

Implementasi Metode Convolutional Neural Network dalam Pengolahan Citra Digital Sidik Bibir

1st Irwansyah

School of Electrical Engineering

Telkom University

Bandung, Indonesia

syahirwan@student.telkomuniversity.ac.id

2nd Sofia Saidah

School of Electrical Engineering

Telkom University

Bandung, Indonesia

sofiasaidahsfi@telkomuniversity.ac.id

3rd Bambang Hidayat

School of Electrical Engineering

Telkom University

Bandung, Indonesia

bhidayat@telkomuniversity.ac.id

Abstrak — Penelitian ini membahas pengembangan sistem identifikasi individu berbasis sidik bibir menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2. Sidik bibir dipilih sebagai alternatif biometrik karena memiliki pola unik dan dapat digunakan baik pada individu hidup maupun korban, sehingga relevan dalam bidang forensik dan keamanan. Dataset diperoleh melalui pengambilan GAMBAR langsung terhadap sepuluh individu (total 200 citra), yang kemudian diproses melalui segmentasi U-Net, konversi grayscale, peningkatan kontras menggunakan CLAHE, ekstraksi fitur dengan Gabor filter, binarisasi adaptif, operasi morfologi, normalisasi, dan augmentasi data untuk meningkatkan kemampuan generalisasi. MobileNetV2 digunakan sebagai *feature extractor* dengan bobot awal ImageNet, diikuti *dense layer* tambahan untuk klasifikasi multikelas. Proses pelatihan dilakukan dengan optimasi *hyperparameter* meliputi augmentasi, *batch size*, *optimizer*, *learning rate*, dan jumlah *epoch*. Hasil pengujian menunjukkan konfigurasi terbaik pada *optimizer* Adam dengan *learning rate* 0,001 dan pelatihan selama 200 *epoch*, yang mampu mencapai akurasi, presisi, *recall*, dan F1-score sebesar 100% pada data uji. Sistem yang dikembangkan terbukti efektif dalam mengenali pola sidik bibir, menunjukkan potensi tinggi sebagai metode identifikasi biometrik alternatif. Pengembangan lebih lanjut direkomendasikan dengan memperluas variasi dataset dan menguji data untuk meningkatkan kemampuan generalisasi.

Kata kunci— CNN, MobileNetV2, Identifikasi Individu, Klasifikasi Biometrik, Pengolahan Citra, Sidik Bibir.

I. PENDAHULUAN

Konsep biometrik telah ada selama lebih dari satu abad, awalnya digunakan untuk identifikasi kriminal dan pada tahun 1924 mulai diterapkan juga untuk personel kepolisian[1]. Hingga kini, teknologi ini terus berkembang dengan berbagai modalitas seperti sidik jari, pengenalan wajah, iris, DNA, retina, pembuluh darah telapak tangan, dan bau tubuh[2]. Namun, setiap metode memiliki keterbatasan sidik jari dapat rusak atau hilang, pengenalan wajah dapat terganggu pencahayaan atau perubahan penampilan, dan pemindaian retina bisa dianggap rumit atau tidak nyaman [2].

Sidik bibir menawarkan alternatif yang lebih mudah diakses dan dapat dilakukan pada orang hidup maupun korban. Pola sidik bibir unik bagi setiap individu, sama seperti sidik jari, dan dapat digunakan sebagai metode identifikasi yang akurat dan personal.

Dalam bidang forensik, sidik bibir dapat menjadi metode tambahan ketika sidik jari atau wajah tidak dapat digunakan, seperti pada korban dengan luka di wajah atau tangan. Pola sidik bibir terbentuk sejak usia kandungan enam minggu dan bersifat permanen bahkan setelah kematian [2], sehingga memiliki potensi tinggi sebagai identifikasi biometrik. Meski demikian, faktor sosial, medis, dan biometrik perlu diperhatikan. Kegagalan dalam identifikasi korban dapat menimbulkan keresahan masyarakat, sedangkan kondisi fisik bibir seperti luka, penyakit, atau penuaan dapat memengaruhi pola sidik bibir dan akurasi identifikasi.

Metode identifikasi yang ada saat ini sidik jari, pengenalan wajah, retina, dan iris—masing-masing memiliki kelebihan dan kekurangan. Sidik jari akurat dan murah tetapi sensitif terhadap kerusakan fisik, pengenalan wajah cepat dan non-invasif namun rentan terhadap kondisi lingkungan, sedangkan retina dan iris memiliki akurasi tinggi tetapi mahal dan invasif [2]. Berbeda dengan itu, sidik bibir dapat menjadi biometrik pelengkap yang menawarkan keseimbangan antara aksesibilitas, akurasi, dan keunikan. Penelitian ini berfokus pada pengembangan sistem identifikasi berbasis sidik bibir yang efektif dan efisien, dengan penerapan pada bidang forensik, keamanan digital, dan verifikasi identitas berbasis biometrik.

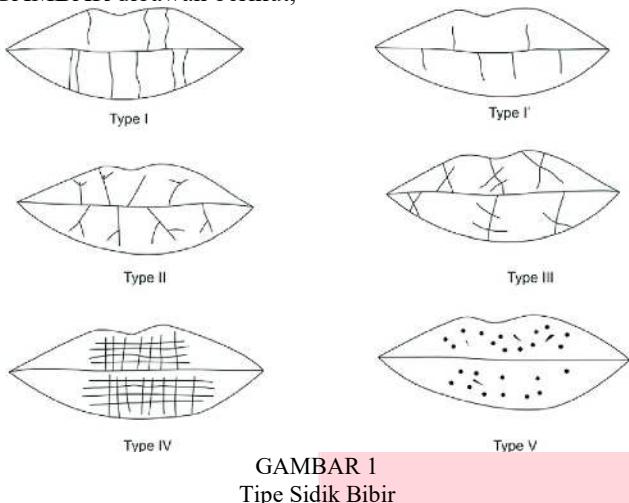
II. KAJIAN TEORI

Dalam bagian ini, menyajikan dan menjelaskan teori-teori yang berkaitan dengan variabel-variabel penelitian, yang berfokus pada sidik bibir, pengolahan citra digital, dan metode *deep learning* yang digunakan.

A. Sidik Bibir

Sidik bibir, atau dalam istilah forensik dikenal sebagai *cheiloscopy*, adalah ilmu yang mempelajari alur atau lekukan pada mukosa bibir manusia[3]. Pola sidik bibir ini bersifat unik dan tidak akan berubah sepanjang hidup, kecuali jika terjadi trauma atau penyakit pada bibir[3]. Penelitian oleh

Suzuki dan Tsuchihashi pada tahun 1970 juga mengukuhkan keunikan pola sidik bibir pada setiap individu[1]. Pola sidik bibir Suzuki dan Tsuchihashi dapat diklasifikasikan seperti GAMBAR dibawah berikut,



GAMBAR 1
Tipe Sidik Bibir

Pola sidik bibir dapat diklasifikasikan menjadi beberapa tipe utama berdasarkan bentuk alur garisnya. Tipe I adalah garis vertikal lurus penuh yang membentuk sepasang bibir, sedangkan Tipe I' merupakan garis vertikal lurus yang tidak penuh atau lebih pendek [4]. Tipe II ditandai dengan dua garis atau lebih yang saling bercabang, sementara Tipe III menunjukkan dua garis atau lebih yang saling menyilang. Tipe IV memiliki pola garis retikuler yang menyerupai jaring atau kisi-kisi, dan Tipe V adalah pola yang tidak beraturan atau tidak dapat dimasukkan ke dalam tipe lainnya [4].

B. Pengolahan Citra Digital

Pengolahan citra digital adalah bidang ilmu yang memanipulasi dan menganalisis citra digital dengan komputer[5]. Ini adalah langkah penting dalam sistem identifikasi sidik bibir. Tahapannya mencakup akuisisi citra, pra-pemrosesan (seperti konversi ke *grayscale*), ekstraksi fitur, dan pencocokan. Berikut pra-pemrosesan yang ada di pengolahan citra digital :

1. Citra RGB



GAMBAR 2
Contoh Citra RGB

Citra RGB merupakan citra warna yang memiliki 3 jumlah kanal, yaitu merah (*red*), hijau (*green*), dan biru (*blue*)[3]. Setiap komponen memiliki intensitas antara 0 hingga 255, dan kombinasi dari ketiganya dapat menghasilkan jutaan warna berbeda. Model ini banyak digunakan pada perangkat digital seperti monitor, kamera, dan ponsel.

2. Segmentasi Citra

Segmentasi Citra merupakan proses untuk memisahkan sebuah objek dari latar belakangnya yang kemudian menjadi *Region of Interest* (ROI), guna meningkatkan fokus dan akurasi model.[6]

3. Citra *grayscale*



GAMBAR 3
Contoh Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya mempunyai satu nilai untuk setiap pikselnya, dengan kata lain nilai *RED*=*GREEN*=*BLUE* yang memiliki warna hitam, keabuan dan putih[3]. Nilai intensitas setiap piksel berada dalam rentang 0–255, di mana 0 berarti hitam pekat dan 255 berarti putih murni [3].

4. Citra Biner



GAMBAR 4
Contoh Citra Biner

Citra biner adalah bentuk paling sederhana dari citra digital yang hanya memiliki dua kemungkinan nilai piksel. Nilai tersebut biasanya 0 (hitam) dan 1 atau 255 (putih), sehingga menghasilkan tampilan yang sangat kontras[3].

5. *Gabor Filter*

Gabor filter merupakan kombinasi dari fungsi gelombang sinusoidal dan fungsi *Gaussian*, teknik pemrosesan citra ini digunakan untuk mengekstraksi fitur tekstur berdasarkan orientasi dan frekuensi spasial [7]. Filter ini didesain menyerupai fungsi kerja sistem visual manusia yang sensitif terhadap garis, tepi, dan pola arah tertentu[7].

6. CLAHE

CLAHE (*Contrast Limited Adaptive Histogram Equalization*) merupakan teknik yang digunakan untuk meningkatkan kontras lokal dan menonjolkan detail halus pada citra[8].

7. Binarisasi Adaptif dan Operasi Morfologi

Binarisasi adaptif menghitung ambang berbeda untuk setiap wilayah kecil pada citra [8]. Setelah binarisasi, dilakukan operasi morfologi untuk membersihkan citra dari noise dan menyempurnakan

bentuk objek[5]. Proses ini mengubah citra grayscale menjadi citra biner (hitam-putih) untuk menonjolkan fitur tertentu dan membersihkan noise [5], [8].

C. Convolutional Neural Network (CNN)

CNN adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data berbentuk grid seperti GAMBAR dan video. CNN memiliki metode canggih yang berasal dari pengembangan *Multilayer Perceptron* (MLP) [9]. CNN dirancang secara khusus untuk memproses data dua dimensi, terutama GAMBAR. Karena strukturnya yang dalam, CNN diklasifikasikan sebagai anggota keluarga *Deep Neural Network* [9].

Pada tahap pembelajaran fitur, terdapat lapisan yang berfungsi untuk menerima gambar sebagai input langsung di awal, kemudian memproses data tersebut hingga menghasilkan keluaran berupa representasi numerik[10]. Proses ini melibatkan lapisan konvolusi dan *pooling*, dimana setiap lapisan akan memproduksi peta fitur (*feature maps*) yang mengGAMBARkan karakteristik GAMBAR, lalu data ini diteruskan ke bagian lapisan klasifikasi[10]. Pada lapisan klasifikasi, terdapat beberapa lapisan yang terdiri dari neuron-neuron yang sepenuhnya saling terhubung (*fully connected*). Lapisan ini mendapatkan input dari hasil keluaran lapisan pembelajaran fitur yang telah melalui proses flattening, kemudian diproses melalui beberapa lapisan tersembunyi sebelum akhirnya menghasilkan output berupa akurasi klasifikasi untuk setiap kelas[10].

TABEL 1
(A)

Algoritma Proses Convolutional Neural Network	
<i>Feature Learning</i>	Lapisan Konvolusi
	Lapisan Aktivasi
	Lapisan <i>pooling</i>
<i>Classification</i>	Lapisan <i>Flatten</i>
	Lapisan <i>Fully Connected</i>
	Lapisan <i>Softmax</i>

1. Lapisan Konvolusi

Lapisan ini adalah fondasi utama CNN. Lapisan konvolusi merupakan sebuah filter berukuran kecil memindai seluruh GAMBAR untuk mendeteksi fitur-fitur dasar seperti garis dan tepi. Berikut adalah rumus umum dari operasi konvolusi yang ada di *convolutional neural network* :

$$Output = i = \sum_{i=1}^m \sum_{j=1}^n = (Input_{i,j} \times Filter_{i,j}) \quad (1)$$

Penjelasan :

- *Output* : Nilai tunggal pada *feature map* yang dihasilkan dari operasi konvolusi.
- Σ : Simbol sigma, yang berarti "penjumlahan".
- i dan j : Indeks untuk baris dan kolom pada matriks.
- m dan n : Dimensi (ukuran) filter. Dalam contoh ini, m=3 dan n=3

- $Input_{i,j}$: Nilai piksel pada matriks input (GAMBAR) yang berada pada baris ke-i dan kolom ke-j.
- $Filter_{i,j}$: Nilai pada matriks filter (kernel) yang berada pada baris ke-i dan kolom ke-j.

Berdasarkan gambar, operasi konvolusi adalah proses perkalian elemen per elemen antara filter dan bagian dari GAMBAR input, lalu semua hasilnya dijumlahkan. Misalnya, untuk mendapatkan nilai '6' pada feature map, filter 3x3 (matriks hijau) bergerak untuk menutupi matriks 3x3 pada GAMBAR input (matriks oranye) dengan nilai berikut:

Matriks Input: [[9, 2, 5], [2, 4, 0], [4, 5, 4]]

Filter: [[1, 0, -1], [1, 0, -1], [1, 0, -1]]

Perhitungannya adalah:

$$\begin{aligned} & (9 \times 1) + (2 \times 0) + (5 \times -1) + \\ & (2 \times 1) + (4 \times 0) + (0 \times -1) + \\ & (4 \times 1) + (5 \times 0) + (4 \times -1) \\ & = (9 + 0 - 5) + (2 + 0 - 0) + (4 + 0 - 4) \\ & = 4 + 2 + 0 = 6 \end{aligned}$$

Nilai 6 inilah yang kemudian menjadi salah satu elemen pada feature map. Proses ini diulangi untuk setiap area yang dipindai oleh filter.

2. Lapisan Aktivasi

Pada lapisan ini, terjadi proses pengubahan nilai nilai feature map pada jarak tertentu tergantung pada fungsi aktivasi yang dipakai, yaitu fungsi aktifasi ReLu[10]., fungsi aktivasi ReLU (*Rectified Linear Unit*) diterapkan pada *feature map*. Fungsi ini sangat sederhana namun penting, karena mengubah semua nilai negatif menjadi nol, sementara nilai positif tetap dipertahankan.

3. Lapisan *pooling*

Lapisan *pooling* berperan penting dalam menyederhanakan peta fitur yang dihasilkan oleh lapisan konvolusi[6]. Selain itu, *pooling* berfungsi untuk menyoroti fitur-fitur yang paling dominan, menjadikan pelatihan model menjadi lebih efisien. Dua jenis pooling yang umum digunakan adalah *max pooling* dan *average pooling*. *Max pooling* bekerja dengan memilih nilai maksimum dari setiap area yang dipindai oleh kernel, sedangkan *average pooling* mengambil nilai rata-rata dari area tersebut[6].

4. Lapisan *Flatten*

Lapisan *Flatten* berfungsi sebagai jembatan transisi antara tahap ekstraksi fitur (di mana data masih memiliki dimensi spasial) dan tahap klasifikasi (yang membutuhkan data linear). Proses *flatten* secara sederhana "membentangkan" semua piksel dari *feature map* 3D menjadi sebuah vektor satu dimensi yang sangat panjang. Tujuannya adalah untuk menyiapkan data agar bisa dimasukkan ke dalam lapisan *Fully Connected* berikutnya, yang secara tradisional hanya bisa menerima input satu dimensi.

5. Lapisan *Fully Connected*

Lapisan ini adalah inti dari tahap klasifikasi dan sering kali disebut sebagai lapisan tersembunyi (*hidden layer*) dalam jaringan saraf tiruan tradisional[6]. Lapisan ini memiliki kemampuan untuk mempelajari hubungan kompleks dan non-linear antar fitur, yang sangat krusial untuk membuat keputusan klasifikasi akhir.

6. Lapisan *Softmax*

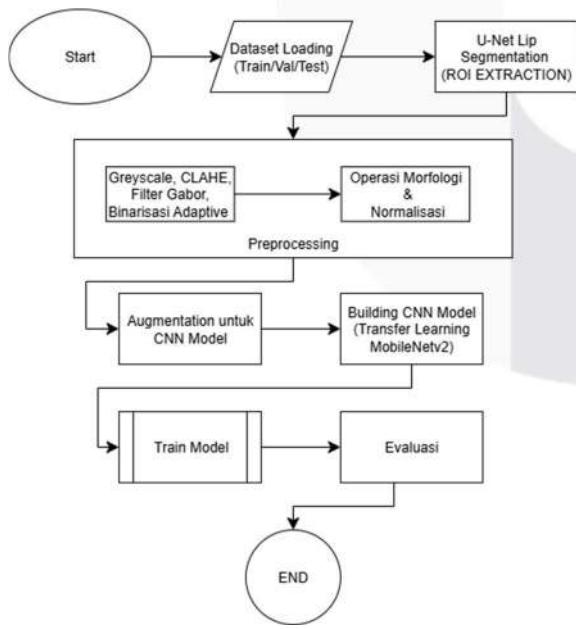
Lapisan *Softmax* adalah lapisan output terakhir untuk tugas klasifikasi multi-kelas[10]. Lapisan ini mengambil output numerik dari lapisan *Fully Connected* yang sering disebut sebagai *logits* dan mengubahnya menjadi distribusi probabilitas[10]. Proses *Softmax* menggunakan fungsi matematis yang mengubah setiap nilai *logit* menjadi nilai antara 0 dan 1, di mana total semua nilai tersebut akan selalu berjumlah 1[6]. Hal ini membuat outputnya sangat mudah diinterpretasikan.

D. MobileNetV2

MobileNetV2 adalah evolusi dari arsitektur CNN yang diciptakan untuk mengatasi keterbatasan sumber daya komputasi[9]. Dibandingkan arsitektur CNN konvensional, MobileNetV2 memiliki pendekatan unik dalam lapisan konvolusinya[9].

Model ini menggunakan teknik-teknik inovatif seperti *depthwise convolution* dan *pointwise convolution* untuk menghemat komputasi[9]. Fitur-fitur lain yang membuatnya sangat efisien adalah *linear bottlenecks* dan *shortcut connections*, yang membantu menjaga ukuran model tetap kecil sambil mempertahankan performa yang tinggi[9]. Kombinasi dari semua fitur ini menjadikan MobileNetV2 pilihan ideal untuk aplikasi yang berjalan pada perangkat dengan sumber daya terbatas, seperti ponsel pintar.

III. METODE



GAMBAR 5
Desain Model Algoritma CNN

Algoritma CNN dibangun dalam beberapa tahapan, dimulai dari pengumpulan dataset hingga evaluasi model.

Proses ini dirancang untuk menciptakan sistem identifikasi individu berdasarkan sidik bibir.

A. Pengumpulan dan Pemrosesan Dataset

Dataset penelitian ini diperoleh dari pengambilan GAMBAR langsung terhadap sepuluh individu, masing-masing difoto sebanyak dua puluh kali sehingga terkumpul 200 GAMBAR sebelum augmentasi. Pengambilan dilakukan dengan kamera 12 megapiksel berformat jpg, jarak 30 cm, variasi posisi (kanan, tengah, kiri), dan pencahayaan minimal 600 lux. Meskipun terkendali, citra mentah tetap memiliki noise, sedikit blur, dan variasi pencahayaan lokal. Struktur dataset diklasifikasikan berdasarkan nama individu dalam folder, kemudian dibagi menjadi 70% data pelatihan, 15% validasi, dan 15% pengujian menggunakan stratifikasi untuk menjaga keseimbangan kelas.

B. Preprocessing

Citra yang telah terkumpul melalui proses akuisisi dan pemuatannya selanjutnya menjalani serangkaian tahapan preprocessing untuk mempersiapkan data sebelum masuk ke tahap pemodelan. Proses dimulai dengan segmentasi menggunakan model U-Net yang berfungsi mendekripsi dan memisahkan area bibir sebagai *Region of Interest (ROI)*. ROI bibir yang telah dipotong kemudian dikonversi menjadi citra grayscale untuk menyederhanakan representasi data dan mengurangi kompleksitas komputasi. Selanjutnya, teknik *Contrast Limited Adaptive Histogram Equalization (CLAHE)* diterapkan pada citra grayscale guna meningkatkan kontras, sehingga detail halus pada pola sidik bibir lebih menonjol.

Setelah itu, filter Gabor digunakan untuk mengekstraksi fitur tekstur berdasarkan frekuensi dan orientasi spasial, yang penting untuk membedakan pola-pola unik pada citra bibir. Proses dilanjutkan dengan binarisasi adaptif yang diikuti operasi morfologi seperti *opening* dan *closing* untuk menghilangkan noise sekaligus memperbaiki bentuk objek. Tahap terakhir adalah pengubahan ukuran citra menjadi 224×224 piksel dan normalisasi nilai piksel ke rentang [-1,1], sehingga seluruh data memiliki skala yang seragam dan siap digunakan dalam proses pelatihan model.

C. Augmentasi Data

Untuk mengatasi keterbatasan jumlah data sekaligus meningkatkan kemampuan generalisasi model, digunakan teknik augmentasi data melalui *ImageDataGenerator* dari TensorFlow. Proses augmentasi ini diterapkan secara khusus pada data pelatihan melalui objek *augmented_train_generator* dengan menerapkan berbagai transformasi yang cukup agresif dan beragam. Beberapa teknik yang digunakan meliputi rotasi citra hingga 10 derajat, pergeseran lebar dan tinggi hingga 10% dari ukuran GAMBAR, transformasi shear sebesar 0.1, pergeseran saluran warna (*channel shift*) sebesar 0.1, serta penyesuaian tingkat kecerahan dalam rentang 0.7 hingga 1.3. Kombinasi transformasi ini dirancang untuk membuat model lebih tangguh terhadap berbagai kondisi pengambilan GAMBAR di dunia nyata, seperti perubahan pencahayaan atau sudut pandang.

D. Pembangunan Model CNN

Tahap selanjutnya adalah membangun dan melatih model CNN dengan menggunakan arsitektur MobileNetV2. Pemilihan arsitektur ini didasarkan pada efisiensinya yang tinggi dari sisi komputasi, namun tetap mampu menghasilkan akurasi yang baik, terutama untuk sistem yang nantinya akan diimplementasikan pada perangkat dengan keterbatasan sumber daya. MobileNetV2 dimanfaatkan sebagai *feature extractor* dengan bobot awal (*pre-trained*) dari dataset ImageNet, di mana lapisan dasarnya dibekukan (*layer.trainable = False*) agar bobot awal tetap terjaga. Bagian atas model dihilangkan (*include_top=False*) dan diganti dengan susunan *dense layer* tambahan yang terdiri dari *GlobalAveragePooling2D*, *Dense* dengan 512 neuron beraktivasi ReLU, diikuti *BatchNormalization* dan *Dropout* sebesar 0.1. Selanjutnya, ditambahkan *Dense* dengan 256 neuron beraktivasi ReLU sebagai *feature output layer*, disertai *BatchNormalization* dan *Dropout* sebesar 0.05. Lapisan terakhir adalah *Dense* dengan jumlah neuron sesuai *num_classes* dan fungsi aktivasi *softmax* untuk menghasilkan klasifikasi akhir multikelas. Selama pelatihan, digunakan *optimizer* Adam dengan *learning rate* sebesar 0.0001 serta fungsi *loss* *categorical_crossentropy* untuk mengoptimalkan proses pembelajaran.

E. Training dan Evaluasi

Setelah model selesai dilatih, tahap berikutnya adalah melakukan evaluasi performa terhadap data pengujian yang sebelumnya tidak pernah dilihat oleh model selama proses pelatihan. Evaluasi ini mencakup pengukuran nilai akurasi dan loss, di mana akurasi dihitung menggunakan *accuracy_score* untuk mengetahui tingkat keberhasilan klasifikasi. Selain itu, confusion matrix divisualisasikan dalam bentuk heatmap menggunakan seaborn guna mempermudah analisis kesalahan klasifikasi antar individu. Untuk memahami perkembangan pembelajaran, tren akurasi dan loss selama proses pelatihan dan validasi diplot dalam bentuk grafik, sehingga dapat dilihat sejauh mana model belajar secara stabil dari waktu ke waktu.

Setelah itu, performa model juga diukur menggunakan metrik tambahan seperti presisi, recall, dan F1-score, yang memberikan GAMBARan lebih komprehensif terkait kualitas prediksi model. Pengujian acak dilakukan pada beberapa GAMBAR uji untuk menilai kinerja secara visual, memastikan model mampu mengenali pola sesuai target klasifikasi. Model yang telah terlatih kemudian disimpan dengan nama *model_deteksi_adam_0.0002.h5* sebagai arsip dan dapat digunakan kembali tanpa perlu melakukan pelatihan ulang.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk memastikan sistem identifikasi biometrik berbasis sidik bibir memenuhi spesifikasi yang telah ditetapkan. Pengujian ini mencakup verifikasi kualitas akuisisi citra, akurasi identifikasi, serta kualitas pencahayaan. Hasil pengujian disajikan secara naratif dan didukung dengan analisis untuk memperoleh pemahaman yang komprehensif terhadap performa sistem.

Pada tahap pengujian, digunakan skema awal dengan parameter sebagai berikut: teknik augmentasi tanpa *zoom range*, ukuran *batch* sebesar 8, *optimizer* Adam dengan nilai *learning rate* 0,001, dan jumlah epoch sebanyak 200. Dengan konfigurasi tersebut, sistem berhasil mencapai akurasi sebesar 100% dan presisi 100%, yang menunjukkan bahwa model mampu melakukan identifikasi secara sempurna pada data uji yang digunakan.

Urutan pengujian dapat dilihat di tabel berikut,

TABEL 2

Urutam Pengujian

No	Tahap Optimasi	Deskripsi
1	Optimasi Augmentasi	Mencari teknik augmentasi terbaik dengan hyperparameter lainnya diatur tanpa <i>zoom range</i> dan dengan <i>zoom range</i> .
2	Optimasi Batch Size	Menggunakan <i>optimizer</i> dan <i>learning rate</i> terbaik untuk menguji batch size yang optimal.
3	Optimasi Optimizer & Learning Rate	Mencari <i>optimizer</i> terbaik dengan parameter umum atau default. Setelah <i>optimizer</i> terbaik ditemukan, <i>optimizer</i> tersebut digunakan untuk mencari <i>learning rate</i> terbaik.
4	Optimasi Epoch	Dengan semua hyperparameter lainnya yang telah dioptimalkan, pengujian terakhir dilakukan untuk menentukan jumlah epoch terbaik.

1. Optimasi Augmentasi

Tahap pertama adalah pengujian augmentasi data yang bertujuan mencari strategi augmentasi terbaik guna meningkatkan kinerja model. Perbandingan dilakukan antara model yang dilatih tanpa *zoom range* dan dengan *zoom range*. Hasil menunjukkan bahwa model tanpa *zoom range* mampu mencapai akurasi pelatihan dan validasi sebesar 100% secara cepat dan stabil. Sementara itu, model dengan *zoom range* juga memperoleh akurasi 100% pada dataset uji, yang mengindikasikan bahwa model memiliki ketangguhan terhadap variasi skala atau jarak citra.

TABEL 3

Tabel Summary Optimasi Augmentasi

Skema	Akurasi	Presisi
Tanpa Zoom Range	100%	100%
Dengan Zoom Range	100%	100%

2. Optimasi Batch Size

Tahap berikutnya adalah pengujian ukuran *batch size* untuk menemukan konfigurasi paling optimal. Tiga ukuran diuji, yaitu 8, 32, dan 64. *Batch size* 8

menunjukkan peningkatan akurasi pelatihan yang cepat dan stabil di angka 100% dengan penurunan *loss* yang tajam pada awal pelatihan. *Batch size* 32 memberikan akurasi yang juga stabil dan mencapai 100% pada dataset uji, meskipun proses konvergensiya sedikit lebih lambat. Sebaliknya, *batch size* 64 cenderung menghasilkan akurasi validasi yang lebih rendah dan kurang stabil, dengan *loss* validasi yang berfluktuasi signifikan, menandakan adanya gradien yang kurang bervariasi.

TABEL 4

Tabel Summary Optimasi *Batch Size*

Skema	Akurasi	Presisi
8	100%	100%
32	100%	100%
64	93.33%	95.00%

3. Optimasi *Optimizer & Learning Rate*

Pengujian selanjutnya berfokus pada pemilihan *optimizer* dan *learning rate* terbaik. Konfigurasi *Adam* dengan *learning rate* 0.001 menghasilkan akurasi pelatihan dan validasi yang sangat tinggi serta stabil, mendekati 100%. *Confusion matrix* menunjukkan kinerja sempurna, di mana semua sampel terklasifikasi dengan benar. Sementara itu, *Adam* dengan *learning rate* 0.0001 memberikan hasil yang tetap impresif, dengan akurasi 96.67% pada data uji, presisi rata-rata 97.50%, *recall* 96.67%, dan F1-score 96.57%. Kurva konvergensi pada konfigurasi ini stabil, menunjukkan bahwa model mampu mempelajari pola secara konsisten.

Tabel 5 Tabel Summary Optimasi *Batch Size*
(E)

Skema	Akurasi	Presisi
Adam & 0.001	100%	100%
Adam & 0.0001	96.67%	97.50%
Skema	Akurasi	Presisi
Nadam & 0.001	96.67%	97.50%
Nadam & 0.0001	93.33%	96.00%
RMSProp & 0.001	100%	100%
RMSProp & 0.001	100%	100%
SGD & 0.001	96.67%	97.50%
SGD & 0.0001	56.67%	61.67%

4. Optimasi *Epoch*

Tahap akhir dilakukan pengujian jumlah *epoch* untuk menentukan lama pelatihan yang paling efektif. Pada *epoch* 100, model mampu mencapai akurasi yang tinggi dan stabil, menunjukkan bahwa model dapat mempelajari pola secara optimal tanpa *overfitting*.

Peningkatan *epoch* menjadi 200 tetap mempertahankan akurasi tinggi, namun terjadi sedikit fluktuasi pada *loss* validasi yang mengindikasikan potensi *overfitting* awal. Sementara itu, pada *epoch* 300, meskipun akurasi pelatihan tetap tinggi, *loss* validasi mengalami kenaikan pada beberapa iterasi akhir, menunjukkan bahwa pelatihan yang terlalu lama dapat mengurangi stabilitas model. Hal ini mengindikasikan bahwa penggunaan *epoch* yang terlalu banyak tidak selalu memberikan

peningkatan kinerja yang signifikan, bahkan berpotensi merugikan jika tidak diimbangi dengan teknik regularisasi.

TABEL 6

Tabel Summary Optimasi *Batch Size*

Skema	Akurasi	Presisi
100	100 %	100 %
200	200 %	200 %
300	300 %	300 %

V. KESIMPULAN

Berdasarkan hasil pengujian, model deteksi sidik bibir berbasis MobileNetV2 terbukti sangat efektif. Konfigurasi terbaik yang ditemukan adalah penggunaan *optimizer* Adam dengan *learning rate* 0.001, yang dilatih selama 100 hingga 200 epoch. Konfigurasi ini, yang didukung oleh augmentasi data, berhasil mencapai akurasi klasifikasi sempurna 100% pada data uji. Keberhasilan ini juga tercermin dari metrik *Precision*, *Recall*, dan F1-Score yang semuanya mencapai 100%, serta visualisasi grafik yang menunjukkan konvergensi yang sangat stabil. Meskipun demikian, pengujian juga menunjukkan bahwa memperpanjang pelatihan hingga 300 epoch dapat menyebabkan sedikit penurunan akurasi, mengindikasikan adanya titik optimal dalam durasi pelatihan.

Hasil akhir dari pengujian menegaskan bahwa sistem yang dikembangkan sangat efektif dalam mengenali pola sidik bibir dan memiliki potensi tinggi untuk diterapkan dalam konteks identifikasi biometrik. Sistem ini mampu mencapai akurasi dan presisi sebesar 100%, yang berarti seluruh data uji berhasil diklasifikasikan dengan benar tanpa adanya kesalahan. Pencapaian ini membuktikan bahwa sidik bibir memiliki potensi signifikan sebagai data biometrik alternatif yang efektif, dan metode CNN berbasis MobileNetV2 yang digunakan terbukti sangat menjanjikan.

Untuk pengembangan sistem di masa mendatang, ada beberapa saran yang dapat dipertimbangkan. Pertama, jumlah dataset perlu ditingkatkan agar sistem mampu beradaptasi dengan variasi individu yang lebih luas. Kedua, penambahan data dari berbagai kondisi pencahayaan, orientasi bibir, atau ekspresi wajah akan membuat sistem lebih tahan terhadap *noise* dan variasi lingkungan. Ketiga, perlu dilakukan pengujian terhadap data baru (data luar) untuk memastikan sistem tidak mengalami *overfitting* dan mampu melakukan generalisasi dengan baik. Terakhir, pengembangan sistem segmentasi otomatis berbasis YOLO atau MediaPipe dapat mempercepat proses identifikasi dan meningkatkan akurasi deteksi area bibir.

REFERENSI

- [1] E. Novita Shandra, B. Darma Setiawan, and Y. Arum Sari, “Klasifikasi Pola Sidik Bibir Untuk Menentukan Jenis Kelamin Manusia Dengan Metode Gray Level Co-Occurrence Matrix Dan Support Vector Machine,” 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [2] W. Farrukh and D. van der Haar, “Lip print-based identification using traditional and deep learning,” *IET Biom*, vol. 12, no. 1, pp. 1–12, Jan. 2023, doi: 10.1049/bme2.12073.

- [3] I. A. Priyambudi, H. Fauzi Tresna, N. Kumalasari, and C. Pratiwi, "IDENTIFIKASI JENIS KELAMIN INDIVIDU MELALUI POLA SIDIK BIBIR BERBASIS SMARTPHONE ANDROID IDENTIFICATION OF GENDER THROUGHT LIPS PRINT PATTERN BASED ON SMATRPHONE ANDROID."
- [4] N. Ghimire *et al.*, "Lip print pattern: an identification tool," *Health Renaissance*, vol. 11, no. 3, pp. 229–233, Jan. 2014, doi: 10.3126/hren.v11i3.9637.
- [5] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," 1987.
- [6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015, [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [7] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognit*, vol. 30, no. 2, pp. 295–309, 1997, doi: 10.1016/S0031-3203(96)00068-4.
- [8] "Adaptive Histogram Equalization and Its Variations," 1987.
- [9] I Made Subrata Sandhiyasa and Devi Valentino Waas, "Real Time Face Recognition for Mobile Application Based on Mobilenetv2," *Jurnal Multidisiplin Madani*, vol. 3, no. 9, pp. 1855–1864, Sep. 2023, doi: 10.55927/mudima.v3i9.5924.
- [10] K. Azmi, S. Defit, and U. Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," vol. 16, no. 1, p. 2023.