC-AUC)

ROC-AUC merupakan ukuran kemampuan model dalam membedakan antara pasangan wajah yang sama dengan yang berbeda. ROC-AUC dihitung dengan memplot hubungan antara True Accept Rate (TAR) dan False Accept Rate (FAR) pada berbagai nilai ambang (threshold), kemudian menghitung area di bawah kurva tersebut. Semakin tinggi nilai AUC (maksimal 1.0), semakin baik model dalam memisahkan pasangan wajah yang sama dengan yang berbeda (Kaya and Bilge, 2019).

1.3 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka rumusan masalah pada penelitian ini yakni sebagai berikut:

1. Bagaimana mengenali wajah dalam kondisi gelap?
2. Bagaimana mengenali wajah dalam kondisi gelap dengan jarak jauh dan resolusi gambar rendah?

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini yakni:

1. Sistem yang dapat mengenali wajah dalam kondisi gelap
2. Sistem yang dapat mengenali wajah dalam kondisi gelap dengan jarak jauh dan resolusi gambar yang rendah

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Menghasilkan sistem yang dapat mengenali wajah jarak jauh dalam kondisi gelap dan gambar dengan resolusi rendah
2. Mendorong pengembangan dan penyediaan dataset wajah NIR yang akan sangat bermanfaat bagi komunitas riset dan pengembangan di bidang pengenalan wajah.

1.6 Batasan Masalah

Adapun batasan masalah penelitian ini adalah:

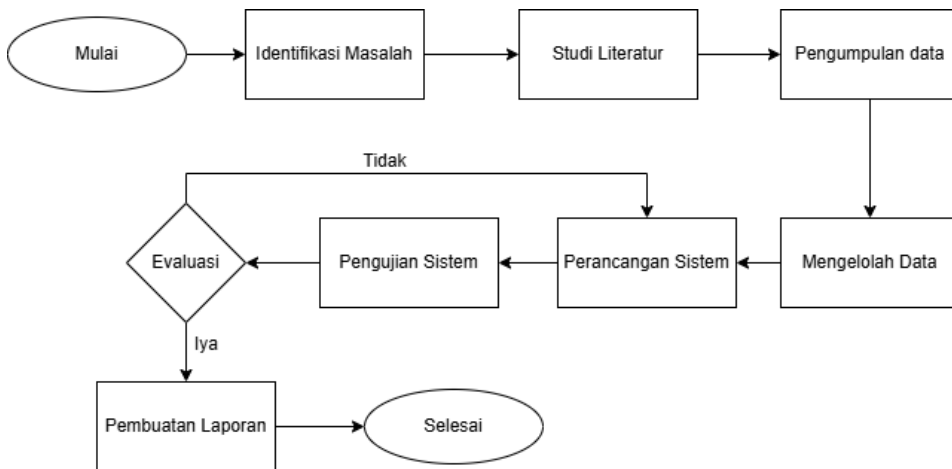
1. Jenis citra yang digunakan adalah citra dengan spektrum gelombang 850nm atau 0.85 μ m
2. Data akan diambil menggunakan kamera khusus NIR.
3. Jarak wajah dari kamera dengan 5 jarak berbeda (4, 8, 12, 16 dan 20 meter).

BAB II

METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini merupakan penelitian dengan jenis eksperimental. Pengumpulan data berupa studi literatur dengan dari jurnal dan artikel ilmiah sehingga ditemukan ruang lingkup masalah pada penelitian yang dilakukan, berikut disajikan tahapan penelitian:



Gambar 15 Tahapan Penelitian

2.1.1 Identifikasi Masalah

Dalam penelitian ini, terdapat tantangan dalam mengenali wajah jarak jauh dalam kondisi gelap. Pencahayaan yang rendah atau sama sekali tidak ada cahaya (gelap) dapat mengurangi kejelasan detail wajah, sehingga berdampak pada efektivitas analisis fitur. Selain itu, pemilihan fitur wajah yang kurang tepat dapat menyebabkan redundansi informasi dan menghambat kinerja algoritma pengenalan wajah. Oleh karena itu, diperlukan perbaikan pada kualitas citra untuk mengatasinya agar sistem dapat berjalan dengan maksimal.

2.1.2 Studi Literatur

Tahapan awal yang dilakukan adalah studi literatur. Tahapan studi literatur merupakan tahapan pengumpulan informasi dan referensi-referensi yang terkait dengan penelitian yang akan dilakukan mengenai pengenalan wajah dalam kondisi gelap baik jarak dekat maupun jarak jauh. Pada tahap ini juga peneliti menentukan metode yang akan digunakan yaitu metode CNN dengan arsitektur InceptionResnetV2.

2.1.3 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah data berupa foto diambil menggunakan kamera khusus NIR Apexel. Dataset ini berjumlah 5.514 yang didapatkan dari 10 koresponden yang masing-masing di video dari jarak dan posisi yang berbeda hasil video kemudian di ekstrak untuk menjadi gambar.

2.1.4 Mengelolah Data

Dilakukan pengolahan data citra wajah untuk memperoleh informasi visual yakni embeeding wajah. Proses ini mencakup beberapa Langkah pada tahap preprocessing menggunakan selective CLAHE, crop, resize dan split data

2.1.5 Perancangan Sistem

Tahapan perancangan sistem adalah tahap dimana dilakukan semua perancangan terhadap sistem yang nantinya akan dikembangkan serta dapat menyelesaikan masalah dalam mengenali wajah dalam kondisi jarak jauh dan gelap. Tahapan ini, sistem dirancang secara rinci dan mudah untuk dipahami agar mempermudah dalam proses implementasi ke dalam program.

2.1.6 Pengujian dan Analisis

Tahap pengujian dilakukan dengan menggunakan dataset uji terpisah untuk mengevaluasi kinerjanya. Setiap gambar dari dataset uji dianalisis untuk pengenalan wajah dari keseluruhan jarak dan setiap jarak yang berbeda. Hasilnya dievaluasi menggunakan metrik confusion matrix, precision, recall, dan F1-score untuk memvalidasi keandalan sistem dalam memberikan prediksi wajah yang dikenali.

2.1.7 Evaluasi

Evaluasi sistem ini dilakukan untuk mengukur sejauh mana sistem berhasil melakukan pengenalan wajah dalam kondisi gelap serta jaraknya.

2.1.8 Pembuatan Laporan

Bagian ini akan membahas hasil eksperimen dan analisis yang dilakukan, menginterpretasikan temuan-temuan yang ada, serta membandingkan hasil yang diperoleh dengan penelitian sebelumnya

2.2 Sumber Data

Sumber data memanfaatkan kamera inframerah termal tipe Apexel NV001, yang merekam citra dengan resolusi 1920 x 1080 piksel. Pengambilan data dilakukan pada lima variasi jarak antara subjek dan kamera, yaitu 4, 8, 12, 16, dan 20 meter, untuk memperoleh dataset yang beragam. Untuk menjamin kualitas, pengaturan fokus dan tingkat inframerah pada kamera dilakukan secara manual.



Gambar 16 Ilustrasi Pengambilan Gambar

Keseluruhan proses pengambilan data dilaksanakan dalam tiga sesi terpisah pada waktu yang berbeda. Data mentah berupa video dengan laju 20 frame per second (fps) kemudian diekstraksi menjadi gambar sekuensial dengan laju 5 fps. Langkah ini bertujuan untuk meminimalisir redundansi data dan mengurangi tingkat kemiripan antar-frame yang berurutan. Selain dari data dari kamera Apexel NV001, penulis juga mengambil data dari device berbeda yakni dari kamera CCTV dengan 5 jarak berbeda (1, 2, 3, 4 dan 5 meter) untuk perbandingan hasil dengan dataset dari kamera Apexel NV001.

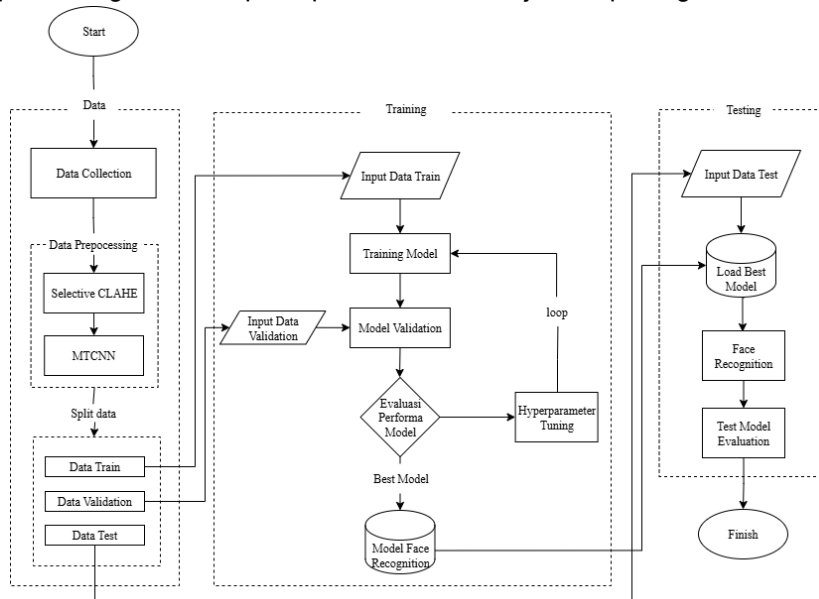
2.3 Perangkat Penelitian

Adapun perangkat-perangkat yang digunakan pada penelitian ini adalah sebagai berikut:

1. Perangkat Keras
 - a. Laptop Asus ROG Strix
 - b. Kamera Apexel NV001
 - c. Kamera CCTV
2. Perangkat Lunak
 - a. Sistem Operasi Linux Ubuntu LTS 22.04
 - b. Sistem Operasi Windows 11 64-Bit
 - c. Anaconda Python
 - d. Microsoft Office

2.4 Perancangan Sistem

Alur perancangan sistem pada penelitian ini ditunjukkan pada gambar berikut:



Gambar 17 Alur Perancangan Sistem

2.4.1 Data

1. Data Collection

Pengumpulan data pada penelitian ini dilakukan dengan cara merekam video sampel penelitian sebanyak 10 orang. Proses perekaman video dilakukan dalam kondisi gelap. Video direkam dengan durasi 10 hingga 15 detik setiap jarak yang telah ditentukan. Setelah proses perekaman selesai, video yang diperoleh diekstrak menjadi gambar (*frame*). Setiap 5 frame diambil setiap 1 detik, hasil ekstraksi disimpan dalam format .jpg. proses pengumpulan data menghasilkan jumlah data sebanyak 5.514 gambar.



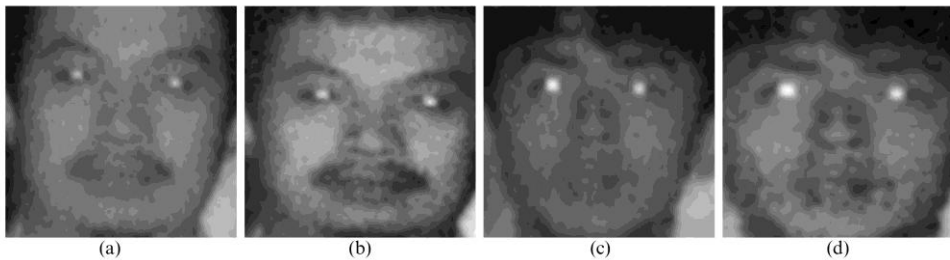
Gambar 18 Hasil Ekstrak Video Menjadi Gambar

2. Data Preprocessing

Setelah proses pengumpulan data dilakukan, dilanjutkan ke tahap pra-pemrosesan (*preprocessing*), untuk mempersiapkan gambar sebelum diproses lebih lanjut dengan tujuan untuk memperoleh hasil yang optimal. Tahapan dalam proses ini meliputi:

a. *Selective Contrast Limited Adaptive Histogram Equalization (CLAHE)*

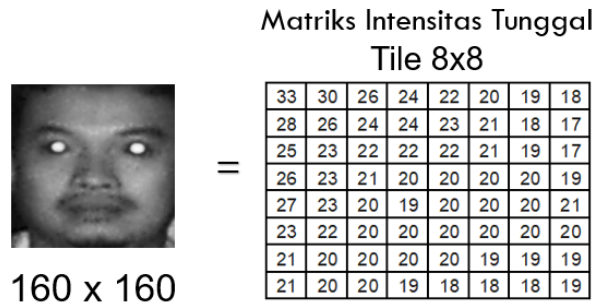
Contrast Limited Adaptive Histogram Equalization (CLAHE) adalah teknik peningkatan kontras gambar yang efektif, terutama untuk gambar dengan kondisi pencahayaan rendah atau kontras yang tidak merata. Teknik ini bekerja dengan membagi gambar menjadi blok-blok kecil (disebut *tile*) dan kemudian melakukan ekualisasi histogram secara lokal pada setiap *tile* (Paul dan Aslan, 2021). Pada penelitian ini, penulis menggunakan CLAHE Selektif, di mana CLAHE baru akan diterapkan pada citra ketika jarak akuisisi adalah 16 dan 20 meter. Keputusan ini didasarkan pada observasi bahwa pada jarak tersebut, pengenalan wajah menjadi semakin sulit dideteksi karena penurunan kualitas citra, sebuah pendekatan yang sejalan dengan penelitian yang menyoroti dampak kualitas citra pada kinerja pengenalan (Zhang, 2023). Gambar 19 dalam menunjukkan perbandingan visual citra wajah sebelum dan sesudah penerapan CLAHE.



Gambar 19 (a) citra wajah jarak 16 m original; (b) citra wajah jarak 16 m CLAHE; (c) citra wajah jarak 20 m original; (d) citra wajah jarak 20 m CLAHE

Adapun alur kerja algoritma *CLAHE* dijabarkan sebagai berikut:

- 1) Citra wajah diinput kedalam algoritma dalam hal ini citra wajah 160x160 pixel untuk melihat metriknya berupa *Gray Scale Level* dimana tiap kolom mewakili dari 1 dot pixel.



Gambar 20 *Gray Scale Level* pada Citra NIR

- 2) Tahapan selanjutnya yakni menghitung nilai Intensitas (I) dan Probabilitas (P) pada masing-masing channel warna. Tabel 2 menyajikan propabilitas data dari nilai metrik dimana nilai I = 20 paling sering muncul sebanyak 21 kali.

Tabel 2 Nilai I dan P *Channel Red*

I	P
17	2
18	5
19	9
20	21
21	6
22	5
23	5
24	3
25	1
26	3

I	P
27	1
28	1
30	1
33	1
Total	64

- 3) Selanjutnya menentukan nilai *Clip Limit* dalam studi ini, penulis menggunakan nilai *Clip Limit* = 2 untuk *grid size* = 8x8. Jika $P > \text{clip limit}$, maka kelebihan dari nilai P akan dibagi ke semua Intensitas yang memiliki nilai $P < \text{clip limit}$. I yang memiliki nilai $P > \text{Clip Limit}$ akan dikurangi dengan nilai clip limit.

I=18 (P=5): Kelebihan 5 - 2 = 3

I=19 (P=9): Kelebihan 9 - 2 = 7

I=20 (P=21): Kelebihan 21 - 2 = 19

I=21 (P=6): Kelebihan 6 - 2 = 4

I=22 (P=5): Kelebihan 5 - 2 = 3

I=23 (P=5): Kelebihan 5 - 2 = 3

I=24 (P=3): Kelebihan 3 - 2 = 1

I=26 (P=3): Kelebihan 3 - 2 = 1

Total Kelebihan: 3 + 7 + 19 + 4 + 3 + 3 + 1 + 1 = **41**

I=25 (P=1)

I=27 (P=1)

I=28 (P=1)

I=30 (P=1)

I=33 (P=1)

Jumlah Pembagi: **5**

Nilai Redistribusi: $\frac{41}{5} = 8,2$

- 4) Tahapan selanjutnya yaitu menentukan nilai *Cumulative Distribution Function* (CDF) dimana Nilai $P > 2$ diatur menjadi 2; Nilai $P < 2$ ditambahkan nilai redistribusi (8.2); dan Nilai $P = 2$ nilainya tetap. Sehingga nilai CDF dapat dihitung pada tabel 3 berikut.

Tabel 3 Nilai *Cumulative Distribution Function*

I	P (Awal)	P (Baru)	CDF
17	2	2	2
18	5	2	2 + 2 = 4
19	9	2	4 + 2 = 6
20	21	2	6 + 2 = 8
21	6	2	8 + 2 = 10
22	5	2	10 + 2 = 12

23	5	2	$12 + 2 = \mathbf{14}$
24	3	2	$14 + 2 = \mathbf{16}$
25	1	$1 + 8.2 = \mathbf{9.2}$	$16 + 9.2 = \mathbf{25.2}$
26	3	2	$25.2 + 2 = \mathbf{27.2}$
27	1	$1 + 8.2 = \mathbf{9.2}$	$27.2 + 9.2 = \mathbf{36.4}$
28	1	$1 + 8.2 = \mathbf{9.2}$	$36.4 + 9.2 = \mathbf{45.6}$
30	1	$1 + 8.2 = \mathbf{9.2}$	$45.6 + 9.2 = \mathbf{54.8}$
33	1	$1 + 8.2 = \mathbf{9.2}$	$54.8 + 9.2 = \mathbf{64}$

- 5) Setelah nilai CDF didapatkan, selanjutnya nilai tersebut akan dinormalisasikan sebelum diterapkan pada citra wajah yang akan diterapkan *CLAHE* dengan formula *CDF Normalized*.

$$CDF_{normalized} = \left(\frac{CDF - \min(CDF)}{\max(CDF) - \min(CDF)} \right) \times 255 \quad (8)$$

dimana:

CDF : Nilai *Cumulative Distribution Function*

Min(CDF) = Nilai terkecil dari CDF

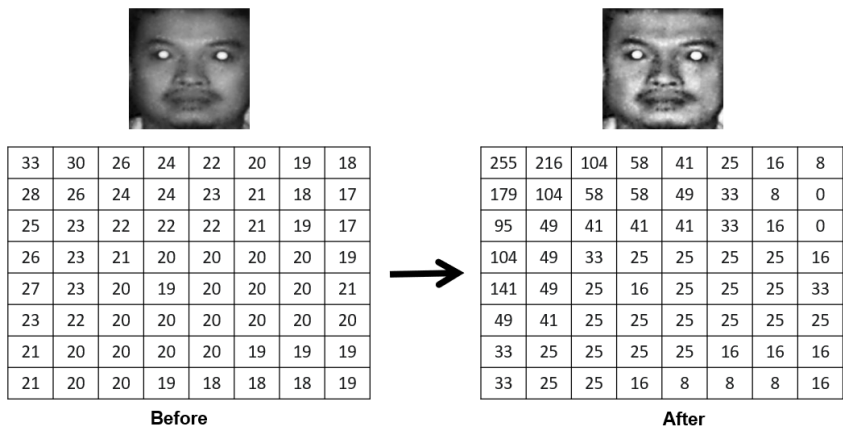
Max(CDF) = Nilai terbesar dari CDF

Semua nilai CDF dinormalisasi sehingga menghasilkan nilai Intensitas (I) yang baru. Proses ini memetakan ulang nilai intensitas lama ke nilai baru.

Tabel 4 Hasil Normalisasi

I (Lama)	CDF	CDF Normalisasi	I (Baru)
17	2	0	0
18	4	8,226	8
19	6	16,452	16
20	8	24,677	25
21	10	32,903	33
22	12	41,129	41
23	14	49,355	49
24	16	57,581	58
25	25,2	95,419	95
26	27,2	103,645	104
27	36,4	141,226	141
28	45,6	178,806	179
30	54,8	216,387	216
33	64	255	255

- 6) Setelah mendapatkan nilai Intensitas yang baru, sebagai contoh Piksel apa pun di tile 8x8 yang memiliki nilai 17 akan diubah menjadi 0. Piksel apa pun yang memiliki nilai 20 akan diubah menjadi 25. Piksel apa pun yang memiliki nilai 33 akan diubah menjadi 255. Gambar 21 memperlihatkan nilai metrik sebelum dan sesudah penerapan *CLAHE*.



Gambar 21 Nilai Metrik sebelum dan sesudah penerapan CLAHE

b. Multi-Task Convolutional Neural Network (MTCNN)

Penerapan MTCNN diawali dengan memuat satu per satu citra dari dataset, kemudian memanggil modul detektor yang akan mengembalikan koordinat bounding box (x, y, width, height) untuk setiap wajah yang terdeteksi. Jika sebuah citra mengandung lebih dari satu wajah, hanya satu wajah utama yang diambil berdasarkan ukuran bounding box terbesar, untuk menjaga konsistensi bahwa setiap citra wajah mewakili satu subjek. Setelah wajah berhasil dideteksi maka selanjutnya wajah akan dipotong dengan ukuran 160x160 pixel dan menyimpannya ke direktori baru untuk proses training dan validasi. Gambar 22 adalah contoh citra data yang dihasilkan setelah proses MTCNN



Gambar 22 Contoh citra wajah berdasarkan jarak yang telah di crop

Setelah itu, penulis melakukan pembersihan dataset dikarenakan terdapat beberapa citra wajah yang salah deteksi oleh detektor MTCNN sehingga pada tahap ini data berkurang 23,78% yang tidak berhasil dideteksi oleh MTCNN. Pada tabel 5, penulis juga membandingkan beberapa *detector* untuk melihat detector dengan persentase tertinggi.

Tabel 5. Perbandingan *Detector*

No.	Detector	Citra Berhasil Dideteksi	
		LWF Dataset (Serengil <i>et al</i> , 2024)	NIR Dataset
1	SSD	88.9%	54.04%
2	OpenCV	88.2%	63.42%
3	MTCNN	95.2%	65.02%
4	MTCNN CLAHE	-	76.22%

Sejalan dengan penelitian (Serengil *et al*, 2024) dimana *detector* yang digunakan dengan persentase tertinggi yakni MTCNN namun pada penelitian ini, penulis menambahkan *Selective CLAHE* pada MTCNN pada jarak 16 dan 20 meter agar citra wajah pada jarak ini banyak yang terdeteksi. Gambar 23 menunjukkan data yang gagal dideteksi oleh MTCNN namun berhasil dideteksi ketika menerapkan *Selective CLAHE* pada jarak tersebut.



Gambar 23 (a) jarak 16 meter, (b) jarak 20 meter

c. Split Data

Selanjutnya dilakukan split data untuk membagi dataset wajah ke dalam tiga bagian, yaitu data training, data validation, dan data testing. Pembagian data ini bertujuan untuk melatih, memvalidasi, dan menguji kinerja model pengenalan wajah nantinya. Adapun pembagian data dilakukan dengan proporsi sebagai berikut:

1) Data Training

Data training digunakan untuk proses melatih model dalam mengenali dan mempelajari embedding wajah berdasarkan 10 kelas individu yang berbeda dengan data training yang digunakan sebanyak 2.170 citra wajah.

2) Data Validation

Data validation digunakan untuk melakukan evaluasi performa model secara sementara pada saat proses pelatihan sedang berlangsung. Data ini berfungsi untuk memantau nilai loss dan separation margin pada setiap epoch, serta untuk mencegah terjadinya overfitting atau underfitting. Pada penelitian ini, data validation dialokasikan sebanyak 707 citra wajah.

3) Data Testing

Data testing digunakan untuk menguji kinerja akhir model setelah proses pelatihan selesai. Data ini bersifat independen, karena tidak dilibatkan dalam proses pelatihan maupun validasi. Data testing yang digunakan dalam penelitian ini merupakan citra wajah tiap kelas individu. Data testing pada penelitian ini dibagi menjadi dua kategori yakni keseluruhan data testing dan data testing untuk tiap jarak berbeda. Data testing untuk setiap kelas individu dapat dilihat pada tabel 6. Total data testing yang digunakan sebanyak 1.267 citra wajah.

Tabel 6. Pembagian data testing berdasarkan jarak

Kelas	4 m	8 m	12 m	16 m	20 m	Total
Amri	23	27	24	27	20	121
Erwin	33	34	29	35	30	161
Fifi	19	21	26	33	30	129
Inna	17	19	19	22	10	87
Iska	21	21	21	22	18	103
Niar	16	18	16	15	11	76
Raiz	38	36	36	34	36	180
Syahid	30	32	28	32	30	152
Tina	21	22	21	19	16	99
Wahyu	35	34	33	30	27	159
TOTAL	253	264	253	269	228	1267

2.4.2 Training

1. Input Data Train

Setelah melalui preprosesing data, data training yang berjumlah 2.170 digunakan untuk melatih model. Data training ini menjadi dasar bagi model dalam menyesuaikan bobot atau *weight*, sehingga mampu melakukan pengenalan dan prediksi secara akurat terhadap citra wajah yang belum pernah dilihat sebelumnya.

2. Training Model CNN

Convolutional Neural Network (CNN) dengan arsitektur *InceptionResNetV2* dipilih untuk training model untuk ekstraksi fitur karena memiliki keseimbangan yang baik antara kedalaman jaringan dan efisiensi komputasi. Arsitektur ini menggabungkan efisiensi dari blok Inception dengan stabilitas koneksi residual (*residual connections*). Kombinasi ini memungkinkan pembelajaran fitur yang lebih dalam dan kompleks sambil tetap menjaga kestabilan proses pelatihan. *InceptionResNetV2* mampu menangkap informasi spasial dari berbagai skala dan menggunakan koneksi pintasan untuk menghindari penurunan performa saat jaringan menjadi lebih dalam. Pendekatan ini telah terbukti efektif dalam menangani tugas klasifikasi dan embedding, termasuk dalam pengenalan wajah,

sehingga dimanfaatkan untuk menghasilkan vektor representasi dari citra wajah dalam penelitian ini.

a. *Augmentasi Data Train*

Untuk meningkatkan keandalan model dan memperbesar ukuran dataset secara efektif, data dari himpunan *training* melalui proses augmentasi dinamis (*on-the-fly*). Teknik yang diterapkan meliputi *RandomFlip (horizontal)*, *RandomBrightness* (faktor 0.2), *RandomContrast* (faktor 0.2), dan *RandomRotation* (faktor 0.05). Penerapan augmentasi secara dinamis ini memastikan model melihat versi citra yang sedikit berbeda pada setiap epoch, yang mendorongnya untuk belajar fitur-fitur esensial yang invarian terhadap variasi pose dan pencahayaan (Shorten & Khoshgoftaar, 2019).

b. *Generate Triplet Dinamis (Online Semi-Hard Negative Mining)*

Input sebenarnya untuk model Siamese bukanlah citra individual, melainkan sebuah set triplet (*Anchor, Positive, Negative*). Efektivitas pelatihan sangat bergantung pada pemilihan triplet yang informatif. Untuk itu, penelitian ini mengimplementasikan strategi *Online Semi-Hard Negative Mining*. Disebut "*online*" karena proses *mining* dilakukan secara dinamis pada setiap awal epoch berdasarkan keadaan model saat itu. Proses ini secara cerdas memilih sampel-sampel yang paling menantang namun tetap valid untuk kondisi model saat itu. Secara spesifik, untuk setiap pasangan *Anchor-Positive*, dicarikan sampel *Negative* yang memenuhi kriteria semi-hard:

$$d(f(A), f(P))^2 < d(f(A), f(N))^2 < d(f(A), f(P))^2 + \alpha \quad (9)$$

dimana:

A : Anchor

P : Positive

N : Negative

f(...): Siamese Network

d: jarak (*distance*)

α (alpha): Margin

Strategi ini menghindari triplet yang terlalu mudah (yang tidak menghasilkan gradien) dan triplet yang terlalu sulit (yang dapat menyebabkan ketidakstabilan), sehingga proses pembelajaran menjadi jauh lebih efisien dan efektif (Xuan, Souvenir, & Pless, 2020). Triplet yang dihasilkan inilah yang menjadi "Input Data Train" yang sebenarnya untuk setiap siklus pelatihan.

c. *Generate Quadruplets Citra*

Metodologi *Knowledge Distillation* telah diadaptasi secara luas untuk mengatasi tantangan *cross-domain*, seperti perbedaan kualitas gambar pada *face recognition*. Pada penelitian ini model menunjukkan penurunan kinerja yang signifikan ketika dihadapkan pada gambar berkualitas rendah, seperti gambar

buram, resolusi rendah, atau gambar dari jarak jauh, dibandingkan dengan gambar berkualitas tinggi. Untuk mengatasi kesenjangan performa ini, *Knowledge Distillation* digunakan untuk mentransfer pengetahuan dari citra *high resolution* (sebagai guru) ke domain citra *low resolution* (sebagai murid), pendekatan ini fokus pada pencocokan representasi fitur atau *embedding* di lapisan-lapisan dalam. Dengan melatih model murid untuk menghasilkan *embedding* fitur yang semirip mungkin dengan *embedding* yang dihasilkan oleh model guru dari identitas yang sama, model murid dipaksa untuk mempelajari representasi yang lebih baik dan diskriminatif.

Setiap quadruplet terdiri dari (1) Anchor Murid (*Student Anchor*) adalah citra wajah berkualitas rendah dari satu kelas individu sebagai contoh kelas Amri pada jarak 16 meter; (2) Positive Murid (*Student Positive*) adalah citra wajah berkualitas rendah lainnya dari identitas yang sama sebagai contoh kelas Amri pada jarak 20 meter; (3) Negative Murid (*Student Negative*) adalah citra wajah dengan kualitas yang lain dari identitas yang berbeda sebagai contoh kelas Raiz pada jarak 8 meter; dan (4) Anchor Guru (*Teacher Anchor*) adalah citra wajah berkualitas tinggi dari identitas yang sama dengan Anchor Murid sebagai contoh kelas "Amri" pada jarak 4 meter.

d. Arsitektur Model dan *Loss Function*

Pada Arsitektur Siamese Network dimana jaringan ini menggunakan tiga jalur *input* paralel (*Anchor*, *Positive*, *Negative*) yang diproses oleh satu encoder yang sama dengan bobot bersama (*shared weights*). Penggunaan bobot bersama memastikan bahwa semua citra diproyeksikan ke dalam ruang *embedding* yang sama, sehingga perbandingan jarak di antara mereka menjadi valid.

- 1) Encoder (*InceptionResNetV2*): Encoder yang digunakan adalah arsitektur *InceptionResNetV2*, yang diinisialisasi menggunakan bobot pra-terlatih dari *FaceNet* (Schroff, Kalenichenko, & Philbin, 2015). Ini merupakan implementasi dari strategi transfer learning, yang memungkinkan model memanfaatkan fitur-fitur visual kaya yang telah dipelajari dari dataset berskala besar. Untuk mengadaptasi model ke dataset penelitian, strategi *fine-tuning* diterapkan dengan hanya melatih ulang sebagian lapisan teratas dari model.
- 2) *Triplet Loss*: Model dilatih untuk mengoptimalkan *Triplet Loss*. Fungsi *Loss* ini secara matematis memaksa model untuk meminimalkan jarak antara pasangan *Anchor-Positive* dan memaksimalkan jarak antara pasangan *Anchor-Negative* dalam ruang *embedding*, yang dipisahkan oleh sebuah margin α .
- 3) *Distillation Loss* : *Loss* ini dihitung sebagai *Mean Squared Error* (MSE) antara *embedding* Anchor Murid dan *embedding* Anchor Guru yang bertujuan untuk memaksa *embedding* yang dihasilkan dari gambar

berkualitas rendah (Murid) agar menjadi semirip mungkin dengan embedding yang dihasilkan dari gambar berkualitas tinggi (Guru).

- 4) *Combined Loss* : Loss akhir yang digunakan untuk mengoptimalkan model adalah gabungan dari kedua loss tersebut, yang dibobot oleh hyperparameter alpha (α).

$$L_{Total} = (\alpha * L_{Triplet}) + ((1.0 - \alpha) * L_{Distillation}) \quad (10)$$

- 5) Metrik Jarak (Jarak Kosinus): Untuk mengukur kemiripan antar vektor *embedding*, digunakan Jarak Kosinus. Metrik ini dipilih karena keunggulannya dalam ruang berdimensi tinggi yang hanya mengukur sudut (orientasi) antar vektor, sehingga lebih tahan terhadap variasi kecerahan citra yang dapat mengubah magnitudo *embedding* (Wang, Wang, & Tao, 2020).

3. Siklus Pelatihan dan Validasi

Pada tahap ini model akan melakukan pelatihan (training) serta memvalidasi model.

- 1) Pelatihan Model: Model dilatih selama satu *epoch* menggunakan *batch* data *triplet* yang telah dihasilkan. Selama proses ini, gradien dari *triplet loss* dihitung dan digunakan untuk memperbarui bobot lapisan-lapisan yang tidak dibekukan (*unfrozen*) pada *encoder*.
- 2) Validasi Model: Setelah satu *epoch* pelatihan selesai, model dievaluasi pada himpunan validasi. Proses ini sesuai dengan blok *Model Validation*. Pada tahap ini, model tidak lagi belajar; bobotnya tetap dan hanya digunakan untuk melakukan inferensi pada data validasi.

4. Evaluasi Performa Model

Metrik utama yang digunakan untuk evaluasi adalah margin separasi (*val_separation_margin*), yaitu selisih rata-rata antara jarak *anchor-negative* dan *anchor-positive* pada himpunan validasi. Metrik ini memberikan ukuran yang lebih intuitif dan langsung mengenai seberapa baik model mampu memisahkan antara identitas yang berbeda, dibandingkan dengan nilai loss yang abstrak.

5. Hyperparameter Tuning dan Kontrol Pelatihan

Hasil evaluasi performa digunakan untuk mengontrol jalannya proses pelatihan secara otomatis. Secara sistematis sistem akan melakukan proses berikut:

- 1) Grid Search: Keseluruhan proses pelatihan dijalankan beberapa kali dengan kombinasi hiperparameter yang berbeda (*learning_rate*, *margin*, *trainable_layer*). Setiap eksekusi ini merupakan satu iterasi dari

hyperparameter tuning. Tabel 7 menunjukkan *hyperparameter tuning* yang penulis gunakan pada penelitian ini.

Tabel 7. *Hyperparameter Tuning*

No.	Trainable Layer	Margin	Learning Rate
1.	Block8_6_Branch_0_Conv2d_1x1	[1.0, 1.2, 1.4]	0.0001
2.	Block8_1_Branch_0_Conv2d_1x1	[1.0, 1.2, 1.4]	0.0001

- 2) ReduceLROnPlateau: Jika metrik *val_separation_margin* tidak meningkat setelah beberapa *epoch*, *learning rate* akan diturunkan secara otomatis. Ini adalah bentuk penalaan mikro di dalam satu sesi eksperimen untuk membantu model keluar dari plato lokal yaitu kondisi saat perbaikan kinerja melambat secara signifikan.
- 3) EarlyStopping: Jika performa validasi tidak membaik untuk periode yang ditentukan, training untuk konfigurasi hiperparameter tersebut akan dihentikan untuk efisiensi dan mencegah *overfitting*.

6. Seleksi dan Penyimpanan Model Terbaik

Selama proses *looping* ini, mekanisme *ModelCheckpoint* terus memantau *val_separation_margin*. Bobot model akan disimpan setiap kali nilai metrik ini mencapai rekor baru. Setelah serangkaian eksperimen dilakukan untuk mengevaluasi dampak kombinasi *hyperparameter* terhadap kinerja model. Metrik utama yang digunakan untuk perbandingan adalah *Area Under the Curve* (AUC) dan *Equal Error Rate* (EER), karena metrik ini secara langsung merefleksikan kemampuan model dalam memisahkan kelas dalam ruang fitur. Pada penelitian ini, penulis akan menggunakan model yang memiliki EER terendah sebagai tolak ukur performa utama, sebuah pendekatan yang konsisten dengan penelitian sebelumnya (Tolosana *et al*, 2020).

2.4.3 Testing

Setelah tahap pelatihan selesai dan model *encoder* dengan performa terbaik telah terseleksi, tahap selanjutnya adalah pengujian komprehensif untuk mengevaluasi performa dan keandalan model pada data yang sepenuhnya baru (*unseen data*).

1. Input Data Test

Kumpulan data yang diinput adalah data terpisah dan tidak pernah digunakan selama proses pelatihan atau validasi, yaitu Data Test. Kumpulan data ini berisi citra dari individu yang sama dengan yang ada di data latih, namun diambil dalam kondisi yang berbeda dalam penelitian ini, secara spesifik pada variasi jarak pengambilan gambar (4, 8, 12, 16 dan 20 meter). Penggunaan data yang terisolasi ini sangat krusial untuk mencegah evaluasi yang bias dan untuk secara objektif mengukur kemampuan model dalam menangani variasi yang umum terjadi, seperti perubahan resolusi dan detail wajah akibat jarak.

2. Memuat Model Terbaik

Tahapan inisialisasi pengujian dimulai dengan restorasi checkpoint model yang menunjukkan performa paling optimal selama fase validasi. Proses ini bertujuan untuk memastikan bahwa evaluasi dilakukan menggunakan instansi model dengan kapabilitas generalisasi terbaik pada data yang belum pernah ditemui sebelumnya. Pemilihan model tidak dilakukan secara arbiter, melainkan didasarkan pada kriteria kuantitatif, yaitu titik Equal Error Rate (EER) terendah yang tercatat dalam log pelatihan. Titik EER merepresentasikan tingkat kesalahan yang paling seimbang antara penolakan keliru (*false rejection*) dan penerimaan keliru (*false acceptance*), sehingga menjadi indikator yang andal untuk performa biometrik. Dengan memuat kembali model dari checkpoint optimal ini, arsitektur jaringan beserta seluruh bobot parameternya yang telah terkalibrasi dipulihkan secara presisi, sehingga menjamin bahwa proses pengujian yang dilakukan memiliki validitas metodologis yang tinggi.

3. Pengenalan Wajah Menggunakan *Convolutional Neural Network* (CNN) dalam Kondisi Gelap

Sistem pengenalan wajah yang dikembangkan dalam penelitian ini mengadopsi pendekatan pembelajaran mendalam (*deep learning*) yang berpusat pada paradigma pembelajaran metrik (*metric learning*). Pendekatan ini secara fundamental berbeda dari metode klasifikasi konvensional. Alih-alih melatih model untuk memprediksi label identitas secara langsung dari sebuah citra, tujuannya adalah untuk melatih sebuah *Convolutional Neural Network* (CNN) agar mampu memetakan citra wajah berdimensi tinggi ke dalam ruang vektor berdimensi rendah (*embedding space*). Di dalam ruang ini, kemiripan semantik antar wajah direpresentasikan secara geometris: vektor-vektor yang merepresentasikan wajah individu yang sama akan berada pada posisi yang berdekatan, sementara vektor-vektor dari individu yang berbeda akan terpisah secara signifikan.

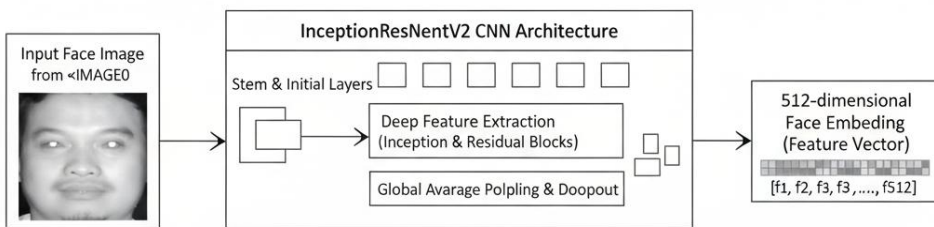
Metodologi ini sangat relevan untuk mengatasi tantangan utama dalam pengenalan wajah, seperti variasi intra-kelas yang tinggi (perbedaan pose, pencahayaan, ekspresi, dan penuaan pada individu yang sama) dan kemiripan antar-kelas yang tinggi. Alur kerja metodologis ini dapat diuraikan secara sistematis ke dalam tiga tahapan utama yakni:

a. Ekstraksi Fitur

Penelitian ini memanfaatkan arsitektur *InceptionResNetV2*, sebuah model CNN hibrida yang menggabungkan keunggulan dari dua arsitektur yakni modul *Inception* dan *Residual Connections*. Modul *Inception* memungkinkan jaringan untuk melakukan operasi konvolusi dengan berbagai ukuran filter secara paralel dalam satu lapisan. Kemampuan ini membuat model dapat menangkap pola visual dalam berbagai skala secara efisien, mulai dari detail halus hingga fitur wajah yang lebih global, tanpa meningkatkan beban komputasi secara signifikan. Koneksi Residual (*ResNet*) mengatasi masalah degradasi akurasi dan *vanishing gradient* yang sering terjadi pada jaringan yang sangat dalam. Dengan

menyediakan jalur pintas yang memungkinkan gradien mengalir langsung ke lapisan-lapisan awal, koneksi residual memastikan bahwa pelatihan jaringan yang sangat dalam tetap efektif dan konvergen. Kombinasi ini menghasilkan sebuah arsitektur yang sangat dalam dan kuat, mampu mempelajari representasi fitur yang kaya dan hierarkis dari data wajah. Proses ini mengubah citra masukan, yang merupakan matriks piksel, menjadi sebuah vektor fitur atau embedding dengan dimensi yang telah ditentukan (512 dimensi), yang secara matematis merepresentasikan identitas wajah tersebut.

Gambar 25 menunjukkan alur pemrosesan untuk mentransformasi citra wajah menjadi representasi fitur numerik melalui arsitektur *InceptionResNetV2*. Secara konseptual, citra diproses melalui serangkaian lapisan konvolusional secara hierarkis, di mana fitur-fitur visual mulai dari tingkat rendah (tepi dan tekstur) hingga tingkat tinggi (komponen wajah) diekstraksi dan diabstraksi. Puncak dari proses ini adalah tahap agregasi fitur, yang meringkas seluruh informasi spasial yang relevan menjadi sebuah vektor fitur berdimensi 512. Vektor keluaran ini yang disebut *embedding*, berfungsi sebagai proksi matematis yang padat dan diskriminatif dari citra asli, yang kemudian siap digunakan untuk analisis kuantitatif seperti pengukuran jarak kemiripan dalam tugas identifikasi.



Gambar 24 Ekstraksi Fitur Wajah

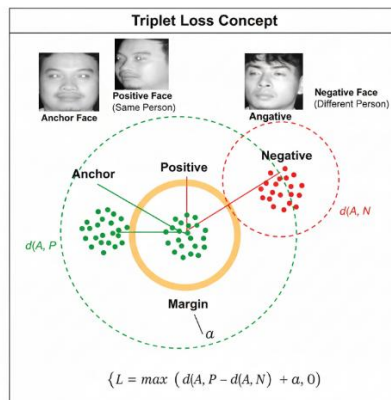
b. Optimalisasi *Embedding Space* Melalui Paradigma *Triplet Loss*

Setelah arsitektur didefinisikan, tahapan selanjutnya adalah melatihnya untuk menghasilkan embedding yang terorganisir dan diskriminatif menggunakan Triplet Loss, yang secara eksplisit dirancang untuk tugas pembelajaran metrik. Pelatihan dengan Triplet Loss memerlukan masukan berupa satu set data yang terdiri dari tiga sampel, yang disebut triplet, yang dipilih secara strategis dari himpunan data pelatihan yakni *Anchor* (A) adalah Citra individu, *Positive* (P) adalah citra lain dari individu yang sama dengan Anchor, dan *Negative* (N) adalah citra dari individu yang berbeda.

Tujuan dari *loss function* ini adalah untuk secara iteratif menyesuaikan bobot parameter dalam CNN sehingga jarak antara pasangan *Anchor-Positive* menjadi lebih kecil dibandingkan jarak antara pasangan *Anchor-Negative*, dengan selisih minimal sebesar margin (α). Margin ini mendorong separasi yang

lebih jelas antar kelas individu. Gambar 25 merepresentasikan konsep fundamental dari **Triplet Loss** dalam sebuah ruang fitur (*embedding space*).

Anchor dan *Positive*, yang merupakan citra dari individu yang sama, dipetakan ke titik-titik yang berdekatan secara geometris, mengindikasikan jarak yang kecil (kemiripan tinggi). Sebaliknya, *Negative*, yang merupakan citra dari individu berbeda, dipetakan ke titik yang jauh dari *Anchor*.



Gambar 25 Konsep *Triplet Loss*

c. Pengenalan Wajah pada Data Baru

Setelah proses pelatihan konvergen dan menghasilkan model *encoder* yang optimal, model tersebut siap digunakan untuk tugas identifikasi pada data yang belum pernah dilihat sebelumnya. Proses ini terbagi menjadi dua fase operasional yakni:

1) Fase 1 Pendaftaran (*Enrollment*)

Pada fase ini, sebuah basis data referensi dibangun. Untuk setiap individu yang identitasnya perlu dikenali, satu atau beberapa citra wajahnya diproses oleh model *encoder* yang telah dilatih untuk menghasilkan *embedding*. Untuk meningkatkan kehandalan *template biometric, embedding* dari beberapa citra dapat dirata-ratakan untuk menghasilkan satu vektor representatif tunggal yang lebih stabil terhadap variasi minor.

2) Fase 2 Verifikasi atau Identifikasi

Ketika sebuah citra uji (*probe*) baru disajikan ke sistem, citra tersebut diumpankan ke *encoder* untuk menghasilkan vektor *embedding*-nya. *Embedding* dari citra uji ini kemudian dibandingkan dengan setiap *embedding* representatif yang tersimpan di dalam basis data referensi. Perbandingan dilakukan dengan menghitung jarak kosinus (*cosine distance*) antara dua vektor *embedding*. Jarak kosinus dipilih karena lebih fokus pada orientasi (sudut) antar vektor daripada magnitudo, membuatnya lebih tahan terhadap variasi pencahayaan yang dapat

mengubah panjang vektor tanpa mengubah fitur wajah yang mendasarinya.

Hasil perhitungan jarak kosinus kemudian dibandingkan dengan nilai ambang batas (*threshold*) yang telah ditentukan sebelumnya. Jika jarak yang dihitung berada di bawah *threshold*, sistem akan menyimpulkan bahwa citra uji cocok dengan individu yang ada di basis data. Jika melebihi *threshold*, citra tersebut dianggap sebagai individu yang tidak dikenali (*unknown*).

2.5 Evaluasi Sistem

Untuk melihat hasil kinerja dari sistem pengenalan wajah dalam kondisi gelap yang dirancang pada masing-masing skenario maka digunakan evaluasi confusion matrix yang menggunakan 4 variabel yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Pada pengujian ini yakni skenario yang memiliki akurasi tertinggi merupakan skenario dengan algoritma terbaik untuk melakukan deteksi wajah dalam kondisi gelap pada penelitian ini. Penelitian ini melakukan pengujian akurasi untuk setiap huruf lontara yang ingin dikenali dan divalidasi menggunakan nilai akurasi, *precision*, *recall* dan *f1 score*.

Penentuan tabel confusion matrix dapat dilihat pada tabel 8 di bawah

Tabel 8 Tabel Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	<i>TP</i>	<i>FN</i>
	Negative	<i>FP</i>	<i>TN</i>

Fungsi dari penentuan tabel confusion matrix, adalah untuk mengevaluasi pembelajaran supervised learning, guna untuk memberi ukuran kinerja yang model algoritma yang dipilih. Berikut ukuran kinerja yang sering sangat sederhana namun efisien untuk digunakan:

- Accuracy*, ini adalah ukuran kinerja yang akan memberikan tingkat keakuratan dari keseluruhan model atau dalam penjelasan lain adalah menghitung semua prediksi yang benar dari total jumlah data. Rumus *Accuracy* adalah $(TP + TN) / \text{Total semua data}$.
- Precision*, adalah ukuran kinerja yang akan memberi informasi dari prediksi sebagai kelas positif yang sebenarnya positif. Kamu dapat menghitung dengan rumus $TP / (TP + FP)$.
- Sensitivity / Recall*, adalah ukuran kinerja yang memberi informasi dari prediksi kelas positif yang di prediksi negatif. Rumus nya adalah sebagai berikut $TP / (TP + FN)$.

2.6 Waktu dan Lokasi Penelitian

Penelitian ini dimulai pada bulan Oktober 2024 – Agustus 2025. Penelitian dilaksanakan di Laboratorium Artificial Intelligence and Multimedia Processing (AIMP), Departemen Teknik Informatika Universitas Hasanuddin, Jalan Poros Malino, Kelurahan Borongloe, Kecamatan Bontomarannu, Kabupaten Gowa, Provinsi Sulawesi Selatan.