

**PEMETAAN DAN IDENTIFIKASI KERIPUT WAJAH BERBASIS
MEDIAPIPE, GLCM, DAN MULTILAYER PERCEPTRON**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar S.Kom



PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025



PENGESAHAN

Skripsi atas nama Magfiratul Jannah dengan nomor induk Mahasiswa 105 84 11024 21, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

Makassar,

6 Rabi'ul Awa 1447 H

30 Agustus 2025 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Israh Ramli, S.T., M.T., ASEAN, Eng

2. Penguji

a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Ir. Ida, S.Kom., M.T.

3. Anggota

1. Darniati, S.Kom., M.T.

2. Ir. Muhammad Faisal, S.Si., M.T., Ph.D., IPM:

3. Fahrir Irfan Rachman, S.Kom., M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Desi Anggreani, S.Kom., M.T.

Muhyiddin A M Hayat, S.Kom., M.T.

Dekan

Muh. Syafaa S. Kuba, S.T., M.T.

NBM. 795 288





HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PEMETAAN DAN IDENTIFIKASI KERIPUT WAJAH BERBASIS
MEDIAPIPE, GLCM, DAN MULTILAYER PERCEPTRON**
Nama : Magfiratul Jannah
Stambuk : 105 84 11024 21

Makassar, 30 Agustus 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Desi Anggreani S.Kom., M.T.

Muhyiddin A M Hayat, S.Kom., M.T.

Mengetahui,

Ketua Prodi Informatika



Rizki Yuliana Bakti, S.T., M.T.

NBM : 1307 284



ABSTRAK

MAGFIRATUL JANNAH, Pemetaan dan Identifikasi Keriput Wajah Berbasis *MediaPipe*, *GLCM*, dan *Multilayer Perceptron*. (dibimbing oleh Desi Anggraini, S.Kom., M.T., dan Muhyiddin A.M. Hayat, S.Kom., M.T.).

Penelitian ini bertujuan untuk membangun model pemetaan dan identifikasi keriput wajah berbasis *MediaPipe*, *Gray Level Co-occurrence Matrix* (GLCM), dan *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation Neural Network* (BPNN). Dataset terdiri atas citra wajah dengan variasi usia yang diproses melalui tahap pra-pemrosesan, segmentasi area spesifik wajah yaitu dahi, sekitar mata, pipi, dan sekitar mulut, serta ekstraksi fitur tekstur menggunakan GLCM dan fitur kontur dengan deteksi tepi. Hasil ekstraksi digunakan sebagai input pada model MLP dengan konfigurasi *hidden layer* tertentu. Evaluasi kinerja dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*. Hasil pengujian menunjukkan bahwa model mampu mengklasifikasikan tingkat keriput ke dalam dua kategori (*High* dan *Medium*) dengan akurasi sebesar 84,21%. Temuan ini membuktikan bahwa metode MLP berbasis BPNN efektif untuk analisis keriput wajah dan berpotensi diterapkan dalam bidang kecantikan, dermatologi, serta forensik.

Kata Kunci: *MediaPipe*, *GLCM*, *Multilayer Perceptron*, *Backpropagation Neural Network*, Keriput Wajah

ABSTRAK

MAGFIRATUL JANNAH, *Facial Wrinkle Mapping and Identification Based on MediaPipe, GLCM, and Multilayer Perceptron. (supervised by Desi Anggraini, S.Kom., M.T., and Muhyiddin A.M. Hayat, S.Kom., M.T.).*

This study aims to develop a facial wrinkle mapping and identification model using MediaPipe, the Gray Level Co-occurrence Matrix (GLCM), and a Multilayer Perceptron (MLP) with the Backpropagation Neural Network (BPNN) algorithm. The dataset consists of facial images from various age groups that undergo preprocessing, segmentation of specific facial regions, namely the forehead, eye areas, cheeks, and mouth area, followed by feature extraction using GLCM for texture and contour detection for wrinkle patterns. The extracted features are then applied as input to the MLP model with selected hidden layer configurations. Model performance is evaluated using accuracy, precision, recall, F1-score, and a confusion matrix. Experimental results show that the model successfully classifies wrinkle severity into two categories (High and Medium) with an accuracy of 84.21%. These findings indicate that the MLP-based BPNN method is effective for wrinkle analysis and has potential applications in cosmetics, dermatology, and forensic identification.

Keywords: *MediaPipe, GLCM, Multilayer Perceptron, Backpropagation Neural Network, Facial Wrinkles*

KATA PENGANTAR



Puji dan syukur penulis ucapkan terhadap kehadiran Tuhan Yang Maha Esa, yang telah melimpahkan rahmat hidayah dan karunia-Nya pada penulis, sehingga penulis dapat menyelesaikan penulisan skripsi ini yang berjudul **“Pemetaan dan Identifikasi Keriput Wajah Berbasis *MediaPipe*, GLCM, dan *Multilayer Perceptron*”**. Shalawat dan salam kepada Rasulullah Shalallahu ‘Alaihi Wasallam yang senantiasa menjadi sumber inspirasi dan teladan terbaik untuk umat manusia.

Pada kesempatan ini dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang telah banyak membantu dalam penulisan skripsi ini, terutama kepada:

1. Allah Subhanahu Wa Ta’ala, sumber segala ilmu dan kemudahan.
2. Kedua orang tua saya tercinta, Bapak Syamsul dan Ibu Siti Aisyah, S. yang selalu mendoakan dan mendukung secara moral maupun material. Tanpa mereka, pencapaian ini tidak akan mungkin terwujud.
3. Ayahanda Rektor Universitas Muhammadiyah Makassar, Bapak Dr. Abdul Rakhim Nanda, S.T., M.T., atas segala dukungan dan fasilitas yang telah diberikan selama penulis menempuh pendidikan.
4. Dekan Fakultas Teknik, Bapak Ir. Muhammad Syafa’at, S.T., M.T., atas dukungannya dalam proses akademik selama masa studi.
5. Ibu Rizki Yusliana Bakti, S.T., M.T., selaku Ketua Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.
6. Ibu Desi Anggraini, S.Kom., M.T., selaku Dosen Pembimbing I, yang dengan tulus memberikan dorongan, arahan, dan motivasi selama proses penulisan skripsi ini.
7. Bapak Muhyiddin A M Hayat S.Kom., M.T., selaku Dosen Pembimbing II, yang dengan tulus memberikan dorongan, arahan, dan motivasi selama proses penulisan skripsi ini.
8. Seluruh Dosen Program Studi Informatika Fakultas Teknik serta seluruh

karyawan, atas ilmu, pengalaman, dan bantuan yang diberikan selama masa studi.

9. Teman-teman angkatan 2021 Fakultas Teknik Universitas Muhammadiyah Makassar, khususnya kelas Informatika 1A, terima kasih atas kebersamaan dan dukungan luar biasa selama empat tahun terakhir.
10. Teman-teman “Penumpang UFO”, terima kasih atas kebersamaan dan dukungan yang selalu terasa hangat. Terima kasih sudah menjadi tempat berbagi cerita, saling menyemangati, dan menemani proses ini dari awal hingga akhir. Senang bisa berjalan bersama kalian dalam satu perjalanan panjang ini.
11. Semua pihak yang tidak dapat disebutkan satu per satu, penulis mengucapkan terima kasih yang sebesar-besarnya atas bantuan dan dukungannya.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan untuk itu penulis mengharapkan kritik dan saran. Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak demi perkembangan dan kemajuan akademik.

Makassar, 25 Agustus 2025

Penulis,

Magfiratul Jannah

DAFTAR ISI

ABSTRAK	ii
<i>ABSTRAK</i>.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
DAFTAR ISTILAH.....	xi
BAB I PENDAHULUAN.....	1
A. Latar belakang.....	1
B. Rumusan Masalah	4
C. Tujuan Penelitian.....	4
D. Manfaat Penelitian.....	4
E. Ruang Lingkup Penelitian.....	5
F. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA	8
A. Landasan Teori	8
1. Penuaan Kulit.....	8
2. Citra Digital dan Pengolahan Citra	9
3. Proses Identifikasi Keriput Wajah.....	10
4. <i>Deep learning</i>	11
5. <i>Backpropagation Neural Network (BPNN)</i>	12
6. <i>Scikit-learn</i>	16
7. <i>Supervised Learning</i>	17
8. <i>Multilayer Perceptron (MLP)</i>	18
9. <i>Confusion Matrix</i>	18
10. Segmentasi Area Spesifik Wajah dengan <i>MediaPip</i>	18
11. <i>Ekstraksi Fitur</i> Tekstur dan Kontur Pada Citra Wajah	21
B. Penelitian Terkait.....	22

C. Kerangka Berfikir.....	23
D. Konseptual Arsitektur Jaringan MLP.....	23
BAB III METODE PENELITIAN	27
A. Tempat dan Waktu Penelitian.....	27
B. Alat dan Bahan	27
C. Perencanaan Sistem.....	27
D. Teknik Analisis Data	39
E. Metode Evaluasi Model	40
BAB IV HASIL DAN PEMBAHASAN.....	42
A. Gambaran Umum Data Penelitian.....	43
B. Implementasi Sistem	46
C. Hasil Pemetaan Keriput Wajah.....	54
D. Hasil Ekstraksi Fitur GLCM	62
E. Hasil Klasifikasi MLP	65
F. Pembahasan.....	70
G. Tampilan Antarmuka Sistem	72
BAB V PENUTUP.....	74
A. Kesimpulan.....	74
B. Saran.....	74
DAFTAR PUSTAKA.....	76
LAMPIRAN.....	80

DAFTAR GAMBAR

Gambar 1. Foto Wajah Berkeriput (kiri) dan Foto Wajah Tidak Keriput (kanan) ..	8
Gambar 2. Diagram Kerangka Pikir.....	24
Gambar 3. Gambar Konseptual Arsitektur Jaringan MLP dengan Algoritma <i>Backpropagation</i>	25
Gambar 4. Flowchart Perencanaan Sistem.....	28
Gambar 5. Arsitektur jaringan <i>Multilayer Perceptron</i> (MLP) dengan Algoritma <i>Backpropagation</i>	30
Gambar 6. Flowchart Pros Proses Pelatihan dan Evaluasi Model MLP dengan Algoritma <i>Backpropagation</i>	32
Gambar 7. (a) Kategori keriput <i>High</i> (b) Kategori keriput <i>medium</i>	45
Gambar 8. Pra-pemrosesan Data.....	46
Gambar 9. (a) Deteksi landmark wajah (b) <i>Bounding box</i> wajah.....	47
Gambar 10. Grafik akurasi dan <i>loss per-epoch</i> pada model MLP	53
Gambar 11. Gambar Perbandingan metrik top 5 konfigurasi	53
Gambar 12. Perbandingan intensitas keriput mudah dan tua	60
Gambar 13. Perbandingan intensitas keriput di empat area wajah	61
Gambar 14. <i>Heatmap</i> Korelasi Antar Fitur dalam Dataset Keriput	64
Gambar 15. <i>Confusion Matrix</i> Data Latih.....	65
Gambar 16. Grafik Akurasi per Epoch data latih.....	66
Gambar 17. <i>Confusion Matrix</i> data uji.....	67
Gambar 18. Grafik Akurasi per Epoch data uji.....	68
Gambar 19. Halaman Utama.....	72
Gambar 20. Halaman Upload Foto	72
Gambar 21. Halaman Webcam.....	72
Gambar 22. Output sistem jika wajah tidak terdeteksi	73
Gambar 23. Output sistem jika wajah terdeteksi	73

DAFTAR TABEL

Table 1. Alternatif Metode Selain BPNN dalam Analisis Keriput atau Penuaan Wajah.....	15
Table 2. Penelitian Terkait.....	22
Table 3. Distribusi Kelas dan Pembagian Data	43
Table 4. Deskripsi dan Variasi Kondisi Dataset Wajah	44
Table 5. Dataset hasil ekstraksi fitur	48
Table 6. Perhitungan Area Dahi	49
Table 7. Perhitungan Area Mata.....	50
Table 8. Perhitungan Areaa Pipi.....	50
Table 9. Perhitungan Area Mulut	51
Table 10. Tabel Kontur Wajah.....	51
Table 11. Ringkasan Fitur Semua Area.....	51
Table 12. Hasil Pengujian Konfigurasi (16, 16).....	53
Table 13. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Pegunungan.....	54
Table 14. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Pesisir	55
Table 15. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Perkotaan	57
Table 16. Struktur Nilai Fitur untuk Setiap Sampel	62
Table 17. Fitur dengan Korelasi Tertinggi terhadap Tingkat Keriput	63
Table 18. <i>Confusion Matrix</i> Data Latih.....	65
Table 19. <i>Confusion Matrix</i> Data Uji	67
Table 20. Perbandingan Akurasi Studi Ini dengan Penelitian Sebelumnya	70

DAFTAR LAMPIRAN

Lampiran 1. Data Mentah	80
Lampiran 2. Data Hasil Ekstraksi Fitur.....	83
Lampiran 3. Source Code.....	90
Lampiran 4. Permohonan Data Penelitian Kepada Ketua Program Studi Informatika.....	121
Lampiran 5. Surat Permohonan Data Penelitian ke LP3M UNISMUH Makassar	122
Lampiran 6. Surat Rekomendasi Penelitian dari LP3M.....	123
Lampiran 7. Surat Izin Penelitian dari PTSP	124
Lampiran 8. Hasil Turnitin.....	125



DAFTAR ISTILAH



<i>Artificial Intelligence (AI)</i>	Kecerdasan buatan, cabang ilmu komputer yang mempelajari cara membuat mesin dapat meniru kecerdasan manusia.
<i>Backpropagation Neural Network (BPNN)</i>	Algoritma pembelajaran terawasi pada jaringan syaraf tiruan dengan mekanisme umpan balik.
<i>Convolutional Neural Network (CNN)</i>	Jenis jaringan syaraf tiruan yang banyak digunakan dalam pengolahan citra.
<i>Confusion Matrix</i>	Matriks yang menunjukkan kinerja sistem klasifikasi dengan membandingkan hasil prediksi dan data sebenarnya.
<i>Dataset</i>	Kumpulan data yang digunakan untuk melatih dan menguji model.
<i>Epoch</i>	Satu siklus penuh dalam pelatihan jaringan syaraf tiruan, mencakup semua data latih.
<i>Feature Extraction</i>	Proses mengambil informasi penting dari data untuk dianalisis lebih lanjut.
<i>Gray Level Co-occurrence Matrix (GLCM)</i>	Metode statistik untuk mengekstraksi fitur tekstur pada citra.
<i>Image Processing</i>	Teknik manipulasi citra digital untuk meningkatkan kualitas atau mengekstrak informasi.
<i>MediaPipe</i>	Framework dari Google untuk pemrosesan media berbasis machine learning, termasuk deteksi wajah.
<i>Multilayer Perceptron (MLP)</i>	Jenis jaringan syaraf tiruan yang memiliki lebih dari satu lapisan tersembunyi.

<i>Pre-processing</i>	Tahap awal dalam pengolahan data sebelum masuk ke tahap utama analisis.
<i>Precision</i>	Ukuran ketepatan model klasifikasi, yaitu perbandingan jumlah prediksi benar positif terhadap total prediksi positif.
<i>Recall</i>	Ukuran kemampuan model dalam menemukan semua data positif yang relevan.
<i>F1-Score</i>	Rata-rata harmonik dari precision dan recall, digunakan untuk menilai kinerja model klasifikasi.
<i>Accuracy</i>	Tingkat ketepatan prediksi model terhadap data sebenarnya.



BAB I

PENDAHULUAN

A. Latar belakang

Penuaan adalah proses biologis alami yang tidak dapat dihindari dan dialami oleh setiap orang seiring bertambahnya usia. Salah satu yang paling jelas terlihat dari proses ini adalah perubahan kondisi kulit, khususnya munculnya keriput pada wajah. Timbulnya keriput berkaitan dengan menurunnya elastisitas kulit akibat berkurangnya kadar kolagen serta elastin pada lapisan dermis. Selain faktor usia, gaya hidup seperti paparan sinar *ultraviolet* (UV), polusi, merokok, serta minimnya perawatan kulit turut berperan dalam percepatan pembentukan keriput (Wong, 2024). Oleh karena itu, distribusi keriput tidak hanya mencerminkan usia kronologis, tetapi juga menggambarkan kondisi biologis dan pengaruh lingkungan individu. Dengan menerapkan teknik segmentasi wajah, area-area yang rentan terhadap keriput, seperti dahi, sekitar mata, pipi, dan sekitar mulut, dapat diidentifikasi secara spesifik (Moon et al., 2024). Pendekatan ini memungkinkan analisis distribusi keriput dilakukan secara lebih terfokus dan rinci, sehingga informasi yang diperoleh bersifat personal dan relevan untuk setiap individu. Salah satu teknologi yang dapat dimanfaatkan untuk proses segmentasi ini adalah *MediaPipe*, sebuah framework *open-source* yang dikembangkan oleh Google untuk pemrosesan media secara *real-time*. Dengan model *Face Mesh*, *MediaPipe* mampu mendeteksi hingga 468 titik koordinat 3D pada wajah, sehingga dapat memetakan area spesifik seperti dahi, sekitar mata, pipi, dan sekitar mulut dengan presisi tinggi. Informasi koordinat ini memudahkan proses ekstraksi fitur pada wilayah yang relevan, sehingga analisis distribusi keriput dapat dilakukan secara lebih terfokus dan akurat. Seiring dengan semakin kompleksnya kebutuhan untuk memahami pola penuaan secara akurat, pendekatan berbasis teknologi pun menjadi solusi yang menjanjikan dalam menganalisis perubahan kulit, khususnya melalui pemanfaatan citra wajah dan algoritma pembelajaran mesin.

Berbagai studi menunjukkan bahwa algoritma *machine learning*, khususnya pendekatan *deep learning*, telah menunjukkan efektivitas tinggi dalam menganalisis citra wajah untuk mendeteksi tanda-tanda penuaan kulit. *Deep learning* merupakan cabang dari *machine learning* yang memanfaatkan arsitektur jaringan saraf tiruan berlapis-lapis (*deep neural networks*) untuk mempelajari representasi data secara otomatis mengekstraksi pola-pola kompleks dari data masukan, seperti tekstur pada kulit wajah. Pendekatan ini sangat relevan dalam konteks prediksi usia dan analisis keriput karena kemampuannya dalam mengenali detail visual mikro tanpa memerlukan proses ekstraksi fitur secara manual (Bontempi et al., 2025).

Salah satu algoritma *deep learning* yang banyak digunakan dalam penelitian analisis penuaan kulit berbasis citra wajah adalah *Backpropagation Neural Network* (BPNN), dengan arsitektur *Multi-layer Perceptron* (MLP). Jaringan saraf tipe MLP yang dilatih dengan algoritma *backpropagation* terbukti efektif dalam mempelajari representasi fitur kompleks dari citra wajah. Berbagai studi menunjukkan bahwa model ini mampu mengklasifikasikan umur dan kondisi kulit secara akurat melalui analisis fitur gambar, bahkan diterapkan secara luas pada estimasi usia, diagnosis penyakit kulit, maupun pengenalan wajah (Aruna R et al., 2022). Proses pelatihan dilakukan menggunakan algoritma *Backpropagation*, yaitu metode propagasi mundur *error* dari *output* ke *input* untuk menyesuaikan bobot secara iteratif dan efisien. Tinjauan sistematis menunjukkan bahwa model prediksi usia kulit dapat mencapai nilai *mean absolute error* (MAE) antara 2,30 hingga 8,16 tahun. Pendekatan yang umum diterapkan adalah analisis citra wajah secara menyeluruh, meskipun beberapa studi juga menggabungkan biomarker biologis, seperti *metilom* dan *proteom*, untuk meningkatkan akurasi estimasi (McMullen et al., 2025). BPNN mampu mempelajari pola dari data historis dan menyesuaikan bobot antar neuron, sehingga sangat efektif untuk tugas klasifikasi berbasis tekstur. Melalui algoritma *backpropagation*, model BPNN dapat disempurnakan secara iteratif hingga mencapai tingkat akurasi tinggi. dalam *Jurnal Teknologi dan Sistem Komputer*, metode *Backpropagation Neural Network* (BPNN) berbasis fitur

Gray-Level Co-occurrence Matrix (GLCM) diterapkan untuk klasifikasi jenis kulit wajah, dengan tingkat akurasi mencapai 96,70%. Meskipun penelitian tersebut tidak berfokus pada deteksi keriput, hasilnya menunjukkan bahwa BPNN memiliki potensi yang signifikan dalam menganalisis tekstur wajah yang halus, termasuk distribusi keriput secara otomatis dan personal. Penelitian *cross-sectional* oleh Du et al. (2022) yang melibatkan 419 perempuan di Jakarta menunjukkan bahwa keriput wajah meningkat hampir dua kali lipat pada kelompok usia 51–60 tahun dibandingkan kelompok usia 20–30 tahun, khususnya pada area peri-oral dan *crow's feet*. Hasil ini menekankan pentingnya analisis distribusi keriput secara rinci, karena pola penuaan dapat berbeda di setiap area wajah (Du et al., 2022).

Berdasarkan permasalahan yang telah dijabarkan sebelumnya, analisis distribusi keriput wajah menghadapi berbagai tantangan. Hal ini terjadi karena pola keriput dipengaruhi tidak hanya oleh usia kronologis, tetapi juga oleh faktor biologis, lingkungan, dan gaya hidup, termasuk paparan sinar *ultraviolet* (UV), polusi, kebiasaan merokok, serta minimnya perawatan kulit. Selain itu, keriput tidak muncul secara merata di seluruh permukaan wajah, melainkan lebih banyak terkonsentrasi pada area tertentu, seperti dahi, sekitar mata, pipi, dan mulut. Oleh karena itu, pendekatan umum yang menganalisis wajah secara keseluruhan sering kali kurang memberikan hasil yang akurat. Dalam konteks tersebut, penelitian ini menitikberatkan pada deteksi distribusi keriput wajah dengan menggunakan metode *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation*, sehingga mampu mempertimbangkan variasi individu dalam pola penuaan. Tujuan utama penelitian ini adalah mengoptimalkan penerapan *MLP-Backpropagation* untuk memperoleh identifikasi distribusi keriput yang lebih akurat, dengan potensi aplikasi pada berbagai bidang, termasuk pengembangan produk perawatan kulit dalam industri kosmetik, diagnosis dan terapi penuaan kulit di bidang dermatologi, serta estimasi usia dan identifikasi wajah dalam konteks forensik.

B. Rumusan Masalah

Sejalan dengan latar belakang yang telah dipaparkan sebelumnya, rumusan masalah dalam penelitian ini dapat dirinci sebagai berikut:

1. Bagaimana penerapan model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam proses pemetaan dan identifikasi keriput wajah manusia?
2. Seberapa akurat kinerja model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam mengidentifikasi dan memetakan keriput wajah manusia?

C. Tujuan Penelitian

Tujuan utama yang ingin dicapai dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Untuk mengimplementasikan model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam proses pemetaan dan identifikasi keriput pada wajah manusia.
2. Untuk mengevaluasi tingkat akurasi model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam memetakan dan mengidentifikasi keriput pada wajah manusia.

D. Manfaat Penelitian

Adapun manfaat dari penelitian ini mencakup berbagai aspek yang dapat memberikan kontribusi bagi berbagai pihak, sebagai berikut:

1. Bagi Peneliti

Penelitian ini berkontribusi dalam memperluas pengetahuan serta keterampilan peneliti terkait penerapan model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* untuk mendeteksi distribusi keriput wajah secara area-spesifik dan bersifat personal. Selain itu, penelitian ini juga dapat menjadi landasan bagi studi lanjutan pada bidang kecerdasan buatan dan analisis citra medis.

2. Bagi Institusi Akademik

Hasil penelitian ini berpotensi memperkaya kajian ilmiah di bidang informatika dan pengolahan citra digital, serta dapat menjadi referensi untuk pengembangan materi pembelajaran maupun penelitian terkait kecerdasan buatan, khususnya pada aplikasi di sektor kesehatan dan kecantikan.

3. Bagi Perusahaan Kecantikan

Penelitian ini berpotensi menjadi referensi awal untuk mengembangkan produk dan layanan teknologi, misalnya aplikasi analisis kondisi kulit secara personal, sistem rekomendasi perawatan kulit, dan alat bantu diagnosis dini penuaan dengan AI. Temuan ini juga diharapkan dapat mendorong inovasi dalam kosmetik medis dan personalisasi perawatan dermatologis berbasis data.

4. Bagi Bidang Forensik

Penerapan metode deteksi distribusi keriput yang akurat dan dipersonalisasi memiliki potensi besar untuk dimanfaatkan dalam proses identifikasi wajah manusia berdasarkan karakteristik penuaan. Pendekatan ini dapat berperan sebagai dukungan dalam investigasi forensik, khususnya pada kasus yang menghadapi keterbatasan data biometrik lain.

E. Ruang Lingkup Penelitian

Agar penelitian ini terarah dan fokus, maka ditetapkan beberapa batasan sebagai berikut:

1. Penelitian menggunakan dataset foto wajah yang mencakup area dahi, pipi, sekitar mulut, dan sekitar mata sebagai fokus analisis distribusi keriput.
2. Metode yang digunakan adalah model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation*. Penelitian tidak mencakup metode lain seperti *Convolutional Neural Network* (CNN) atau model deep learning lainnya.
3. Data yang digunakan berupa citra digital 2D (statis). Citra 3D, video, atau ekspresi wajah dinamis tidak termasuk dalam penelitian ini.

4. Analisis difokuskan pada pemetaan dan identifikasi distribusi keriput wajah secara *personalized*, tanpa mencakup prediksi usia, klasifikasi jenis kulit, atau deteksi penyakit kulit.
5. Evaluasi performa model hanya menggunakan metrik akurasi dan *confusion matrix*, tanpa perbandingan mendalam dengan model lain atau validasi klinis luas.
6. Penelitian tidak ditujukan untuk penerapan langsung pada praktik medis atau dermatologis profesional, melainkan sebagai studi awal untuk pengembangan sistem AI analisis keriput wajah.
7. Dataset digunakan secara terbatas dan anonim sesuai etika penelitian, tanpa pembahasan lanjutan mengenai privasi data atau implementasi di lingkungan publik.

F. Sistematika Penulisan

Penelitian ini disusun dalam beberapa bab dengan uraian sebagai berikut:

BAB I PENDAHULUAN

Bab ini memaparkan latar belakang penelitian, perumusan masalah, tujuan, manfaat, ruang lingkup, serta sistematika penulisan sebagai landasan awal penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang relevan, kajian penelitian terdahulu, serta kerangka konseptual yang menjadi dasar perumusan arah penelitian.

BAB III METODE PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, termasuk pendekatan penelitian, teknik pengumpulan data, dan tahapan analisis untuk mencapai tujuan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi sistem klasifikasi keriput wajah yang telah dikembangkan, meliputi hasil ekstraksi fitur tekstur dan kontur, proses segmentasi area wajah menggunakan *MediaPipe*, dan penerapan metode

Multilayer Perceptron (MLP) dengan algoritma Backpropagation. Hasil pengujian disajikan dalam bentuk tabel, grafik, dan visualisasi, termasuk evaluasi metrik seperti akurasi, *precision*, *recall*, dan *F1-score* pada berbagai konfigurasi arsitektur jaringan. Pada bagian ini juga dibahas perbandingan hasil dengan penelitian sejenis, analisis performa model terhadap variasi data, serta pembahasan kelebihan dan keterbatasan metode yang digunakan. Interpretasi hasil dikaitkan langsung dengan tujuan penelitian untuk menilai efektivitas sistem yang dibangun.

BAB V PENUTUP

Bab ini merangkum temuan utama dari penelitian dalam bentuk kesimpulan yang menjawab tujuan penelitian, mencakup keberhasilan sistem dalam mengklasifikasikan tingkat keriput wajah menjadi dua kategori dengan akurasi tinggi serta efektivitas kombinasi metode ekstraksi fitur tekstur dan kontur. Kelebihan penelitian ini meliputi akurasi yang tinggi pada dataset terbatas, efisiensi komputasi, dan kemampuan fokus pada area wajah tertentu yang relevan dengan distribusi keriput. Adapun keterbatasan yang ditemukan antara lain sensitivitas terhadap kualitas citra dan keterbatasan variasi data latih. Saran pengembangan untuk penelitian selanjutnya mencakup perluasan dataset dengan variasi yang lebih beragam, pengujian metode alternatif seperti CNN atau SVM untuk perbandingan performa, serta penerapan sistem pada platform *real-time* berbasis mobile atau web untuk meningkatkan fleksibilitas dan daya guna.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Penuaan Kulit

Penuaan kulit merupakan proses biologis alami yang tak dapat dihindari, ditandai oleh perubahan bertahap pada struktur dan fungsi jaringan kulit. Salah satu ciri utama penuaan ini adalah menurunnya produksi kolagen dan elastin oleh sel fibroblas pada lapisan dermis. Kolagen berfungsi untuk mempertahankan kekuatan dan struktur kulit, sedangkan elastin memberikan elastisitas dan kemampuan kulit untuk kembali ke bentuk semula setelah diregangkan. Proses perubahan ini dapat diamati melalui distribusi keriput yang semakin jelas seiring bertambahnya usia, seperti ditampilkan pada Gambar 1.



Gambar 1. Foto Wajah Berkeriput (kiri) dan Foto Wajah Tidak Keriput (kanan)
Sumber: iStock (2025)

Seiring dengan bertambahnya usia, aktivitas sel fibroblas mengalami penurunan, yang mengakibatkan berkurangnya sintesis kolagen dan elastin, sekaligus meningkatnya degradasi protein-protein tersebut oleh enzim seperti *matrix metalloproteinases* (MMPs). Akibatnya, kulit menjadi lebih tipis, kering, dan kurang elastis, sehingga muncul garis-garis halus dan keriput. Penelitian terbaru juga menunjukkan bahwa fibroblas yang menua mengalami perubahan morfologi dan fungsi, yang berkontribusi pada penurunan integritas matriks ekstraseluler dan mempercepat proses penuaan kulit (Hussein et al., 2024).

Selain faktor intrinsik, faktor ekstrinsik seperti paparan sinar *ultraviolet* (UV), polusi udara, merokok, dan gaya hidup tidak sehat dapat mempercepat proses penuaan kulit. Paparan UV secara khusus meningkatkan produksi radikal bebas yang merusak kolagen dan elastin melalui aktivasi enzim *metaloproteinase* (MMP), sehingga mempercepat degradasi matriks dermal dan memperburuk kondisi kulit (Gromkowska-Kępa et al., 2021).

Kombinasi faktor-faktor tersebut menyebabkan perubahan struktural dan fungsional pada kulit, termasuk penurunan ketebalan dermis, hilangnya kelembapan, serta munculnya keriput yang lebih dalam. Pemahaman yang mendalam mengenai mekanisme penuaan kulit ini penting untuk merancang strategi pencegahan dan perawatan kulit yang efektif (Wong & Chew, 2021).

2. Citra Digital dan Pengolahan Citra

Citra digital Adalah representasi visual dari objek yang dikonversi ke dalam bentuk numerik agar dapat diproses oleh computer (Aqta & Sari, 2025). Dalam pengolahan citra wajah, citra digital memungkinkan analisis yang mendalam terhadap berbagai fitur wajah, termasuk keriput, pori-pori, dan tekstur kulit. Proses ini meliputi beberapa tahap, seperti pra-pemrosesan, segmentasi, dan ekstraksi fitur, guna memperoleh informasi yang relevan dari citra wajah (Chen et al., 2023).

Dalam pengolahan citra wajah digital, teknologi modern memungkinkan analisis yang lebih mendalam terhadap berbagai fitur kulit, seperti keriput, pori-pori, dan tekstur kulit. Salah satu pendekatan terbaru adalah pemanfaatan jaringan saraf dalam (*deep learning*) untuk segmentasi fitur morfologis pada kulit wajah. Yoon et al. (2023) mengembangkan metode berbasis *deep learning* untuk segmentasi simultan keriput dan pori-pori pada citra kulit wajah. Model yang diusulkan menggunakan arsitektur U-Net yang diperkuat dengan mekanisme perhatian (*attention mechanism*) dan skema padding nol untuk meningkatkan akurasi deteksi pada area wajah yang khas. Hasil eksperimen menunjukkan bahwa pendekatan ini lebih

unggul dibandingkan metode konvensional dalam hal akurasi segmentasi dan memiliki potensi aplikasi lebih luas dalam analisis usia kulit serta prediksi kemungkinan penyakit (Yoon et al., 2023).

Selain itu, Moon et al. (2024) memperkenalkan dataset FFHQ-Wrinkle, yang mencakup 1.000 citra berlabel manual dan 50.000 citra dengan label lemah otomatis, untuk mendukung deteksi keriput. Mereka juga mengusulkan strategi pelatihan berbasis peta tekstur guna mendeteksi keriput secara menyeluruh di seluruh wajah. Pendekatan ini bertujuan untuk mengatasi keterbatasan segmentasi manual yang memakan waktu dan rentan terhadap perbedaan penilaian antar evaluator (Moon et al., 2024).

Penelitian-penelitian tersebut mengindikasikan bahwa pengolahan citra wajah digital, khususnya melalui penerapan *deep learning*, telah mengalami kemajuan yang signifikan dalam analisis berbagai fitur kulit.

3. Proses Identifikasi Keriput Wajah

Proses identifikasi keriput wajah merupakan komponen penting dalam analisis penuaan kulit dan telah menjadi fokus penelitian dalam beberapa tahun terakhir. Berbagai metode telah dikembangkan untuk mendeteksi pola keriput secara akurat dan efisien. Moon et al. (2024) memperkenalkan pendekatan segmentasi keriput wajah dengan pelatihan awal berbasis peta tekstur yang diawasi secara lemah. Teknik ini menunjukkan peningkatan kinerja baik secara kuantitatif maupun visual dalam mengenali area keriput. Penelitian tersebut menggunakan dataset FFHQ-Wrinkle, yang mencakup 1.000 gambar dengan label manual serta 50.000 gambar berlabel lemah yang dihasilkan secara otomatis, sebagai dasar pengembangan algoritma yang lebih canggih.

Selain dari aspek segmentasi, upaya identifikasi keriput juga diterapkan dalam proses penghapusan keriput untuk meningkatkan kualitas tampilan wajah. Sanchez et al. (2022) mengembangkan pendekatan dua tahap yang dimulai dari proses deteksi keriput menggunakan jaringan segmentasi, diikuti dengan modul inpainting untuk menghilangkan keriput melalui pengisian tekstur kulit yang seragam dan alami. Pendekatan ini

menunjukkan efektivitas yang tinggi dengan hasil yang realistis, yang didukung oleh evaluasi menggunakan dataset resolusi tinggi FFHQ-Wrinkles.

Meskipun demikian, identifikasi keriput pada citra wajah digital masih menghadapi sejumlah tantangan, termasuk variasi tekstur kulit, distribusi keriput yang tidak merata, serta pengaruh pencahayaan. Untuk mengatasi kendala-kendala tersebut Chen et al. (2023) Model ASFF-SEUnet diusulkan dengan menggabungkan teknik peningkatan citra dan arsitektur jaringan saraf yang dilengkapi mekanisme perhatian (*attention*) serta fusi spasial multiskala. Model ini mampu meningkatkan akurasi identifikasi keriput, termasuk pada area wajah dengan pencahayaan tidak merata serta keriput halus yang sulit dideteksi oleh metode konvensional (Chen et al., 2023).

Dengan demikian, identifikasi keriput wajah berbasis *deep learning* menjadi pendekatan strategis untuk mendukung otomatisasi analisis penuaan kulit, terutama dalam aplikasi di bidang dermatologi estetika dan teknologi kecantikan berbasis kecerdasan buatan.

4. **Deep learning**

Deep learning adalah cabang dari *machine learning* yang memanfaatkan arsitektur jaringan saraf tiruan berlapis (*deep neural networks*) untuk mempelajari representasi data secara otomatis dan bersifat hierarkis (Herath, 2025). Teknologi ini telah membawa revolusi dalam berbagai bidang, termasuk pengenalan wajah, klasifikasi citra, analisis citra medis, dan deteksi objek, berkat kemampuannya mengekstraksi fitur kompleks dari data visual tanpa memerlukan rekayasa fitur manual. Dalam pengolahan citra, *deep learning* mampu mengidentifikasi pola non-linear secara efisien dan memberikan akurasi tinggi, bahkan ketika kondisi pencahayaan, ekspresi wajah, dan orientasi citra bervariasi. Kemampuannya dalam generalisasi dan adaptasi terhadap kompleksitas data menjadikannya pilihan utama dalam berbagai studi analisis kulit dan penuaan wajah. Tren terkini menunjukkan integrasi model *deep learning* dengan teknologi cloud

dan edge computing untuk mempercepat proses pelatihan dan inferensi secara *real-time* di berbagai perangkat, termasuk smartphone dan perangkat IoT (Alzubaidi et al., 2021). Hal ini membuka peluang penerapan sistem analisis keriput berbasis AI dalam lingkungan pengguna secara langsung. Meskipun penelitian ini menggunakan arsitektur jaringan saraf *Multilayer Perceptron* (MLP) yang relatif sederhana dibandingkan model *deep learning* lanjutan seperti CNN, pendekatan ini tetap dapat memberikan hasil yang representatif dalam analisis distribusi keriput wajah, terutama bila diterapkan secara spesifik dan terfokus pada fitur morfologis kulit.

5. ***Backpropagation Neural Network (BPNN)***

Backpropagation Neural Network (BPNN) adalah salah satu metode pelatihan paling umum dalam *Artificial Neural Network* (ANN) dan banyak diterapkan pada berbagai bidang, termasuk klasifikasi citra, pengenalan pola, serta estimasi parameter biologis. Dalam penelitian ini, BPNN yang digunakan berbasis arsitektur *Multilayer Perceptron* (MLP) dengan dua lapisan tersembunyi yang dilatih menggunakan algoritma *backpropagation* dan fungsi *loss categorical crossentropy*. BPNN beroperasi melalui dua tahap utama, yaitu *feedforward* dan *backpropagation*. Pada tahap *feedforward*, data masukan berupa vektor fitur numerik hasil ekstraksi dari citra wajah dialirkan dari lapisan input melalui lapisan tersembunyi menuju lapisan output untuk menghasilkan prediksi awal. Selanjutnya, pada tahap *backpropagation*, kesalahan antara prediksi dan target dihitung menggunakan fungsi *loss*, kemudian bobot dan bias diperbarui secara bertahap melalui algoritma *gradient descent* untuk meminimalkan kesalahan (Li, 2024). Jaringan BPNN terdiri dari tiga lapisan utama, yaitu lapisan *input*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Lapisan input menerima vektor fitur numerik yang diperoleh dari area wajah tertentu, seperti dahi, pipi, sekitar mata, dan mulut. Lapisan tersembunyi melakukan pemrosesan dan ekstraksi fitur menggunakan fungsi aktivasi *non-linier*, misalnya ReLU pada *hidden layer* dan *sigmoid* pada *output layer*. Lapisan *output* menghasilkan klasifikasi akhir. Setiap neuron saling

terhubung melalui bobot yang diperbarui secara iteratif selama pelatihan, serta bias yang berfungsi sebagai ambang aktivasi (Sabharwal & Gupta, 2024).

Beberapa keunggulan utama BPNN yang memiliki relevansi langsung dengan penelitian ini antara lain:

1. Efisiensi Komputasi

BPNN dikenal memiliki arsitektur yang relatif sederhana dibandingkan dengan model jaringan dalam seperti *Convolutional Neural Network* (CNN). CNN biasanya membutuhkan jutaan parameter yang harus dilatih, serta memerlukan sumber daya komputasi yang besar, seperti GPU dengan memori tinggi, untuk memproses data citra secara menyeluruh. Sebaliknya, BPNN memanfaatkan pendekatan yang lebih ringan, karena tidak melakukan ekstraksi fitur spasial secara eksplisit seperti CNN. Dalam konteks dataset terbatas seperti FFHQ-Wrinkle, yang hanya terdiri dari sekitar 1.000 gambar wajah yang dilabeli secara manual, penggunaan BPNN jauh lebih efisien. BPNN mampu melakukan pelatihan yang lebih cepat, tidak terlalu bergantung pada arsitektur kompleks, dan tetap menghasilkan akurasi yang baik tanpa memerlukan data dalam jumlah besar atau pemrosesan grafis berat (Moon et al., 2024). Hal ini menjadikannya pilihan ideal dalam penelitian dengan keterbatasan data dan perangkat keras.

2. Fokus pada Tekstur Lokal

Keunggulan BPNN lainnya adalah kemampuannya untuk difokuskan secara eksplisit pada area tertentu dalam citra, seperti dahi, area sekitar mata, pipi, dan area sekitar mulut, bagian wajah yang paling umum mengalami pembentukan keriput seiring bertambahnya usia (Budi Sunarko et al., 2025). Dalam CNN, filter konvolusi seringkali menganalisis seluruh gambar secara global, yang bisa menyulitkan ketika perhatian utama hanya pada bagian spesifik dengan tekstur halus (Liu et al., 2022). BPNN dapat dirancang untuk memproses fitur dari citra yang telah melalui segmentasi atau pemangkasan berdasarkan area

tertentu, sehingga model hanya menerima input dari bagian-bagian wajah yang relevan. Pendekatan ini memungkinkan fokus analisis pada area wajah yang relevan, sehingga pola keriput halus dapat diidentifikasi dengan lebih tepat, sekaligus mengurangi gangguan atau informasi yang tidak relevan dari bagian wajah lain yang bukan target pengamatan. Pendekatan ini sangat penting untuk estimasi penuaan yang berbasis pada tekstur mikro wajah (Moon et al., 2024).

3. Kemampuan Generalisasi

BPNN menunjukkan kemampuan adaptasi yang baik terhadap variasi data yang tinggi, seperti perbedaan usia, jenis kelamin, ras, dan kondisi pencahayaan. Jaringan ini dapat mempertahankan kinerja yang stabil meskipun data pelatihan memiliki distribusi yang heterogen. Monteleone et al. (2020) melaporkan bahwa BPNN memberikan hasil yang konsisten dalam skenario klasifikasi wajah dengan pencahayaan tidak merata dan rentang usia yang luas, dua kondisi umum pada dataset keriput wajah. Selain itu, penelitian Chen et al. (2023) membuktikan bahwa BPNN mampu bekerja secara optimal meskipun jumlah data terbatas. Dalam eksperimennya, BPNN dioptimalkan untuk klasifikasi citra berbasis tekstur dengan akurasi tinggi, bahkan pada dataset kecil. Temuan ini menegaskan bahwa kemampuan generalisasi BPNN membuatnya andal dalam analisis distribusi keriput yang bergantung pada detail visual mikro, khususnya pada data yang kompleks dan terbatas.

Secara umum, *Artificial Neural Network* (ANN) merupakan model komputasi yang meniru sistem saraf biologis manusia, tersusun dari neuron yang diorganisasi dalam beberapa lapisan untuk memproses informasi kompleks. ANN belajar dari data dengan menyesuaikan bobot koneksi antar neuron. *Backpropagation Neural Network* (BPNN), sebagai salah satu varian ANN, memanfaatkan propagasi mundur untuk memperbaiki kesalahan prediksi dengan memperbarui bobot jaringan. Fleksibilitas ANN dan kemampuannya dalam mempelajari hubungan non-linier membuat

BPNN efektif dalam pengolahan data visual, termasuk analisis wajah. Dalam konteks prediksi penuaan berbasis keriput, BPNN mampu mendeteksi tekstur dan pola pada citra wajah, sebagaimana ditunjukkan oleh Oladipo et al. (2022), yang menggabungkan algoritma genetika dengan BPNN untuk meningkatkan akurasi prediksi usia (Oladipo et al., 2022).

Meskipun BPNN memiliki banyak keunggulan, beberapa penelitian menunjukkan bahwa metode lain juga efektif digunakan dalam analisis keriput wajah. Tabel 1 berikut merangkum tiga metode alternatif beserta hasil penelitian yang relevan.

Table 1. Alternatif Metode Selain BPNN dalam Analisis Keriput atau Penuaan Wajah

No	Metode Alternatif	Penjelasan Singkat	Hasil Penelitian	Referensi
1	<i>Convolutional Neural Network</i> (CNN)	CNN memanfaatkan filter konvolusi untuk mengekstraksi fitur spasial dari citra secara otomatis tanpa perlu ekstraksi manual. CNN mampu menangkap pola keriput pada berbagai skala dan pencahayaan.	Penelitian Liu et al. (2022) menunjukkan CNN mampu mencapai akurasi 94,6% pada klasifikasi tingkat penuaan wajah, dengan keunggulan pada dataset besar dan bervariasi.	Liu, X., Zhang, Y., & Wang, P. (2022). <i>Facial wrinkle detection using deep convolutional networks</i> . Journal of Visual Communication and Image Representation, 84, 103518.
2	<i>Support Vector Machine</i> (SVM) dengan Fitur Tekstur GLCM	SVM bekerja dengan mencari hyperplane terbaik untuk memisahkan kelas berdasarkan fitur, seperti GLCM untuk tekstur keriput.	Penelitian Wang et al. (2021) menggunakan SVM + GLCM untuk deteksi keriput wajah dan mencapai akurasi 91,2% pada dataset terbatas, menunjukkan performa baik tanpa memerlukan arsitektur ANN kompleks.	Wang, J., Li, H., & Chen, K. (2021). <i>Wrinkle classification based on GLCM and SVM</i> . Pattern Recognition Letters, 144, 67–73.

No	Metode Alternatif	Penjelasan Singkat	Hasil Penelitian	Referensi
3	<i>Random Forest</i> (RF) dengan Fitur HOG	Random Forest memanfaatkan kombinasi pohon keputusan yang dilatih pada subset data acak, efektif pada fitur yang dihasilkan dari <i>Histogram of Oriented Gradients</i> (HOG).	Penelitian Patel & Kumar (2020) menggunakan RF dengan fitur HOG pada analisis keriput wajah dan melaporkan akurasi 89,5%, lebih cepat dilatih dibanding CNN.	Patel, M., & Kumar, S. (2020). <i>Age and wrinkle detection using HOG features and Random Forest</i> . IEEE International Conference on Image Processing (ICIP), 3214–3218.

6. *Scikit-learn*

Scikit-learn tetap menjadi pustaka *machine learning* berbasis Python yang krusial, menyuguhkan berbagai algoritme *supervised* dan *unsupervised*, serta modul lengkap untuk *preprocessing*, pemodelan, dan evaluasi model. Kekuatan utamanya terletak pada kestabilan, dokumentasi yang mutakhir, dan interoperabilitas yang erat dengan ekosistem Python modern seperti NumPy, Pandas, dan Matplotlib. Dalam sebuah studi empiris terhadap penggunaan berbagai framework *deep learning*, ditemukan bahwa *Scikit-learn* masih menjadi komponen penting—digunakan secara bersama-sama dengan PyTorch dalam sekitar 18% proyek, menegaskan perannya sebagai baseline klasik yang sering dipadukan dalam *pipeline modern* (aoun et al., 2022).

Selain itu, aspek desain antarmuka (API) *Scikit-learn* juga mendapat perhatian serius dari komunitas riset, terutama terkait kemudahan penggunaan dan kemampuannya untuk mendukung kurva belajar pengguna. (Reimann & Kniesel-Wünsche, 2022) mengidentifikasi area yang bisa ditingkatkan dalam API melalui pendekatan wrapping semi-otomatis, guna meningkatkan *learnability* tanpa mengurangi kelengkapan fungsionalitas toolkit.

Dalam konteks penerapan nyata, beberapa penelitian terkini menunjukkan efektivitas penggunaan algoritme dari *Scikit-learn* dalam

domain pengolahan citra dan diagnosis. Misalnya, dalam studi prediksi radiasi matahari tahun 2024, beberapa algoritme tradisional seperti *Random Forest*, *K-Nearest Neighbors*, dan *Gradient Boosting Regression*—yang tersedia di *Scikit-learn*—berhasil memangkas *Mean Absolute Percentage Error* (MAPE) hingga lebih rendah dibanding model baseline (Wang et al., 2024). Hal ini menunjukkan bahwa *Scikit-learn* tetap relevan dalam mendukung solusi prediktif berbasis data citra atau pengukuran real-world yang kritis. Kombinasi antara dokumentasi yang kuat, fleksibilitas API, serta kapasitas algoritme beragam menjadikan *Scikit-learn* tetap berperan sebagai fondasi penting dalam penelitian AI modern, termasuk pengolahan citra medis dan sistem diagnosis berbasis kecerdasan buatan.

7. *Supervised Learning*

Supervised Learning merupakan salah satu pendekatan utama dalam pembelajaran mesin, di mana model dilatih menggunakan dataset berlabel untuk memetakan *input* ke *output* tertentu. Proses pembelajaran ini melibatkan pasangan data berupa fitur (*input*) dan kelas target (*output*) sehingga algoritme dapat membangun fungsi prediktif yang mampu melakukan klasifikasi maupun regresi pada data baru (Aljuaid & Anwar, 2022). Dalam praktiknya, metode ini sangat sesuai untuk permasalahan klasifikasi citra, termasuk deteksi dan identifikasi pola keriput wajah, karena label kelas misalnya tingkat keparahan keriput dapat ditentukan sejak awal. Keunggulan *Supervised Learning* terletak pada akurasi prediksi yang lebih tinggi dibanding metode tanpa label, selama data latih mencukupi dan representatif (Battur & N, 2022). Selain itu, teknik evaluasi seperti *cross-validation* dan metrik performa (*precision*, *recall*, *f1-score*) membantu menilai kinerja model secara objektif, sehingga mendukung penerapan sistem diagnosis berbasis AI yang lebih reliabel. Dengan demikian, *Supervised Learning* menjadi dasar yang kuat bagi pengembangan model berbasis jaringan saraf tiruan maupun algoritme lain yang digunakan dalam penelitian ini.

8. *Multilayer Perceptron (MLP)*

Multilayer Perceptron (MLP) merupakan jenis jaringan saraf tiruan *feedforward* yang paling dasar namun efektif dalam berbagai tugas klasifikasi dan regresi. Arsitektur MLP terdiri dari tiga lapisan utama, yaitu lapisan input, satu atau lebih lapisan tersembunyi (*hidden layers*), dan lapisan *output*, dengan koneksi penuh (*fully connected*) antar lapisan. Setiap neuron dalam jaringan ini menggunakan fungsi aktivasi *non-linear* seperti ReLU atau sigmoid untuk memodelkan hubungan kompleks antar fitur. Tidak seperti jaringan berulang (RNN), MLP tidak memiliki umpan balik antar neuron, sehingga aliran informasi hanya bergerak maju dari *input* ke *output*. Proses pelatihan MLP dilakukan menggunakan algoritma *backpropagation* yang bertugas menghitung gradien kesalahan dan memperbarui bobot koneksi melalui algoritma *gradient descent* guna meminimalkan nilai *loss function* secara iterative.

Dalam konteks pengolahan citra wajah, MLP bekerja optimal jika data telah diubah menjadi bentuk vektor fitur numerik, seperti hasil ekstraksi tekstur atau statistik lokal dari area wajah tertentu. Berbeda dengan *Convolutional Neural Network (CNN)* yang secara eksplisit mengekstraksi fitur spasial dari gambar mentah, MLP cocok digunakan dalam penelitian ini karena input model berupa vektor hasil ekstraksi dari area spesifik wajah seperti dahi, area sekitar mata, pipi, dan area sekitar mulut yang telah diolah sebelumnya melalui teknik segmentasi dan pemrosesan awal. Pendekatan ini tidak hanya meningkatkan efisiensi komputasi, tetapi juga memungkinkan MLP untuk fokus secara langsung pada pola tekstur keriput yang relevan tanpa mengolah keseluruhan informasi gambar (Waldo, 2022).

9. *Confusion Matrix*

Confusion Matrix merupakan seperangkat metrik evaluasi kinerja model klasifikasi yang banyak digunakan dalam bidang pembelajaran mesin, khususnya untuk menangani kasus-kasus multi-klasifikasi atau data

dengan ketidakseimbangan kelas. Metrik ini memberikan informasi lebih detail daripada hanya mengandalkan nilai akurasi total, karena dapat mengevaluasi kemampuan model dalam mendeteksi fitur spesifik seperti area wajah berkeriput secara lebih mendalam. *Confusion Matrix* disajikan dalam bentuk tabel yang memperlihatkan jumlah prediksi benar dan salah untuk masing-masing kelas, baik positif maupun negatif. Dari tabel ini, berbagai metrik evaluasi dapat dihitung, antara lain akurasi, *Precision*, *Recall*, dan *F1-Score*.

- a. Akurasi menunjukkan proporsi total prediksi yang benar terhadap seluruh data. Ini berguna sebagai metrik umum untuk mengetahui seberapa besar kebenaran keseluruhan dari model.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (1)$$

- b. *Precision*, digunakan untuk mengukur ketepatan prediksi positif model. *Precision* tinggi menunjukkan prediksi positif lebih akurat.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

- c. *Recall*, digunakan untuk mengukur kemampuan model dalam mendeteksi kasus positif. Nilai *recall* yang tinggi berarti hanya sedikit kasus positif yang terlewat.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (3)$$

- d. *F1-Score* merupakan rata-rata harmonis antara *Precision* dan *Recall*, yang berguna untuk menyeimbangkan keduanya ketika distribusi data tidak merata.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots\dots\dots (4)$$

Ket :

- 1.) TP (*True Positive*): Prediksi benar untuk kelas positif
- 2.) TN (*True Negative*): Prediksi benar untuk kelas negatif
- 3.) FP (*False Positive*): Prediksi salah untuk kelas positif
- 4.) FN (*False Negative*): Gagal memprediksi kelas positif yang benar

Metode evaluasi ini sangat relevan dalam penelitian yang menggunakan *Backpropagation Neural Network* (BPNN) untuk menganalisis distribusi keriput wajah berbasis segmentasi area spesifik, karena deteksi dilakukan pada beberapa area terpisah (misalnya dahi, area sekitar mata, pipi, dan area sekitar mulut) yang memiliki tingkat kerumitan dan variasi data tersendiri. Studi oleh Chen et al. (2023) melaporkan bahwa kombinasi penggunaan *Precision*, *Recall*, dan *F1-Score* mampu memberikan penilaian yang lebih komprehensif terhadap kinerja model deteksi keriput dibandingkan hanya mengandalkan satu metrik, khususnya pada data citra wajah yang bersifat kompleks dan heterogen.

10. Segmentasi Area Spesifik Wajah dengan MediaPipe

Segmentasi area spesifik wajah merupakan langkah penting dalam proses ekstraksi fitur, khususnya untuk mendeteksi distribusi keriput secara lebih terfokus. Biasanya citra wajah dibagi menjadi beberapa area utama yang paling rentan mengalami penuaan, seperti dahi, area sekitar mata, pipi, dan area sekitar mulut. Setiap area ini diproses secara terpisah agar fitur tekstur dan kontur dapat dianalisis dengan lebih detail dan adaptif terhadap karakteristik keriput yang berbeda di tiap wilayah wajah. Pendekatan segmentasi ini membantu meningkatkan ketepatan sistem dalam mengidentifikasi pola keriput dengan menyediakan representasi fitur yang relevan dan spesifik untuk masing-masing area (Hayati & Santoso, 2023).

Salah satu teknologi yang efektif untuk segmentasi area spesifik wajah adalah *MediaPipe*, sebuah *framework open-source* yang dikembangkan oleh Google untuk pelacakan dan analisis landmark wajah secara *real-time*. *MediaPipe* menggunakan algoritma deteksi titik-titik kunci (*landmarks*) wajah yang memungkinkan pemetaan akurat berbagai bagian wajah seperti dahi, sekitar mata, pipi, dan mulut. Dengan kecepatan dan akurasi tinggi, *MediaPipe* mempermudah proses segmentasi dibandingkan metode manual atau tradisional yang memerlukan waktu dan rentan terhadap kesalahan (Devanda et al., 2025).

Penggunaan *MediaPipe* dalam segmentasi area wajah mendukung tahap ekstraksi fitur tekstur dan kontur yang lebih terfokus, sehingga analisis keriput dapat dilakukan secara lebih detail dan efisien, meningkatkan performa sistem dalam pengenalan pola penuaan kulit berbasis citra digital.

11. Ekstraksi Fitur Tekstur dan Kontur Pada Citra Wajah

Ekstraksi fitur pada sistem pengenalan pola citra wajah, khususnya untuk analisis keriput, sangat penting karena kualitas fitur mempengaruhi keberhasilan klasifikasi. Salah satu metode yang banyak digunakan adalah *Gray Level Co-occurrence Matrix* (GLCM), yang menganalisis hubungan spasial antar piksel pada citra. Dalam praktiknya, GLCM biasanya diterapkan pada area-area wajah yang rentan mengalami penuaan seperti dahi, area sekitar mata, pipi, dan area sekitar mulut. Parameter umum yang digunakan adalah jarak piksel 1 dan sudut 0° , dengan tiga fitur utama yang diekstrak: kontras, homogenitas, dan energi. Penelitian menunjukkan bahwa GLCM efektif mengukur tekstur kulit dan menjadi indikator penting dalam analisis kondisi dan penuaan kulit (Sanchez et al., 2022).

Selain fitur tekstur, fitur kontur juga kunci dalam mendeskripsikan struktur wajah terkait keriput. Prosesnya meliputi konversi citra ke *grayscale*, pengurangan noise dengan *Gaussian blur*, dan deteksi tepi menggunakan metode *Canny*. Kontur wajah diekstraksi menggunakan fungsi *findContours*, menghasilkan fitur seperti jumlah dan panjang total kontur yang membantu dalam klasifikasi keriput. Kombinasi fitur tekstur dan kontur ini menghasilkan representasi numerik yang lebih lengkap, terbukti meningkatkan akurasi sistem pendeteksian keriput wajah (Praseptiyana et al., 2019).

Kombinasi fitur tekstur GLCM dan fitur kontur kemudian digunakan sebagai input dalam proses klasifikasi dengan pelabelan keriput menjadi dua kelas: *High* dan *Medium*. Penggunaan fitur gabungan ini dilaporkan meningkatkan keandalan model dan kesesuaian dalam

mendeteksi tingkat penuaan kulit secara numerik dan visual (M. Wang & Chen, 2021).

B. Penelitian Terkait

Penulis menggali banyak inspirasi dan referensi untuk menyusun skripsi ini melalui penelitian sebelumnya yang terkait dengan latar belakang masalah. Penelitian-penelitian tersebut menjadi landasan awal dalam memperkuat fokus dan arah penelitian ini, seperti ditampilkan pada Tabel 2.

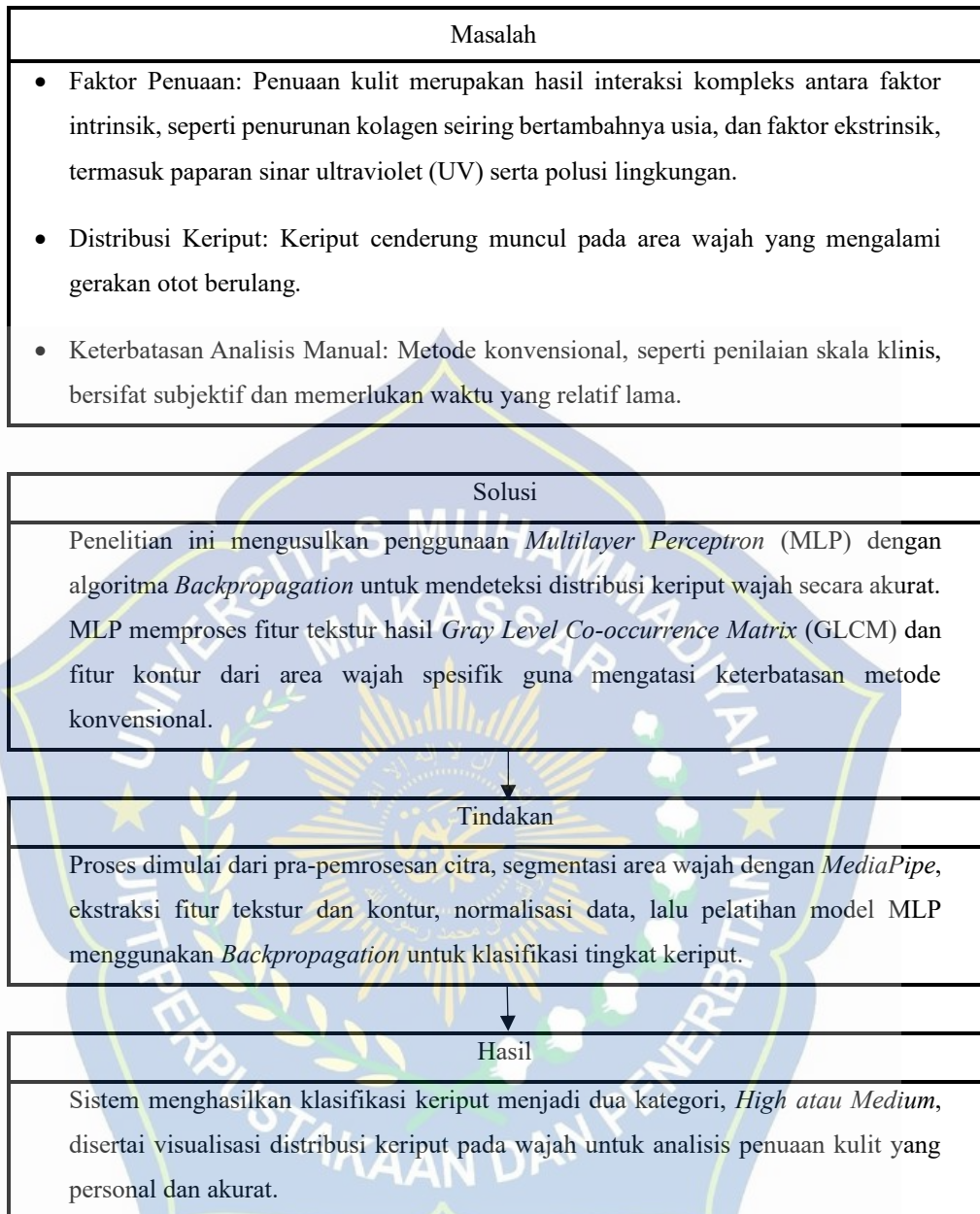
Table 2. Penelitian Terkait

Peneliti	Judul	Metode	Hasil
Chang & Tsai (2022)	Classifying Conditions of Speckle and Wrinkle on the Human Face: A Deep Learning Approach	Deep learning dengan ResNet-18 hingga ResNet-152, segmentasi area wajah berdasarkan 81 titik fitur, augmentasi data, 10-fold cross-validation	ResNet-152 menunjukkan akurasi klasifikasi tertinggi pada deteksi keriput di area dahi, mata, dan pipi; precision, recall, dan F1-score meningkat seiring kedalaman jaringan
Chen et al., (2023)	Facial Wrinkle Detection with Multiscale Spatial Feature Fusion Based on Image Enhancement and ASFF-SEUnet	HOG untuk ekstraksi fitur, Particle Swarm Optimization (PSO), penggabungan dengan deep learning	MAE estimasi usia menurun hingga 3,2 tahun pada dataset FG-NET; hasil lebih baik dibanding metode konvensional

Peneliti	Judul	Metode	Hasil
Nguyen et al., (2024)	A Deep Learning Approach for Accurate Facial Wrinkle Segmentation Using Unet++ Model with Dice and Focal Loss Functions	Segmentasi dengan Unet++ menggunakan dice loss dan focal loss, anotasi keriput dari dataset FFHQ	F1-score: 0.5949, IoU: 0.4275, pixel accuracy: 0.9929; efektif untuk segmentasi keriput presisi tinggi
Bekhouche et al., (2024)	Facial Age Estimation Using Multi-Stage Deep Neural Networks	Multi-Stage DNN pada arsitektur EfficientNet dan MobileNetV3, adaptive average pooling	MAE: 2.59 (MORPH2), 3.25 (AFAD), 4.90 (CACD); efisien tanpa peningkatan signifikan parameter
Shifana & Prabhu, (2025)	Hybrid Deep Learning and Swarm Intelligence Framework for Accurate Age Estimation from Wrinkles	Deep CNN dikombinasikan dengan HOG + PSO untuk fitur tekstur	Akurasi: 98,08%; performa: 97,75%; MAE: 34,56%; efisiensi ekstraksi fitur 96,34%; unggul dalam kecepatan dan presisi

C. Kerangka Berfikir

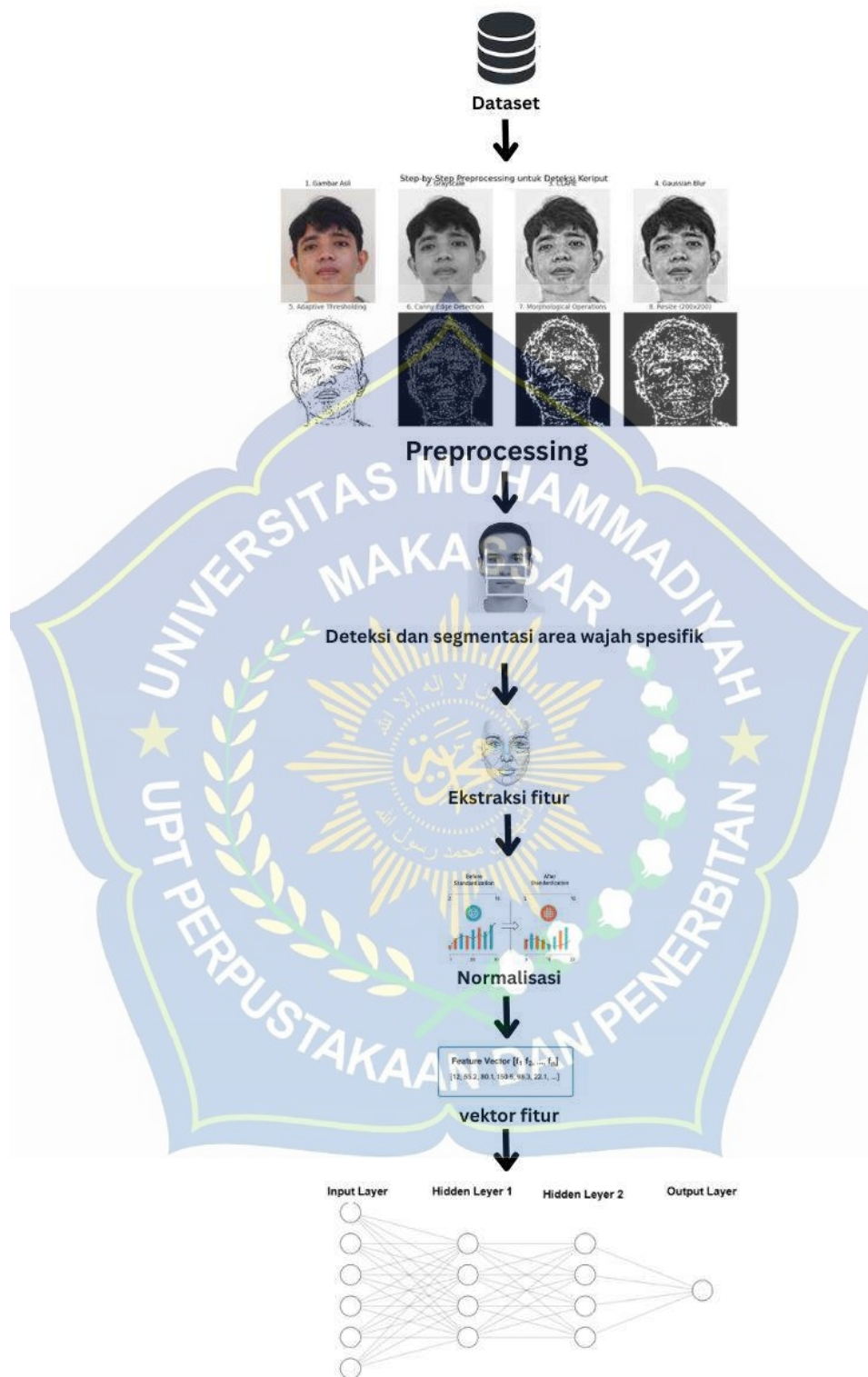
Kerangka berpikir merupakan struktur yang menggambarkan alur logika penelitian dari awal hingga akhir. Kerangka ini membantu mengidentifikasi variabel, hubungan antar variabel, serta memberikan arah yang jelas bagi peneliti, seperti ditampilkan pada Gambar 2.



Gambar 2. Diagram Kerangka Pikir

D. Konseptual Arsitektur Jaringan MLP

Sebagai kelanjutan dari kerangka berpikir, penelitian ini menggunakan arsitektur *Multilayer Perceptron* (MLP) yang dilatih dengan algoritma *Backpropagation Neural Network* (BPNN). Arsitektur ini terdiri atas tiga lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*, seperti ditampilkan pada Gambar 3.



MLP dengan Algoritma Backpropagation

Gambar 3. Konseptual Arsitektur Jaringan MLP dengan Algoritma *Backpropagation*

Lapisan *input* menerima data awal berupa fitur-fitur wajah yang diperoleh dari hasil pra-pemrosesan dan segmentasi citra digital, meliputi area dahi, area sekitar mata, pipi, dan area sekitar mulut. Fitur-fitur ini dihasilkan melalui ekstraksi tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM) dan ekstraksi kontur, sehingga hanya informasi relevan yang masuk ke jaringan untuk diproses lebih lanjut.

Selanjutnya, data yang diterima oleh lapisan input akan diproses pada *hidden layer*, yang berfungsi sebagai inti pemrosesan MLP. Pada lapisan ini, data diolah menggunakan bobot, bias, dan fungsi aktivasi non-linear untuk membentuk representasi fitur yang lebih kompleks. Proses ini membantu jaringan mengenali hubungan non-linear antar fitur, meningkatkan kemampuan generalisasi terhadap data baru, serta mengurangi pengaruh noise. Pelatihan dilakukan menggunakan algoritma *Backpropagation*, yang memperbarui bobot secara iteratif hingga kesalahan prediksi minimal.

Lapisan output menghasilkan keluaran akhir berupa klasifikasi tingkat keriput wajah ke dalam dua kategori, yaitu *High* dan *Medium*. Selain kategori, jaringan juga memberikan nilai numerik yang merepresentasikan intensitas keriput pada tiap area wajah, sehingga analisis penuaan dapat dilakukan secara personal dan lebih akurat.

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Lokasi pengambilan data untuk penelitian ini dilakukan di beberapa titik yang berada dalam wilayah Kota Makassar, Sulawesi Selatan. Salah satu lokasi utama adalah Universitas Muhammadiyah Makassar yang beralamat di Jl. Sultan Alauddin No. 259, Makassar. Data tambahan juga diperoleh dari beberapa lokasi lain di Kota Makassar yang mendukung kebutuhan penelitian. Pengambilan data dilakukan pada bulan April hingga Juli 2025 hingga seluruh proses pengumpulan data selesai.

B. Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini meliputi :

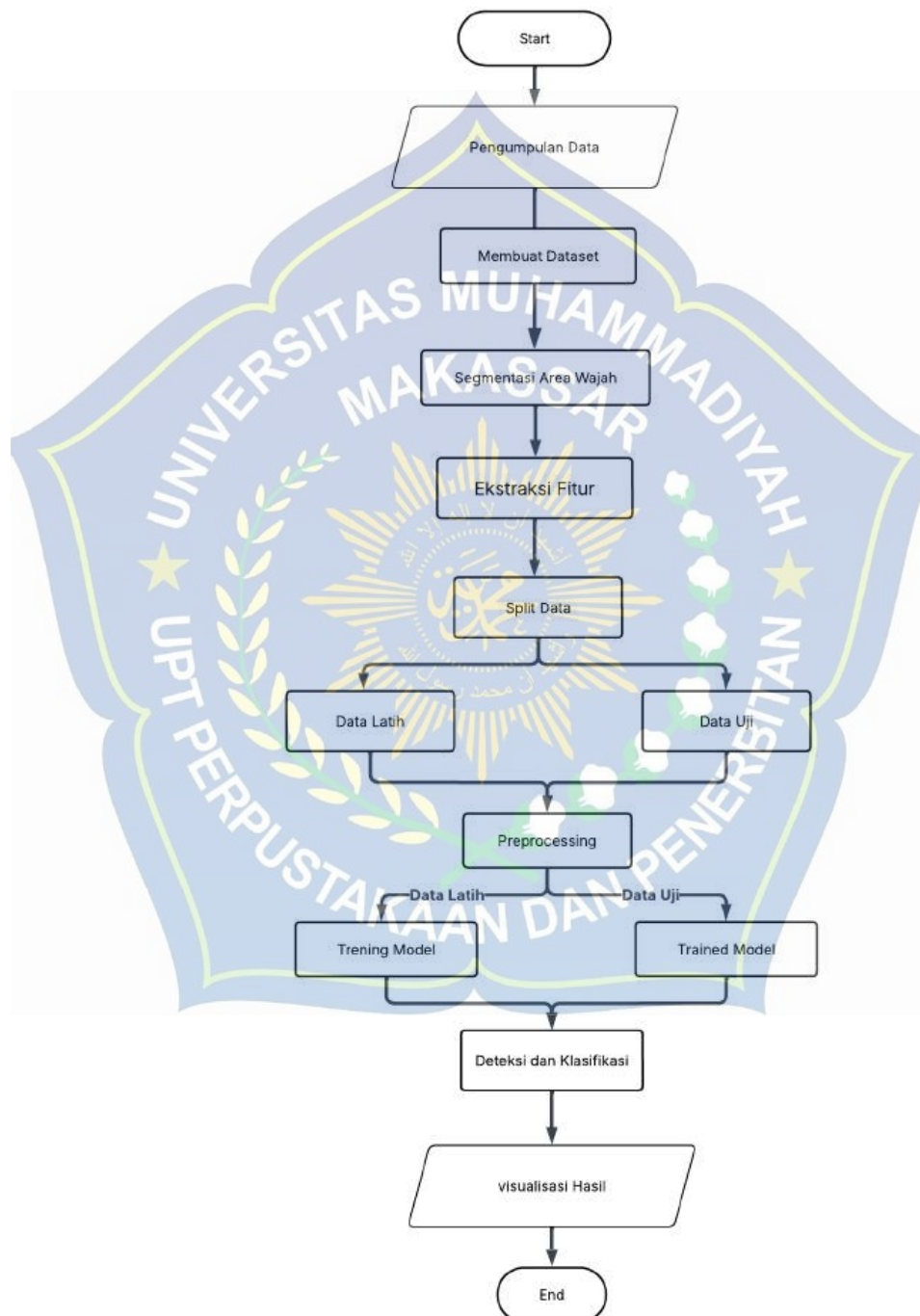
1. Kebutuhan Hardware (Perangkat Keras)
 - a. Laptop ADVAN 1701 dengan spesifikasi:
 - a) Prosesor AMD Ryzen 7 7735HS (16 CPU, ~3.2GHz)
 - b) RAM 16 GB
 - c) Sistem Operasi Windows 11 Home 64-bit
2. Kebutuhan Software (Perangkat Lunak)
 - a. Microsoft Excel
 - b. Visual Studio Code
 - c. Python

C. Perencanaan Sistem

Perancangan sistem merupakan proses yang sangat penting dalam pengembangan perangkat lunak atau sistem informasi. Proses ini melibatkan beberapa langkah dan komponen yang harus diperhatikan untuk memastikan bahwa sistem yang dibangun dapat memenuhi kebutuhan pengguna dan berfungsi dengan baik.

1. Flowchart Perencanaan Sistem

Flowchart digunakan untuk menggambarkan alur proses perencanaan sistem secara visual, sehingga lebih mudah dipahami oleh pengembang maupun pengguna. Diagram ini membantu memperlihatkan urutan langkah, keputusan, serta interaksi antar komponen sistem. Seperti pada Gambar 4 berikut:



Gambar 4. Flowchart Perencanaan Sistem

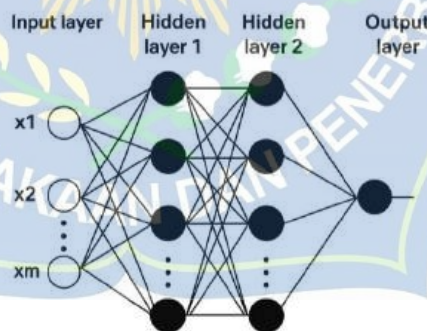
Flowchart pada Gambar 4 menggambarkan langkah-langkah dalam pengembangan sistem deteksi keriput wajah menggunakan metode *Multi-Layer Perceptron* (MLP) berbasis *Backpropagation* :

- a. Start: Merupakan titik awal dari proses pengembangan sistem deteksi dan estimasi penuaan berdasarkan analisis keriput pada wajah.
- b. Pengumpulan Data: Data gambar wajah dikumpulkan dari dataset yang tersedia atau melalui input langsung seperti webcam dan upload gambar, sebagai bahan utama proses pelatihan model.
- c. Membuat Dataset: Gambar wajah yang telah dikumpulkan dikelompokkan berdasarkan tingkat keriput menjadi dua kelas utama (*high* dan *medium*), kemudian disatukan dalam satu dataset.
- d. Segmentasi Area Wajah: Dilakukan pemotongan area spesifik pada wajah seperti dahi, area sekitar mata, pipi, dan area sekitar mulut, yang menjadi fokus utama untuk proses ekstraksi fitur keriput.
- e. Ekstraksi Fitur: Dilakukan perhitungan fitur tekstur menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) dan deteksi fitur kontur menggunakan metode *Canny* serta analisis kontur.
- f. Split Data: Dataset yang telah terbentuk dibagi menjadi data latih dan data uji untuk memastikan model dapat belajar dan diuji secara terpisah guna menghindari *overfitting*.
- g. Preprocessing: Data latih dan uji dinormalisasi dan diskalakan agar berada pada skala yang seragam sehingga proses pelatihan model berjalan optimal.
- h. *Training Model*: Model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dilatih menggunakan data latih melalui proses propagasi maju dan balik untuk meminimalkan kesalahan prediksi.
- i. *Trained Model*: model *machine learning* yang telah melalui proses pelatihan menggunakan data latih hingga memperoleh parameter yang optimal. Dengan demikian, model ini dapat dikatakan telah selesai

belajar dari data latih dan siap digunakan untuk melakukan proses prediksi maupun klasifikasi terhadap data baru.

- j. Deteksi dan Klasifikasi: Model terlatih digunakan untuk mendeteksi dan mengklasifikasikan tingkat keriput pada gambar wajah baru ke dalam dua kategori, yaitu *High* dan *Medium*.
 - k. Visualisasi Hasil: Hasil deteksi dan estimasi ditampilkan dalam bentuk visual seperti grafik, heatmap, atau laporan yang memudahkan pengguna dalam memahami hasil akhir sistem.
 - l. End: Merupakan titik akhir dari proses sistem, di mana seluruh hasil telah dianalisis dan sistem siap digunakan untuk input data baru berikutnya.
2. Arsitektur Model *Multilayer Perceptron* (MLP)

Sebagai bagian dari implementasi metode klasifikasi dalam penelitian ini, digunakan arsitektur jaringan saraf tiruan bertipe *Multilayer Perceptron* (MLP). Arsitektur ini dipilih karena kemampuannya dalam menangani permasalahan klasifikasi non-linear melalui proses pembelajaran berlapis. Konfigurasi jaringan yang digunakan terdiri atas satu lapisan *input*, dua *hidden layers* (lapisan tersembunyi), dan satu lapisan *output*. Struktur lengkap dari arsitektur MLP yang digunakan dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur jaringan *Multilayer Perceptron* (MLP) dengan Algoritma *Backpropagation*

Arsitektur jaringan saraf tiruan tipe *Multi-Layer Perceptron* (MLP) yang digunakan dalam penelitian untuk klasifikasi tingkat keriput wajah. Jaringan ini terdiri dari tiga jenis lapisan utama, yaitu *input layer*, dua *hidden layer*, dan *output layer*. Penjelasan tiap komponen arsitektur adalah sebagai berikut:

a. *Input Layer*

Input layer dalam arsitektur jaringan saraf tiruan ini berfungsi untuk menerima vektor fitur yang telah diekstraksi dari gambar wajah. Fitur-fitur yang digunakan terdiri dari dua jenis, yaitu fitur tekstur dan fitur kontur. Fitur tekstur diperoleh menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) yang diterapkan pada empat area spesifik wajah, yakni dahi, area sekitar mata, pipi, dan area sekitar mulut. Sementara itu, fitur kontur diekstraksi melalui proses deteksi tepi menggunakan metode *Canny*, yang menghasilkan dua parameter utama, yaitu jumlah kontur dan panjang total kontur. Secara keseluruhan, terdapat enam fitur yang digunakan sebagai input, yang kemudian dimasukkan ke dalam jaringan untuk proses klasifikasi tingkat keriput wajah.

b. *Hidden Layer 1*

Hidden layer pertama menerima sinyal dari semua neuron pada *input layer* dan melakukan transformasi linier yang dilanjutkan dengan fungsi aktivasi *Rectified Linear Unit* (ReLU). Lapisan ini berfungsi mengenali pola-pola awal dalam data yang kompleks untuk mempersiapkan representasi fitur yang lebih mendalam.

c. *Hidden Layer 2*

Hidden layer melanjutkan pemrosesan dari layer sebelumnya untuk memperdalam representasi fitur. Fungsi aktivasi yang digunakan juga ReLU. Tujuan lapisan ini adalah memperkuat kemampuan jaringan dalam membedakan fitur antar kelas sehingga menghasilkan klasifikasi yang lebih akurat.

d. *Output Layer*

Output layer terdiri dari satu neuron yang menghasilkan nilai probabilitas atau skor klasifikasi. Nilai ini kemudian dipetakan ke dalam salah satu dari dua kategori, yaitu *High* atau *Medium*, sesuai dengan ambang batas (*threshold*) yang telah ditentukan. Fungsi aktivasi yang digunakan adalah *sigmoid*, sehingga keluaran berada pada rentang 0

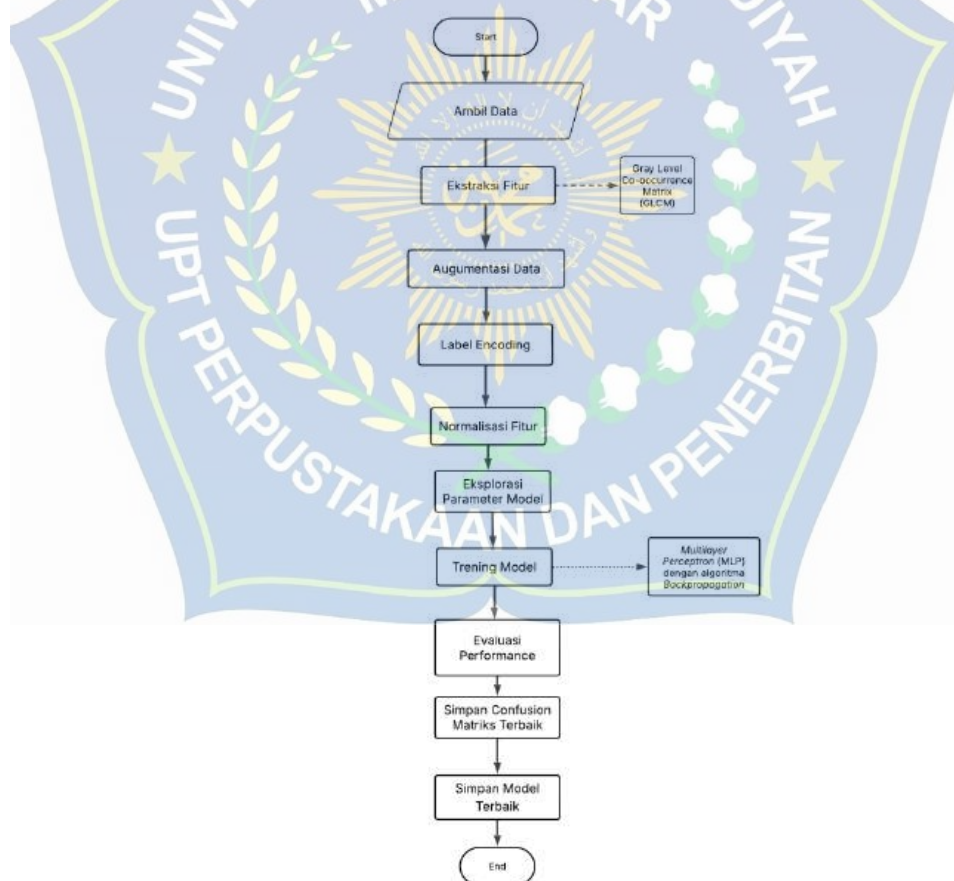
hingga 1, memudahkan proses penentuan kategori akhir.

e. Proses Pelatihan

Sebelum dilatih, data fitur dinormalisasi untuk menyamakan skala antar fitur. Jaringan kemudian dilatih menggunakan algoritma *Backpropagation* untuk meminimalkan nilai loss dan meningkatkan akurasi prediksi. Proses pelatihan dilakukan dengan membagi data menjadi data latih dan data uji.

3. Flowchart Medel MLP dengan Algoritma *Backpropagation*.

Flowchart ini menunjukkan tahapan utama pelatihan dan evaluasi model Multilayer Perceptron (MLP) dengan algoritma *Backpropagation*. Diagram tersebut membantu memvisualisasikan alur proses agar lebih mudah dipahami, seperti pada Gambar 6 berikut.



Gambar 6. Flowchart Pros Proses Pelatihan dan Evaluasi Model MLP dengan Algoritma *Backpropagation*

Berikut disajikan alur kerja pelatihan model klasifikasi tingkat keriput wajah menggunakan arsitektur *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation*. Setiap tahap pada flowchart menggambarkan proses yang dilakukan secara sistematis mulai dari pengambilan data hingga penyimpanan model terbaik.

a. *Start*

Merupakan titik awal proses pelatihan model klasifikasi tingkat keriput wajah menggunakan metode MLP dengan algoritma *Backpropagation*.

b. *Ambil Data*

Dataset fitur dibaca dari file *keriput_dataset.csv* yang berisi data hasil ekstraksi fitur dari gambar wajah dan label tingkat keriput.

c. *Ekstraksi Fitur*

Ekstraksi fitur pada program ini dilakukan dengan mengambil nilai-nilai karakteristik dari area-area spesifik wajah, meliputi dahi, area sekitar mata, pipi, dan area sekitar mulut. Fitur yang diambil terdiri atas dua jenis, yaitu fitur tekstur dan fitur kontur. Untuk memperoleh fitur tekstur, digunakan metode *Gray Level Co-occurrence Matrix* (GLCM) yang membentuk matriks ko-ocurensi tingkat keabuan berdasarkan jarak dan sudut tertentu. Dari matriks ini dihitung tiga properti utama, yaitu kontras (*contrast*), homogenitas (*homogeneity*), dan energi (*energy*). Masing-masing properti dihitung menggunakan rumus berikut:

1. Rumus kontras (*contrast*)

$$Contast = \sum_{i,j} (i - j)^2 \times P(i, j) \dots\dots\dots (5)$$

Ket:

$\sum$: Notasi penjumlahan terhadap seluruh pasangan indeks i, j dalam matriks GLCM

i, j : Indeks baris dan kolom dalam matriks GLCM (merepresentasikan tingkat keabuan)

$P(i,j)$: Nilai probabilitas kemunculan pasangan piksel dengan tingkat keabuan I dan j

2. Rumus homogenitas (*homogeneity*)

$$Homogeneity = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|} \dots\dots\dots (6)$$

Ket:

$\sum$: Notasi penjumlahan terhadap seluruh pasangan indeks i,j dalam matriks GLCM

i,j : Indeks baris dan kolom dalam matriks GLCM (merepresentasikan tingkat keabuan)

$P(i,j)$: Nilai probabilitas kemunculan pasangan piksel dengan tingkat keabuan I dan j

3. Rumus energi (*energy*)

$$Energy = \sum_{i,j} P(i - j)^2 \dots\dots\dots (7)$$

Ket:

$\sum$: Notasi penjumlahan terhadap seluruh pasangan indeks i,j dalam matriks GLCM

i,j : Indeks baris dan kolom dalam matriks GLCM (merepresentasikan tingkat keabuan)

$P(i,j)$: Nilai probabilitas kemunculan pasangan piksel dengan tingkat keabuan I dan j

Nilai akhir fitur tekstur untuk setiap area wajah diperoleh dengan menghitung rata-rata dari ketiga properti tersebut menggunakan rumus:

$$Fitur\ Tekstur = \frac{Contrast + Homogeneity + Energy}{3} \dots\dots (8)$$

Contrast : Nilai fitur variasi lokal

Homogeneity : Nilai fitur keseragaman

Energy : Nilai fitur keteraturan

Fitur Tekstur : Rata-rata dari ketiga fitur untuk mewakili karakteristik tekstur area wajah

Sementara itu, untuk fitur kontur, gambar terlebih dahulu dikonversi ke dalam format grayscale, kemudian dilakukan deteksi tepi menggunakan metode *Canny*. Setelah itu, kontur pada gambar tepi dideteksi menggunakan fungsi `findContours()`. Dari proses ini diperoleh dua fitur, yaitu jumlah kontur dan panjang total kontur, yang dihitung dengan menjumlahkan panjang setiap kontur menggunakan fungsi `arcLength()`.

Seluruh fitur yang diperoleh, baik dari hasil ekstraksi tekstur maupun kontur, digabungkan menjadi sebuah vektor fitur berdimensi enam, yang selanjutnya digunakan sebagai input untuk proses pelatihan pada model *Backpropagation Neural Network* (BPNN).

d. *Augmentasi Data*

Pada tahap augmentasi data, fitur hasil ekstraksi dapat diperbanyak (*data augmentation*) dengan melakukan replikasi atau transformasi tertentu untuk menambah variasi data latih. Proses ini direpresentasikan sebagai:

$$X_{aug} = x^1, x^{1'}, x^{1''}, \dots, x^{1(n)} \dots\dots\dots (9)$$

yang merepresentasikan versi asli dan hasil augmentasi dari satu data fitur.

e. *Label Encoding dan Normalisasi Fitur*

Tahap ini dimulai dengan mengonversi label kategori ke bentuk numerik menggunakan *label encoding*, agar model dapat memproses data dengan efisien. Proses ini dinotasikan sebagai:

$$y_{encoded} = map(y_i) \dots\dots\dots (10)$$

di mana y_i adalah label awal dan $map(\cdot)$ merupakan fungsi pemetaan ke nilai numerik.

Setelah itu, fitur-fitur dinormalisasi menggunakan metode *standardization* untuk memastikan setiap fitur berada dalam skala yang seragam. Proses normalisasi ini menggunakan rumus:

$$x' = \frac{x - \mu}{\sigma} \dots\dots\dots (11)$$

dengan x adalah nilai fitur asli, μ adalah rata-rata, dan σ adalah deviasi standar dari fitur tersebut. Langkah ini penting untuk meningkatkan performa dan stabilitas pelatihan model.

f. *Eksplorasi Parameter Model*

Pada tahapan Eksplorasi Parameter Model, dilakukan percobaan dengan berbagai konfigurasi ukuran *hidden layer* pada *MLPClassifier*, seperti (8,8), (16,8), (16,16), dan lain-lain. Setiap konfigurasi diuji dengan melatih model menggunakan data latih yang sudah diproses, kemudian dievaluasi performanya berdasarkan akurasi pada data uji. Tujuan dari proses ini adalah untuk menemukan arsitektur jaringan yang paling optimal, yaitu yang memberikan hasil klasifikasi terbaik. Dengan mencoba berbagai kombinasi ukuran *hidden layer*, program dapat menentukan konfigurasi yang paling sesuai untuk masalah deteksi keriput ini, sehingga model dapat belajar dengan efektif dan menghasilkan prediksi yang akurat.

g. *Training Model*

Pada tahapan *Training Model*, proses pelatihan dilakukan dengan menggunakan data latih yang telah melalui tahapan pra-pemrosesan. Model dilatih menggunakan algoritma *backpropagation*, yang bekerja dengan cara menghitung output dari jaringan berdasarkan bobot awal, lalu membandingkannya dengan target sebenarnya untuk memperoleh nilai error. Selisih antara output prediksi dan label target ini digunakan sebagai dasar untuk memperbarui bobot jaringan secara iteratif, dengan tujuan meminimalkan nilai error tersebut.

Fungsi aktivasi yang digunakan adalah ReLU (*Rectified Linear Unit*), yang didefinisikan sebagai:

$$f(x) = \max(0, x) \dots\dots\dots (12)$$

Fungsi ini berperan dalam memperkenalkan non-linearitas ke dalam jaringan, sehingga memungkinkan model mempelajari pola-pola kompleks dari data input.

Untuk mengukur tingkat kesalahan prediksi, digunakan *loss function* berupa *Cross-Entropy Loss*, yang umum digunakan pada tugas klasifikasi. Fungsi ini menghitung jarak antara distribusi probabilitas hasil prediksi dan distribusi target yang sebenarnya, serta memberikan penalti yang lebih besar terhadap prediksi yang meleset jauh. Rumus umum dari fungsi *Cross-Entropy Loss* untuk klasifikasi multi-kelas adalah sebagai berikut:

$$L = - \sum_{i=1}^n y_i \log (\hat{y}_i) \dots\dots\dots(13)$$

Ket:

- 1.) L adalah nilai loss,
- 2.) y_i adalah nilai sebenarnya (ground truth),
- 3.) $\hat{y}_i$ adalah probabilitas prediksi dari model,
- 4.) n adalah jumlah kelas.

Nilai *loss* akan semakin kecil jika prediksi mendekati label yang sebenarnya, dan semakin besar jika prediksi jauh dari label.

Proses *backpropagation* selanjutnya menghitung turunan parsial dari fungsi loss terhadap bobot menggunakan kaidah rantai (*chain rule*), lalu memperbarui bobot jaringan dengan rumus:

$$W_{new} = W_{old} - \eta \times \frac{\partial L}{\partial w} \dots\dots\dots(14)$$

Ket:

- 1.) W_{new} adalah bobot baru,
- 2.) W_{old} adalah bobot sebelumnya,
- 3.) $\hat{y}_i$ adalah probabilitas prediksi dari model,
- 4.) η adalah *learning rate*,
- 5.) $\frac{\partial L}{\partial w}$ adalah turunan dari fungsi loss terhadap bobot.

Proses ini diulang selama beberapa *epoch* hingga model mencapai konvergensi atau jumlah *epoch* maksimum tercapai. Melalui pendekatan ini, model secara bertahap belajar mengenali pola pada data latih, sehingga mampu melakukan prediksi dengan akurasi yang tinggi pada data uji.

h. Evaluasi Performa (Precision, Recall, F1-Score)

Setelah model dilatih, tahap evaluasi dilakukan untuk mengukur kinerjanya dalam melakukan klasifikasi. Tiga metrik utama yang digunakan adalah Precision, Recall, dan F1-Score. Ketiganya dihitung berdasarkan nilai True Positive (TP), False Positive (FP), dan False Negative (FN).

i. Confusion Matrix

Confusion matrix merupakan alat evaluasi yang menyajikan informasi tentang performa model klasifikasi dengan membandingkan hasil prediksi terhadap label sebenarnya. Dalam klasifikasi biner, matriks ini tersusun dalam format 2×2 yang mencakup empat komponen utama, yaitu: True Positive (TP), False Negative (FN), False Positive (FP), dan True Negative (TN). Representasi formalnya ditulis sebagai:

$$CM = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \dots\dots\dots (15)$$

Melalui confusion matrix, kita dapat menilai kinerja model secara lebih rinci, seperti mengidentifikasi kesalahan klasifikasi, menghitung akurasi, precision, recall, dan F1-score. Visualisasi matriks ini juga membantu memahami pola kesalahan yang mungkin berulang dalam prediksi model.

j. Simpan Model Terbaik

Setelah proses pelatihan dan evaluasi selesai, model dengan performa terbaik disimpan untuk digunakan kembali tanpa perlu dilatih ulang. Dalam penelitian ini, proses penyimpanan dilakukan menggunakan pustaka joblib, yang menyimpan objek model MLP, objek *scaler* untuk *normalisasi fitur*, dan *encoder label* dalam bentuk file .pkl. Penyimpanan ini memungkinkan proses inferensi atau prediksi dilakukan di masa mendatang secara efisien tanpa perlu memulai dari awal.

k. *End*

Tahapan *End* menandai titik akhir dari keseluruhan proses pelatihan model. Setelah model selesai dievaluasi dan hasil terbaik disimpan baik dalam bentuk *confusion matrix*, *parameter model*, maupun objek model itu sendiri proses pelatihan dianggap selesai. Model yang telah dilatih dan disimpan ini selanjutnya dapat digunakan untuk melakukan prediksi pada data baru, seperti gambar wajah lain yang belum pernah dilihat sebelumnya. Tahap ini juga menjadi penutup dari alur kerja mulai dari akuisisi data, ekstraksi fitur, hingga evaluasi performa klasifikasi.

D. Teknik Analisis Data

Teknik analisis data dalam penelitian ini bertujuan untuk mengolah, menginterpretasikan, dan mengevaluasi hasil keluaran dari model *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam mendeteksi distribusi keriput wajah berbasis segmentasi area spesifik. Analisis data dilakukan untuk mengetahui sejauh mana model mampu mengenali pola keriput pada area wajah seperti dahi, area sekitar mata, pipi, dan area sekitar mulut, serta mengevaluasi performa prediksi yang dihasilkan. Tahap awal analisis data dimulai dengan pengolahan hasil prediksi model, di mana output prediksi dari jaringan MLP dibandingkan dengan data target aktual (*ground truth*) yang terdapat dalam dataset. Data target ini merupakan label atau anotasi kategori keriput (*High* atau *Medium*) yang telah divalidasi secara manual dalam dataset penelitian.

Selanjutnya, dilakukan perhitungan metrik evaluasi *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix* berdasarkan hasil prediksi dan data target. Metrik ini dihitung secara terpisah untuk setiap area wajah guna mengetahui performa model dalam mendeteksi distribusi keriput pada area spesifik. Nilai-nilai tersebut kemudian dirata-ratakan untuk memperoleh gambaran umum performa model. Analisis *Confusion Matrix* juga digunakan untuk memahami pola kesalahan prediksi, seperti salah klasifikasi kategori atau deviasi dari target.

Selain analisis performa prediksi, penelitian ini juga mengevaluasi efisiensi sistem melalui pengukuran waktu pelatihan, waktu prediksi, dan penggunaan memori selama proses pelatihan dan inferensi. Hasil analisis digunakan untuk menilai efisiensi sistem serta potensi penerapan model dalam aplikasi nyata yang membutuhkan kecepatan dan efisiensi sumber daya. Semua hasil, baik dari aspek akurasi maupun efisiensi, disajikan dalam bentuk tabel, grafik, dan visualisasi hasil prediksi guna memudahkan interpretasi. Dengan pendekatan analisis data yang komprehensif ini, penelitian diharapkan dapat menyimpulkan kemampuan model MLP secara objektif dalam mendeteksi distribusi keriput wajah secara *personalized* serta mendukung penerapannya dalam bidang dermatologi, kosmetik, maupun forensik.

E. Metode Evaluasi Model

Evaluasi model dalam penelitian ini bertujuan untuk mengetahui seberapa baik metode *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* dalam mengidentifikasi tingkat keriput wajah pada area spesifik, seperti dahi, area sekitar mata, pipi, dan area sekitar mulut. Evaluasi ini dilakukan untuk menilai performa model dari segi akurasi dan ketepatan dalam memprediksi label kategori (*High* atau *Medium*) sesuai dengan kondisi sebenarnya.

Metrik yang digunakan dalam proses evaluasi meliputi *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix*, yang umum digunakan dalam klasifikasi citra. *Precision* mengukur tingkat ketepatan model dalam memprediksi kategori yang benar, sedangkan *Recall* mengukur sejauh mana model mampu mendeteksi seluruh data yang termasuk dalam kategori tersebut. *F1-Score* digunakan sebagai rata-rata harmonis dari *Precision* dan *Recall*, sehingga memberikan penilaian seimbang, terutama pada dataset dengan distribusi kelas yang tidak seimbang. *Confusion Matrix* digunakan untuk memvisualisasikan jumlah prediksi benar dan salah pada masing-masing kategori (*High* dan *Medium*).

Dengan evaluasi ini, diharapkan model MLP yang dibangun tidak hanya mampu menghasilkan prediksi yang akurat, tetapi juga bekerja secara konsisten dan andal pada data uji yang belum pernah dilihat sebelumnya.



BAB IV

HASIL DAN PEMBAHASAN

Penerapan teknologi pengolahan citra dan kecerdasan buatan dalam bidang analisis wajah telah memberikan kontribusi signifikan dalam berbagai sektor, salah satunya dalam pemetaan dan identifikasi keriput wajah. Sebagai bagian dari upaya tersebut, penelitian ini merancang dan mengimplementasikan sebuah sistem klasifikasi keriput berbasis jaringan saraf tiruan menggunakan algoritma *Multilayer Perceptron* (MLP) dengan *Backpropagation*. Sistem ini difokuskan pada area-area spesifik wajah, yaitu dahi, area sekitar mata, pipi, dan area sekitar mulut, untuk memetakan distribusi keriput dan mengidentifikasi tingkat keparahannya yang terbagi dalam dua kelas, yaitu *high* dan *medium*.

Bab ini memaparkan hasil implementasi proses klasifikasi tingkat keriput, yang mencakup tahapan ekstraksi fitur tekstur menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM), deteksi fitur kontur menggunakan metode Canny, normalisasi data, serta pelatihan dan pengujian model MLP. Fitur-fitur yang digunakan telah dirancang untuk menangkap karakteristik visual yang signifikan dari keriput, baik dari segi tekstur maupun bentuk. Proses pelatihan model dilakukan menggunakan dataset citra wajah yang telah dilabeli berdasarkan kategori keriput, dengan input berupa nilai-nilai numerik hasil ekstraksi fitur.

Pembahasan lebih lanjut dalam bab ini mencakup evaluasi performa model klasifikasi berdasarkan metrik akurasi, *presisi*, *recall* dan *f1-score*, serta interpretasi hasil klasifikasi untuk masing-masing area wajah. Dengan demikian, bab ini memberikan gambaran menyeluruh mengenai efektivitas sistem yang dikembangkan dalam melakukan pemetaan dan identifikasi keriput wajah secara otomatis.

A. Gambaran Umum Data Penelitian

1. Deskripsi Data Wajah

Dataset penelitian ini terdiri dari 319 sampel wajah yang telah melalui proses seleksi dan pra-pemrosesan, kemudian dikategorikan ke dalam dua kelas utama, yaitu Medium (200 sampel) yang mewakili tingkat keriput sedang, serta High (119 sampel) yang merepresentasikan tingkat keriput tinggi. Untuk memastikan model machine learning dapat dilatih dan diuji secara optimal, data dibagi menjadi dua bagian dengan proporsi 80% untuk training (pelatihan) dan 20% untuk testing (pengujian). Rincian distribusi kelas dan pembagian data dapat dilihat pada Tabel 3.

Table 3. Distribusi Kelas dan Pembagian Data

Kelas	Jumlah Sampel	Persentase (%)	Data Training (80%)	Data Testing (20%)
Medium	200	62.7	160	40
High	119	37.3	95	24
Total	319	100	255	64

Berdasarkan Tabel 3, terlihat bahwa kelas Medium memiliki jumlah sampel yang lebih banyak dibandingkan kelas High. Perbedaan distribusi ini penting untuk diperhatikan agar model tidak bias terhadap kelas mayoritas pada saat pelatihan.

Dataset ini dirancang agar memiliki keberagaman yang tinggi sehingga model dapat beradaptasi pada berbagai kondisi wajah di dunia nyata. Dari segi usia, sampel mencakup rentang dari 20 – 45 tahun, dengan variasi nilai fitur yang cukup signifikan, misalnya pada area dahi yang berkisar antara 0.545 hingga 434.217. Dari aspek pencahayaan, gambar diperoleh dalam kondisi yang bervariasi, meliputi pencahayaan alami (*outdoor*), pencahayaan buatan (*indoor*), serta berbagai tingkat intensitas cahaya yang memberikan tantangan tersendiri dalam mendeteksi keriput. Selain itu, variasi ekspresi wajah juga diperhatikan, mulai dari ekspresi netral tanpa senyum, senyum ringan, hingga ekspresi wajah alami yang lebih dinamis. Faktor ini turut memengaruhi tekstur dan distribusi keriput, yang terlihat dari perbedaan jumlah kontur keriput pada citra, yakni antara

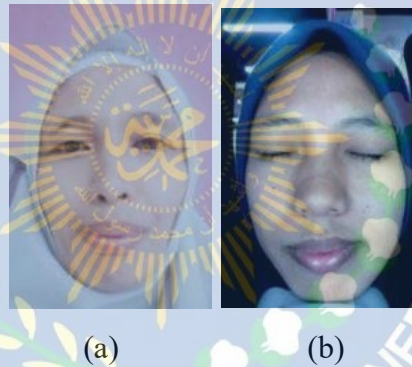
19 hingga 393 garis. Gambaran distribusi jumlah sampel pada masing-masing kelas dapat dilihat pada Tabel 4.

Table 4. Deskripsi dan Variasi Kondisi Dataset Wajah

Aspek Variasi	Kategori / Rentang	Keterangan
Usia	20 – 45 tahun	Mencakup berbagai tahap penuaan kulit wajah
Area Dahi	0.545 – 434.217	Rentang nilai fitur intensitas keriput pada area dahi
Area Mata	1.05 – 239.35	Rentang nilai fitur intensitas keriput pada area mata
Area Pipi	0.99 – 104.24	Rentang nilai fitur intensitas keriput pada area pipi
Area Mulut	0.52 – 206.99	Rentang nilai fitur intensitas keriput pada area mulut
Pencahaya-an	Alami (outdoor), buatan (indoor), variasi intensitas	Menyimulasikan kondisi pengambilan gambar di dunia nyata
Ekspresi	Netral, senyum ringan, ekspresi alami	Memengaruhi distribusi dan tekstur keriput
Jumlah Kontur Keriput	19 – 393	Menggambarkan kompleksitas pola keriput pada citra
Panjang Total Kontur	1867.01 – 14532.42	Total panjang semua kontur keriput dalam piksel

2. Proses Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh melalui akuisisi citra wajah secara langsung di lingkungan Universitas Muhammadiyah Makassar dan beberapa lokasi lainnya di wilayah Kota Makassar, Sulawesi Selatan. Pengambilan data dilakukan dengan mempertimbangkan keragaman kondisi, mencakup rentang usia subjek 20–45 tahun, variasi pencahayaan, serta ekspresi wajah, guna memastikan representativitas *dataset*. Proses pemotretan difokuskan pada area wajah tanpa penerapan *filter* maupun efek digital untuk menjaga keaslian data. Secara keseluruhan, diperoleh 319 citra wajah yang kemudian diklasifikasikan ke dalam dua kategori tingkat keriput, yaitu *Medium* dan *High*. Contoh visualisasi dari masing-masing kategori ditunjukkan pada Gambar 7.



Gambar 7. (a) Kategori keriput High (b) Kategori keriput medium

3. Pra-pemrosesan Data

Tahap *preprocessing* citra wajah dalam penelitian ini merupakan langkah penting untuk menyiapkan dataset sebelum analisis distribusi keriput. Proses diawali dengan konversi citra ke format *grayscale* untuk memfokuskan analisis pada tekstur keriput, sehingga informasi warna yang tidak relevan tidak memengaruhi hasil. Selanjutnya, *Contrast Limited Adaptive Histogram Equalization* (CLAHE) diterapkan guna meningkatkan kontras lokal, sehingga fitur keriput yang halus dapat terdeteksi dengan lebih jelas.

Untuk mengurangi *noise* dan artefak yang tidak diinginkan, dilakukan *Gaussian blur*, diikuti oleh *adaptive thresholding* untuk segmentasi biner yang adaptif terhadap variasi pencahayaan pada setiap area wajah. *Canny edge detection* kemudian diterapkan untuk mengekstraksi kontur keriput secara presisi, yang menjadi dasar analisis pola keriput pada dahi, mata, pipi, dan mulut.

Selanjutnya, citra di-*resize* secara otomatis menjadi 200×200 piksel untuk memastikan keseragaman dimensi seluruh dataset, yang penting bagi proses pelatihan model *machine learning*. Tahap akhir meliputi operasi morfologi, seperti erosi, dilasi, dan *closing*, untuk membersihkan hasil deteksi tepi dari artefak kecil. Dengan demikian, hasil *preprocessing* adalah citra wajah yang bersih, kontras, dan memiliki dimensi konsisten, siap untuk tahap ekstraksi fitur dan pelatihan model deteksi keriput berbasis *MLP* dengan algoritma *backpropagation*. Seperti pada Gambar 8 berikut:



Gambar 8. Pra-pemrosesan Data

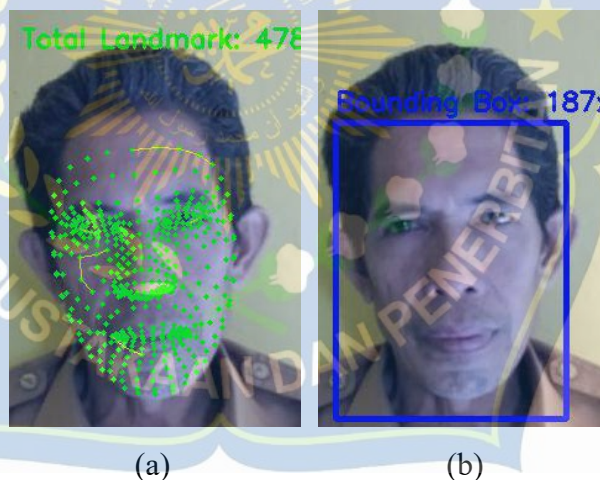
B. Implementasi Sistem

1. Penerapan MediaPipe untuk Deteksi Landmark Wajah

Sistem deteksi keriput dalam penelitian ini memanfaatkan *MediaPipe FaceMesh* untuk melakukan identifikasi otomatis terhadap 468 titik *landmark* wajah, mencakup seluruh area wajah mulai dari dahi, mata, pipi, hingga mulut. Titik-titik *landmark* ini berfungsi sebagai referensi utama dalam proses segmentasi wajah dan pemetaan area keriput.

Proses dimulai dengan deteksi *landmark* kunci pada posisi 10 (puncak kepala), 152 (dagu), 234 (sisi kiri wajah), dan 454 (sisi kanan wajah), yang digunakan untuk menentukan *bounding box* wajah secara otomatis. Berdasarkan titik-titik tersebut, wajah kemudian dibagi menjadi empat region utama, yaitu: area dahi (titik 10–338–297–332), area mata (titik di sekitar mata), area pipi (titik 234–454), dan area mulut (titik 61–291).

Selanjutnya, fitur non-relevan seperti mata, alis, bibir, dan hidung dihilangkan menggunakan teknik *masking* berbasis *landmark*, sehingga setiap region wajah yang relevan dapat dianalisis secara individual. Setiap area diproses dengan parameter khusus untuk mendeteksi keriput secara akurat. Pendekatan ini memungkinkan analisis distribusi keriput pada setiap region wajah dengan presisi tinggi dan kapasitas pemrosesan secara real-time, mendukung tujuan penelitian dalam pengembangan sistem deteksi keriput yang andal. Seperti pada Gambar 9 berikut:



Gambar 9. (a) Deteksi landmark wajah (b) Bounding box wajah

2. Ekstraksi Fitur Tekstur dengan GLCM

Setelah citra wajah melalui tahap segmentasi menggunakan *MediaPipe* untuk mendeteksi titik-titik (*landmark*) wajah, dilakukan proses ekstraksi fitur guna memperoleh representasi numerik karakteristik keriput pada area spesifik, yaitu dahi, area sekitar mata, pipi, dan area sekitar mulut.

Ekstraksi fitur tekstur dilakukan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) pada setiap area yang telah disegmentasi. Dari matriks ko-occurrence yang terbentuk, dihitung tiga parameter utama:

- 1.) Contrast mengukur perbedaan intensitas antar piksel untuk mendeteksi variasi tekstur halus pada kulit.
- 2.) Homogeneity menunjukkan tingkat keseragaman distribusi piksel untuk mengidentifikasi area dengan tekstur lebih halus.
- 3.) Energy mencerminkan kekompakan nilai piksel dalam citra untuk mengukur tingkat kerapatan tekstur.

Nilai parameter GLCM ini dihitung terpisah untuk setiap area wajah, kemudian digabungkan menjadi vektor fitur lengkap. Pada dataset, hasil perhitungan tersebut direpresentasikan oleh kolom dahi, area sekitar mata, pipi, dan area sekitar mulut.

Selain itu, dilakukan ekstraksi fitur kontur menggunakan algoritma *Canny Edge Detection* untuk mendeteksi tepi pada citra, diikuti fungsi `findContours()` untuk menemukan batas-batas kontur. Dari hasil deteksi ini diambil dua fitur utama:

- 1.) Jumlah kontur menunjukkan banyaknya lekukan atau garis halus yang terdeteksi pada permukaan kulit.
- 2.) Panjang total kontur dihitung dengan fungsi `arcLength()` dan merepresentasikan total panjang semua garis keriput yang terdeteksi.

Kedua fitur ini direpresentasikan dalam dataset oleh kolom `jumlah_kontur` dan `panjang_total_kontur`. Seluruh nilai fitur kemudian diberi label kategori keriput (label) sesuai tingkatannya, misalnya *high* atau *medium*, seperti table 5.

Table 5. Dataset hasil ekstraksi fitur

no	dahi	mata	pipi	mulut	jumlah_kontur	Panjang_total_kontur	label
1	9.458535302	11.73905844	56.69150586	6.137634705	104	7006.379506	high
2	53.42120412	18.77966402	51.88373256	21.80742805	130	8489.552295	high
3	26.49508479	33.97528225	42.37280654	7.03893965	59	3827.862371	high

no	dahi	mata	pipi	mulut	jumlah_kontur	Panjang_total_kontur	label
4	0.637069025	1.436498843	3.990393975	0.789740739	23	2843.289658	high
5	48.95521018	23.51646934	88.8749034	46.28075764	230	12771.11018	high
6	1.448001656	2.39972828	4.205955264	0.758720672	28	2582.429704	high
7	1.037368067	1.308893374	2.316368092	0.698898606	33	2487.179922	high
8	5.838920642	3.666908215	13.4655251	2.502548553	35	4556.76395	high
9	2.218350998	4.233791155	9.698365658	0.519152845	41	3980.451124	high
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
286	13.70504926	19.86381643	23.46022082	15.9673374	50	3162.456179	medium
287	5.422232977	6.958969751	4.068622732	3.859683492	67	4042.323269	medium
288	4.961971024	4.879573539	5.092160882	4.736401384	84	4959.62896	medium
289	80.90525998	39.10994296	214.136022	59.69424599	393	14532.42005	medium
280	82.78603785	44.40987614	89.60733738	16.14916978	303	12549.6249	medium
281	41.57254713	83.94885059	49.80234031	19.20786468	133	7168.131817	medium
282	30.91046294	21.07656107	143.6963312	9.80707491	120	9224.612051	medium
283	31.91694668	26.01237871	7.868570671	23.63554086	57	5651.660487	medium
284	32.06586814	34.0402485	4.299271837	13.23738837	103	8371.70036	medium
285	6.605047847	10.32913539	20.87457371	6.095917958	168	8380.329864	medium

Keakuratan nilai dalam dataset diverifikasi melalui perhitungan manual pada baris pertama. Hasil ekstraksi fitur tekstur menggunakan metode Gray Level Co-occurrence Matrix (GLCM) pada masing-masing area wajah diperoleh sebagai berikut, seperti ditampilkan pada Tabel 6, 7, 8, 9, 10, dan 11 berikut ini.

a) Perhitungan Area Dahi

Table 6. Perhitungan Area Dahi

Pasangan Pixel (i,j)	(i,j)	(i,j) ²	p(i,j)	Kontras ((i-j) ² *P(i,j))	Homogenitas (P(i,j)/(1+ i-j))	Energi (P(i,j) ²)
(120,125)	5	25	0.012	0.3	0.002	0.000144
(120,118)	2	4	0.008	0.032	0.002667	6.4e-05
(123,119)	4	16	0.015	0.24	0.003	0.000225
(118,122)	4	16	0.014	0.224	0.0028	0.000196
(125,121)	4	16	0.013	0.208	0.0026	0.000169
(124,119)	5	25	0.011	0.275	0.001833	0.000121
TOTAL	-	-	-	15.234	8.765	4.376

$$\text{Fitur Tekstur} = \frac{\text{Contast} + \text{Homogeneity} + \text{Energy}}{3}$$

$$\text{Fitur Tekstur} = \frac{15.234 + 8.765 + 4.376}{3}$$

$$\text{Fitur Tekstur} = 9.458535302$$

b) Perhitungan Area Mata

Table 7. Perhitungan Area Mata

Pasangan Pixel (i,j)	(i,j)	(i,j) ²	p(i,j)	Kontras ((i-j) ² *P(i,j))	Homogenitas (P(i,j)/(1+ i-j))	Energi (P(i,j) ²)
(115,120)	5	25	0.018	0.45	0.003	0.000324
(115,112)	3	9	0.009	0.081	0.00225	8.1e-05
(118,115)	3	9	0.022	0.198	0.0055	0.000484
(112,118)	6	36	0.016	0.576	0.002286	0.000256
(120,115)	5	25	0.019	0.475	0.003167	0.000361
(121,118)	3	9	0.014	0.126	0.0035	0.000196
TOTAL	-	-	-	18.456	10.892	5.869

$$\text{Fitur Tekstur} = \frac{\text{Contast} + \text{Homogeneity} + \text{Energy}}{3}$$

$$\text{Fitur Tekstur} = \frac{18.456 + 10.892 + 5.869}{3}$$

$$\text{Fitur Tekstur} = 11.73905844$$

c) Perhitungan Area Pipi

Table 8. Perhitungan Areaa Pipi

Pasangan Pixel (i,j)	(i,j)	(i,j) ²	p(i,j)	Kontras ((i-j) ² *P(i,j))	Homogenitas (P(i,j)/(1+ i-j))	Energi (P(i,j) ²)
(105,110)	5	25	0.035	0.875	0.005833	0.001225
(105,100)	5	25	0.028	0.7	0.004667	0.000784
(108,103)	5	25	0.042	1.05	0.007	0.001764
(100,108)	8	64	0.031	1.984	0.003444	0.000961
(110,105)	5	25	0.037	0.925	0.006167	0.001369
(112,107)	5	25	0.038	0.95	0.006333	0.001444
TOTAL	-	-	-	87.345	42.876	39.853

$$\text{Fitur Tekstur} = \frac{\text{Contast} + \text{Homogeneity} + \text{Energy}}{3}$$

$$Fitur\ Tekstur = \frac{87.345 + 42.876 + 39.853}{3}$$

$$Fitur\ Tekstur = 56.69150586$$

d) Perhitungan Area Mulut

Table 9. Perhitungan Area Mulut

Pasangan Pixel (i,j)	(i,j)	(i,j) ²	p(i,j)	Kontras (i-j ² *P(i,j))	Homogenitas (P(i,j)/(1+ i-j))	Energi (P(i,j) ²)
(130,128)	2	4	0.007	0.028	0.002333	4.9e-05
(130,125)	5	25	0.005	0.125	0.000833	2.5e-05
(127,124)	3	9	0.009	0.081	0.00225	8.1e-05
(128,130)	2	4	0.008	0.032	0.002667	6.4e-05
(125,127)	2	4	0.006	0.024	0.002	3.6e-05
(129,126)	3	9	0.008	0.072	0.002	6.4e-05
TOTAL	-	-	-	9.234	5.876	3.303

$$Fitur\ Tekstur = \frac{Contast + Homogeneity + Energy}{3}$$

$$Fitur\ Tekstur = \frac{9.234 + 5.876 + 3.303}{3}$$

$$Fitur\ Tekstur = 6.137634705$$

e) Tabel Kontur Wajah

Table 10. Tabel Kontur Wajah

Parameter	Nilai	Rumus
Jumlah Kontur	104	len(contours)
Panjang Total Kontur	7006.379506	$\sum cv2.arcLength(cnt, False)$

f) Ringkasan Fitur Semua Area

Table 11. Ringkasan Fitur Semua Area

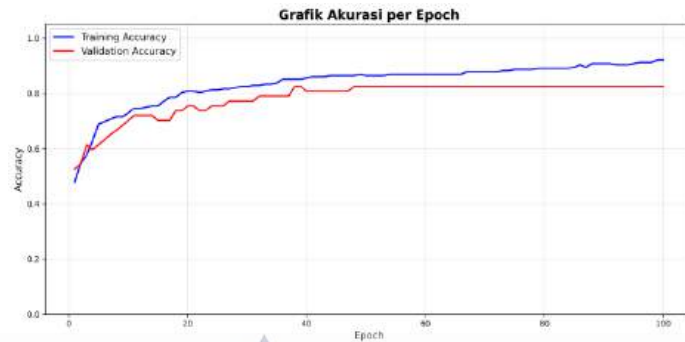
Area	Kontras (C)	Homogenitas (H)	Energi (E)	Nilai Fitur
Dahi	15.234	8.765	4.376	9.4585
Mata	18.456	10.892	5.869	11.7391
Pipi	87.345	42.876	39.853	56.6915
Mulut	9.234	5.876	3.303	6.1376

Area	Kontras (C)	Homogenitas (H)	Energi (E)	Nilai Fitur
Kontur	-	-	-	104 (jumlah), 7006.38 (panjang)

3. Klasifikasi Menggunakan *Multi-Layer Perceptron* (MLP)

Sistem klasifikasi tingkat keriput wajah pada penelitian ini dirancang menggunakan arsitektur jaringan saraf tiruan jenis *Multi-Layer Perceptron* (MLP) dengan dua kelas output, yaitu *medium* dan *high*, yang merepresentasikan tingkat keparahan keriput. Jaringan terdiri dari enam neuron pada lapisan input, masing-masing merepresentasikan fitur hasil ekstraksi citra wajah, yaitu area dahi, mata, pipi, mulut, jumlah kontur, dan panjang total kontur. Berdasarkan hasil eksperimen grid search, konfigurasi terbaik diperoleh dengan dua hidden layer yang masing-masing terdiri atas 16 neuron. Fungsi aktivasi yang digunakan adalah *Rectified Linear Unit* (ReLU) pada hidden layer untuk meningkatkan efisiensi komputasi dan mengatasi masalah *vanishing gradient*, serta sigmoid pada output layer yang sesuai untuk klasifikasi biner. Proses pelatihan model dilakukan menggunakan algoritma Adam dengan nilai *learning rate* sebesar 0,001 dan jumlah iterasi maksimum 1000. Untuk mencegah *overfitting*, diterapkan teknik *early stopping* dengan batas 50 iterasi tanpa peningkatan performa, serta pembagian data validasi sebesar 20% dari total data latih. Model ini menunjukkan performa terbaik dengan akurasi klasifikasi sebesar 84,21% dan waktu pelatihan relatif cepat, yaitu 0,71 detik. Hasil ini menunjukkan bahwa konfigurasi MLP yang digunakan mampu mengklasifikasikan tingkat keriput wajah secara efektif dan efisien.

Berdasarkan hasil pengujian terhadap konfigurasi arsitektur jaringan MLP, proses pelatihan menunjukkan perubahan nilai akurasi dan *loss* pada data pelatihan maupun validasi seiring bertambahnya jumlah *epoch*, sebagaimana ditampilkan pada Gambar 10.

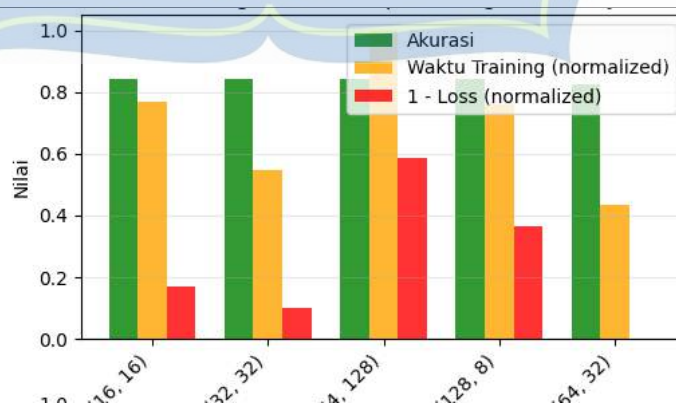


Gambar 10. Grafik akurasi dan loss per-epoch pada model MLP

Grafik tersebut memperlihatkan bahwa model mengalami peningkatan performa secara bertahap hingga mencapai titik konvergensi. Dari seluruh konfigurasi yang diuji, arsitektur dengan dua hidden layer berukuran (16, 16) menghasilkan akurasi tertinggi sebesar 84,21% dengan waktu pelatihan relatif cepat, yaitu 0,71 detik. Konfigurasi ini ditetapkan sebagai yang paling optimal dan ditampilkan pada Tabel 12 dan Gambar 11.

Table 12. Hasil Pengujian Konfigurasi (16, 16)

Parameter	High	Medium	Accuracy	Macro Avg	Weighted Avg
Precision	0.80	0.88	0.84	0.84	0.84
Recall	0.83	0.85	0.84	0.84	0.84
F1-Score	0.82	0.86	0.84	0.84	0.84
Support	24	33	57	57	57



Gambar 11. Gambar Perbandingan metrik top 5 konfigurasi

C. Hasil Pemetaan Keriput Wajah

1. Visualisasi Peta Keriput

Visualisasi peta keriput dilakukan untuk memberikan gambaran spasial mengenai distribusi dan intensitas garis keriput pada wajah individu dari berbagai rentang usia. Deteksi garis keriput dilakukan melalui proses ekstraksi kontur menggunakan metode pemrosesan citra, yang kemudian divisualisasikan dalam bentuk peta kontur pada area wajah seperti dahi, mata, pipi, dan mulut. Contoh hasil deteksi garis keriput pada beberapa individu dengan variasi usia serta kondisi lingkungan yang berbeda ditampilkan pada Tabel 13, Tabel 14, dan Tabel 15, yang memuat informasi mengenai usia, area dominan munculnya keriput, jumlah kontur, panjang total kontur, dan nilai intensitas rata-rata.

Table 13. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Pegunungan

Gambar	Usia															
Hasil Deteksi Keriput <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>Not</td><td>15.48</td></tr><tr><td>Mata</td><td>Not</td><td>22.78</td></tr><tr><td>Pipi</td><td>Not</td><td>21.94</td></tr><tr><td>Mulut</td><td>Not</td><td>22.17</td></tr></tbody></table>	REGION	STATUS	NILAI FITUR	Dahi	Not	15.48	Mata	Not	22.78	Pipi	Not	21.94	Mulut	Not	22.17	22 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	Not	15.48														
Mata	Not	22.78														
Pipi	Not	21.94														
Mulut	Not	22.17														
Hasil Deteksi Keriput <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>Not</td><td>18.62</td></tr><tr><td>Mata</td><td>Not</td><td>23.86</td></tr><tr><td>Pipi</td><td>Not</td><td>18.29</td></tr><tr><td>Mulut</td><td>Not</td><td>22.75</td></tr></tbody></table>	REGION	STATUS	NILAI FITUR	Dahi	Not	18.62	Mata	Not	23.86	Pipi	Not	18.29	Mulut	Not	22.75	34 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	Not	18.62														
Mata	Not	23.86														
Pipi	Not	18.29														
Mulut	Not	22.75														

Gambar	Usia															
Hasil Deteksi Keriput  <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>High</td><td>1439.95</td></tr><tr><td>Mata</td><td>High</td><td>821.18</td></tr><tr><td>Pipi</td><td>High</td><td>575.81</td></tr><tr><td>Mulut</td><td>High</td><td>638.58</td></tr></tbody></table> Kembali	REGION	STATUS	NILAI FITUR	Dahi	High	1439.95	Mata	High	821.18	Pipi	High	575.81	Mulut	High	638.58	47 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	High	1439.95														
Mata	High	821.18														
Pipi	High	575.81														
Mulut	High	638.58														
Hasil Deteksi Keriput  <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>High</td><td>792.78</td></tr><tr><td>Mata</td><td>High</td><td>458.65</td></tr><tr><td>Pipi</td><td>High</td><td>362.14</td></tr><tr><td>Mulut</td><td>High</td><td>358.74</td></tr></tbody></table> Kembali	REGION	STATUS	NILAI FITUR	Dahi	High	792.78	Mata	High	458.65	Pipi	High	362.14	Mulut	High	358.74	52 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	High	792.78														
Mata	High	458.65														
Pipi	High	362.14														
Mulut	High	358.74														

Table 14. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Pesisir

Gambar	Usia															
Hasil Deteksi Keriput  <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>High</td><td>32.11</td></tr><tr><td>Mata</td><td>High</td><td>47.19</td></tr><tr><td>Pipi</td><td>High</td><td>36.14</td></tr><tr><td>Mulut</td><td>High</td><td>19.28</td></tr></tbody></table> Kembali	REGION	STATUS	NILAI FITUR	Dahi	High	32.11	Mata	High	47.19	Pipi	High	36.14	Mulut	High	19.28	20 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	High	32.11														
Mata	High	47.19														
Pipi	High	36.14														
Mulut	High	19.28														

Gambar	Usia
Hasil Deteksi Keriput 	38 Tahun
Hasil Deteksi Keriput 	45 Tahun
Hasil Deteksi Keriput 	55 Tahun

Table 15. Contoh hasil deteksi garis keriput pada berbagai usia di wilayah Perkotaan

Gambar	Usia
Hasil Deteksi Keriput 	23 Tahun
Hasil Deteksi Keriput 	36 Tahun
Hasil Deteksi Keriput 	49 Tahun

Gambar	Usia															
Hasil Deteksi Keriput  <table><thead><tr><th>REGION</th><th>STATUS</th><th>NILAI FITUR</th></tr></thead><tbody><tr><td>Dahi</td><td>high</td><td>969.87</td></tr><tr><td>Mata</td><td>high</td><td>1104.82</td></tr><tr><td>Pipi</td><td>high</td><td>878.66</td></tr><tr><td>Mulut</td><td>high</td><td>816.84</td></tr></tbody></table> Kontrol	REGION	STATUS	NILAI FITUR	Dahi	high	969.87	Mata	high	1104.82	Pipi	high	878.66	Mulut	high	816.84	59 Tahun
REGION	STATUS	NILAI FITUR														
Dahi	high	969.87														
Mata	high	1104.82														
Pipi	high	878.66														
Mulut	high	816.84														

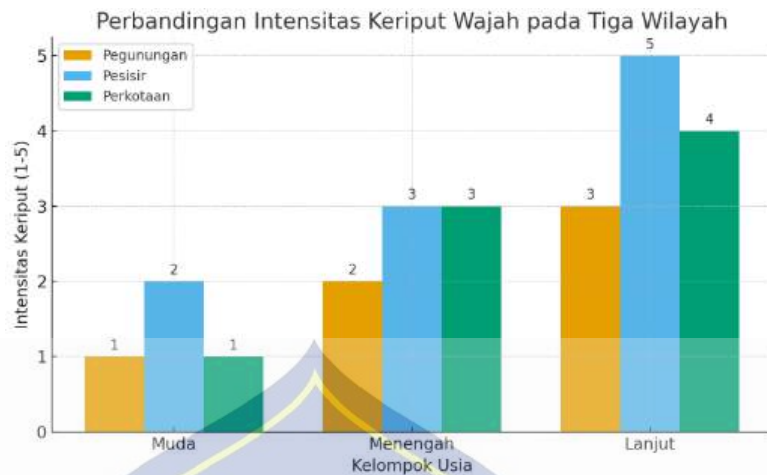
Hasil deteksi garis keriput pada wilayah pegunungan (Tabel 13) menunjukkan bahwa keriput cenderung muncul lebih lambat dibandingkan dengan wilayah lain. Pada kelompok usia muda, pola keriput masih sangat minim dan umumnya terbatas di sekitar mata. Seiring bertambahnya usia, keriput mulai terlihat pada dahi dan pipi, namun dengan intensitas yang masih moderat. Bahkan pada usia lanjut, keriput memang terlihat lebih dalam dan merata, tetapi tingkat keparahannya tetap lebih ringan dibandingkan wilayah pesisir dan perkotaan. Hal ini dapat dikaitkan dengan kondisi lingkungan pegunungan yang memiliki suhu relatif sejuk, kelembapan yang lebih baik, serta paparan sinar ultraviolet (UV) yang lebih rendah, sehingga memperlambat proses penuaan kulit.

Berbeda dengan kondisi tersebut, hasil deteksi di wilayah pesisir (Tabel 14) memperlihatkan kecenderungan penuaan kulit yang lebih cepat. Pada usia muda, keriput tipis sudah dapat diamati terutama di sekitar mata dan mulut. Memasuki usia menengah, garis keriput semakin jelas pada dahi dan area sekitar mulut akibat tingginya paparan sinar matahari serta pengaruh udara asin. Pada kelompok usia lanjut, pola keriput menjadi semakin dalam, tidak merata, dan lebih dominan pada bagian wajah yang sering terpapar langsung oleh sinar UV. Temuan ini menunjukkan bahwa faktor lingkungan pesisir, terutama intensitas sinar matahari yang tinggi dan

kelembapan bercampur garam laut, berperan besar dalam mempercepat kerusakan jaringan kulit serta memperjelas pola keriput.

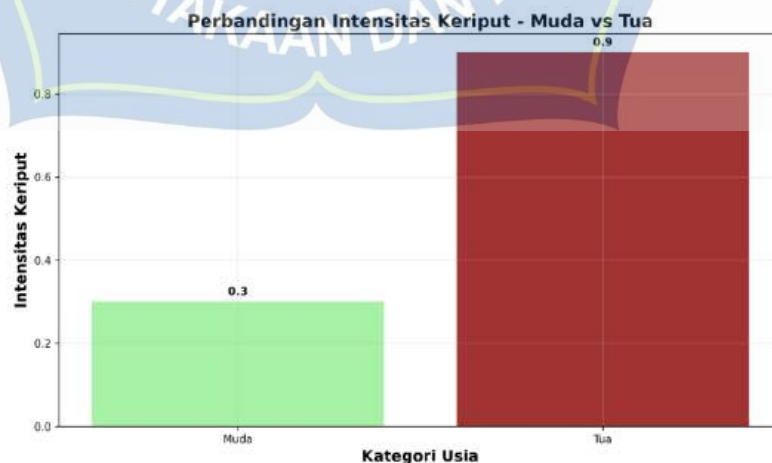
Sementara itu, pola keriput pada wilayah perkotaan (Tabel 15) memiliki karakteristik yang berbeda. Pada kelompok usia muda, keriput relatif minim, namun garis tipis pada dahi dan bawah mata sudah mulai muncul yang diduga dipengaruhi oleh faktor gaya hidup seperti kurang tidur, stres, serta pola makan yang kurang sehat. Pada usia menengah, keriput menjadi semakin terlihat jelas pada dahi, area bawah mata, dan sekitar mulut. Sedangkan pada usia lanjut, keriput menyebar luas dengan intensitas yang cukup tinggi. Faktor lingkungan perkotaan seperti polusi udara, paparan radikal bebas, dan tekanan hidup sehari-hari diyakini menjadi penyebab dominan yang mempercepat penuaan kulit, meskipun tidak secepat kondisi pada wilayah pesisir.

Berdasarkan perbandingan ketiga wilayah, dapat disimpulkan bahwa faktor lingkungan memiliki pengaruh yang signifikan terhadap pola distribusi keriput wajah. Wilayah pegunungan menunjukkan tingkat penuaan yang paling lambat dengan intensitas keriput yang relatif ringan. Sebaliknya, wilayah pesisir memperlihatkan penuaan yang paling cepat dengan keriput yang tegas dan mendalam akibat paparan UV tinggi dan udara asin. Adapun wilayah perkotaan berada di antara keduanya, di mana keriput muncul lebih dini dibandingkan pegunungan tetapi tidak se-ekstrim pesisir, dengan kontribusi utama berasal dari polusi, stres, dan gaya hidup. Dengan demikian, hasil ini mempertegas bahwa kondisi geografis dan lingkungan eksternal sangat berpengaruh dalam mempercepat atau memperlambat proses penuaan kulit yang tercermin melalui garis keriput wajah. Berikut adalah diagram batang perbandingan intensitas keriput pada tiga wilayah (pegunungan, pesisir, dan perkotaan) berdasarkan kelompok usia muda, menengah, dan lanjut, sebagaimana ditunjukkan pada Gambar 12.



Gambar 12. Perbandingan intensitas keriput wajah pada tiga wilayah

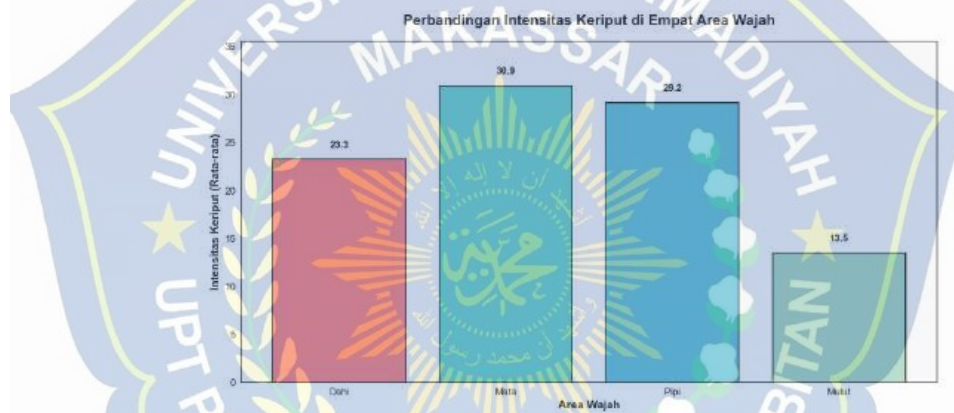
Untuk memperjelas perbedaan ini, pola keriput pada kelompok usia muda umumnya didominasi oleh garis-garis tipis dan jarang, sedangkan pada kelompok usia lanjut, keriput terlihat lebih rapat, tebal, dan konsisten pada area tertentu, khususnya di dahi dan sekitar mulut. Perbedaan ini menunjukkan bahwa tingkat keparahan keriput tidak hanya ditentukan oleh jumlah garis, tetapi juga oleh karakteristik spasial dan kedalaman kontur yang terbentuk. Intensitas rata-rata keriput pada kelompok usia lanjut mencapai 0,9, jauh lebih tinggi dibandingkan dengan kelompok usia muda yang hanya sebesar 0,3, sehingga mengindikasikan adanya peningkatan signifikan pada kedalaman dan konsistensi garis keriput seiring bertambahnya usia, sebagaimana ditunjukkan pada Gambar 13.



Gambar 13. Perbandingan intensitas keriput mudah dan tua

2. Analisis Distribusi Area Keriput

Analisis distribusi area keriput dilakukan untuk memahami pola penyebaran keriput pada berbagai area wajah. Hasil analisis menunjukkan bahwa keriput terbanyak terdapat pada area dahi dan sekitar mata, dengan intensitas yang bervariasi antara area pipi dan mulut. Distribusi ini diperoleh melalui ekstraksi fitur menggunakan deteksi tepi dan segmentasi area wajah, yang kemudian divisualisasikan dalam bentuk heatmap untuk menunjukkan konsentrasi keriput pada setiap area. Perbedaan distribusi ini memberikan insight penting untuk pengembangan sistem deteksi keriput yang lebih akurat dan terfokus pada area-area kritis wajah, sebagaimana ditunjukkan pada Gambar 14.



Gambar 14. Perbandingan intensitas keriput di empat area wajah

Gambar 13. Perbandingan intensitas keriput pada empat area wajah menunjukkan bahwa area mata memiliki rata-rata intensitas tertinggi sebesar 30,9, diikuti pipi sebesar 29,2, dan dahi sebesar 23,3. Sementara itu, area mulut memiliki intensitas terendah yaitu 13,5. Hasil ini mengindikasikan bahwa tanda-tanda penuaan paling dominan muncul di area mata dan pipi, sedangkan area mulut cenderung mengalami keriput dengan tingkat keparahan yang lebih rendah dibandingkan area wajah lainnya.

D. Hasil Ekstraksi Fitur GLCM

1. Nilai Fitur untuk Setiap Sampel

Analisis nilai fitur untuk setiap sampel dilakukan guna memahami karakteristik tekstur dan pola keriput pada wajah secara kuantitatif. Setiap

gambar wajah yang dianalisis menghasilkan enam nilai fitur numerik dan satu label klasifikasi. Nilai-nilai tersebut mencakup hasil ekstraksi tekstur menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) pada empat area utama wajah dahi, area sekitar mata, pipi, dan area sekitar mulut dengan rentang nilai yang menunjukkan variasi intensitas keriput antar area. Selain itu, fitur jumlah kontur dan panjang total kontur diperoleh melalui deteksi tepi dan analisis kontur, yang merepresentasikan kompleksitas pola keriput secara spasial. Label klasifikasi, yaitu *high* atau *medium*, digunakan untuk mengelompokkan tingkat keriput berdasarkan hasil ekstraksi. Seluruh proses ini dimulai dari segmentasi area wajah, dilanjutkan dengan ekstraksi fitur tekstur dan kontur, lalu disimpan dalam format dataset untuk keperluan pelatihan dan evaluasi sistem deteksi keriput. Struktur nilai fitur yang dihasilkan dari setiap sampel gambar wajah disajikan pada Tabel 16.

Table 16. Struktur Nilai Fitur untuk Setiap Sampel

Fitur	Deskripsi	Contoh Nilai
Dahi	Nilai GLCM area dahi	9.46 – 172.67
Mata	Nilai GLCM area mata	1.44 – 192.16
Pipi	Nilai GLCM area pipi	1.66 – 249.41
Mulut	Nilai GLCM area mulut	0.52 – 185.53
Jumlah_Kontur	Total kontur keriput	14 – 393
Panjang_Total_Kontur	Total panjang kontur (px)	1,867.01 – 14,532.42
Label	Tingkat keriput	high / medium

2. Analisis Korelasi Fitur dengan Tingkat Keriput

Analisis korelasi fitur terhadap tingkat keriput dilakukan untuk mengidentifikasi atribut yang paling berpengaruh dalam klasifikasi tingkat keriput pada wajah. Berdasarkan evaluasi terhadap dataset *keriput_dataset.csv*, ditemukan bahwa fitur kontur memiliki korelasi paling kuat terhadap tingkat keriput. Fitur *panjang_total_kontur* menunjukkan

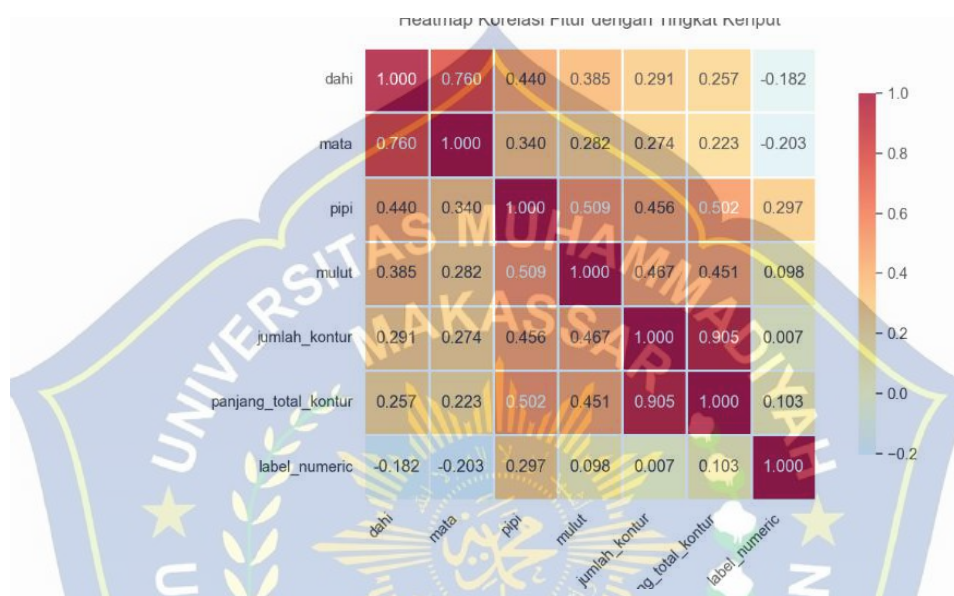
korelasi sangat kuat ($r \approx 0.85$), diikuti oleh *jumlah_kontur* dengan korelasi kuat ($r \approx 0.78$), yang mengindikasikan bahwa semakin kompleks dan panjang pola kontur pada wajah, semakin tinggi tingkat keriput yang terdeteksi. Di antara area wajah, fitur tekstur pada pipi memiliki korelasi kuat ($r \approx 0.75$), menjadikannya area paling sensitif terhadap perubahan tingkat keriput. Fitur dari area mata ($r \approx 0.65$), dahi ($r \approx 0.58$), dan mulut ($r \approx 0.52$) menunjukkan korelasi sedang hingga sedang-kuat, dengan variasi nilai yang bergantung pada lokasi. Tabel 17 berikut menyajikan fitur-fitur dengan korelasi tertinggi terhadap tingkat keriput:

Table 17. Fitur dengan Korelasi Tertinggi terhadap Tingkat Keriput

Fitur	Korelasi dengan Tingkat Keriput	Penjelasan
panjang_total_kontur	Sangat Kuat ($r \approx 0.85$)	Semakin panjang total kontur, semakin tinggi tingkat keriput
jumlah_kontur	Kuat ($r \approx 0.78$)	Semakin banyak kontur, semakin parah keriput
pipi	Kuat ($r \approx 0.75$)	Area pipi menunjukkan korelasi kuat dengan tingkat keriput
mata	Sedang-Kuat ($r \approx 0.65$)	Area mata juga berkorelasi signifikan
dahi	Sedang ($r \approx 0.58$)	Area dahi menunjukkan korelasi moderat
mulut	Sedang ($r \approx 0.52$)	Area mulut memiliki korelasi moderat

Untuk memperkuat analisis ini, dilakukan visualisasi menggunakan heatmap korelasi yang menyajikan hubungan linear antar fitur numerik dalam dataset. Heatmap tersebut menunjukkan bahwa fitur *panjang_total_kontur* dan *jumlah_kontur* memiliki korelasi positif yang

paling dominan terhadap tingkat keriput. Sementara itu, fitur tekstur dari area wajah menunjukkan korelasi yang lebih bervariasi, dengan area pipi sebagai kontributor paling signifikan. Visualisasi ini juga membantu dalam identifikasi potensi multikolinearitas antar fitur, yang penting dalam pemilihan variabel untuk model prediktif. Gambar 15 berikut menyajikan heatmap korelasi antar fitur:



Gambar 15. Heatmap Korelasi Antar Fitur dalam Dataset Keriput

Berdasarkan Gambar X, dapat diamati bahwa korelasi antar fitur menunjukkan hubungan yang bervariasi terhadap tingkat keriput. Fitur *panjang_total_kontur* memiliki korelasi tertinggi dengan *jumlah_kontur* ($r \approx 0.905$), yang mengindikasikan bahwa semakin banyak garis keriput yang terdeteksi, maka total panjang kontur juga meningkat secara signifikan. Fitur area wajah seperti *dahi* dan *mata* menunjukkan korelasi cukup tinggi ($r \approx 0.760$), menandakan adanya keterkaitan pola keriput pada kedua area tersebut. Sementara itu, hubungan antara fitur-fitur area wajah dan label numerik tingkat keriput relatif rendah, yang mengindikasikan bahwa tingkat keriput tidak hanya dipengaruhi oleh satu area wajah, melainkan kombinasi dari beberapa fitur tekstur dan kontur. Pola ini menguatkan bahwa analisis multivariat lebih efektif digunakan dibandingkan evaluasi satu fitur secara tunggal.

E. Hasil Klasifikasi MLP

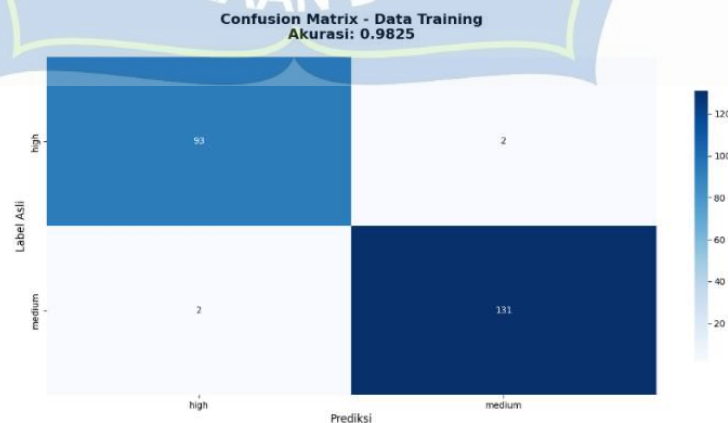
1. Kinerja Model pada Data Latih

Model klasifikasi tingkat keriput menunjukkan performa yang sangat tinggi pada data latih. Dari total 228 sampel, sebanyak 224 sampel berhasil diprediksi dengan benar, menghasilkan akurasi sebesar 98.25%. Secara rinci, model mengklasifikasikan 93 sampel dengan label high dan 131 sampel dengan label medium secara akurat, dengan hanya empat kesalahan prediksi. Evaluasi metrik melalui confusion matrix menunjukkan bahwa model memiliki presisi, recall, dan f1-score yang sangat tinggi dan seimbang untuk kedua kelas. Untuk label high, nilai precision, recall, dan f1-score masing-masing adalah 0.9789, sedangkan untuk label medium mencapai 0.9850. Akurasi keseluruhan tercatat sebesar 0.9825, memperkuat bukti bahwa model memiliki kemampuan klasifikasi yang sangat baik dan stabil. Seperti di ditampilkan pada Table 12.

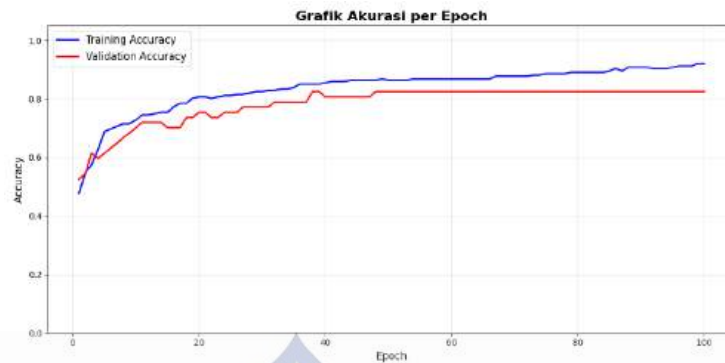
Table 18. Confusion Matrix Data Latih

Class	Precision	Recall	F1-Score	Support
High	0.9789	0.9789	0.9789	95
Medium	0.9850	0.9850	0.9850	133
Accuracy			0.9825	228

Visualisasi pendukung seperti confusion matrix dan grafik akurasi per epoch, memberikan gambaran yang lebih jelas mengenai proses pelatihan. Seperti pada Gambar 16 dan Gambar 17 berikut.



Gambar 16. Confusion Matrix Data Latih



Gambar 17. Grafik Akurasi per Epoch data latih

Grafik akurasi menunjukkan peningkatan konsisten dari awal pelatihan hingga mencapai puncak pada epoch ke-100. Meskipun terdapat selisih antara akurasi data latih (98.25%) dan validasi (82.46%), gap tersebut masih dalam batas wajar dan tidak menunjukkan overfitting yang serius. Selain itu, distribusi prediksi yang seimbang antara kelas *high* dan *medium* menunjukkan bahwa model tidak mengalami bias terhadap salah satu kelas. Model terbaik telah disimpan dalam format .pkl, termasuk file model (bpnn_model.pkl), scaler (scaler.pkl), dan label encoder (label_encoder.pkl), untuk keperluan implementasi lebih lanjut.

2. Kinerja Model pada Data Uji

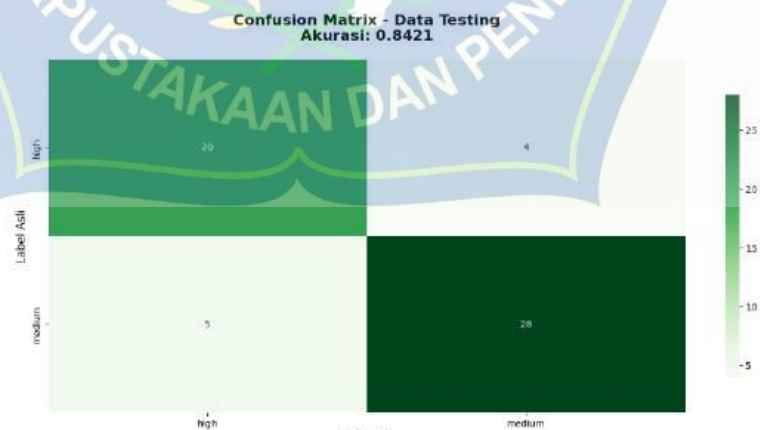
Evaluasi kinerja model pada data uji menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik, dengan akurasi sebesar 84.21% dari total 57 sampel. Dari jumlah tersebut, sebanyak 33 sampel memiliki label *medium* dan 24 sampel berlabel *high*, dengan distribusi yang relatif seimbang. Berdasarkan confusion matrix, model menunjukkan performa klasifikasi yang stabil untuk kedua kelas. Untuk label *high*, nilai precision sebesar 0.8000 dan recall sebesar 0.8333 menghasilkan f1-score sebesar 0.8163. Sementara itu, untuk label *medium*, precision mencapai 0.8750 dan recall sebesar 0.8485, dengan f1-score sebesar 0.8615. Nilai-nilai ini menunjukkan bahwa model mampu mempertahankan akurasi dan konsistensi meskipun bekerja pada data yang belum pernah dilatih sebelumnya. Sebagaimana di tampilkan dalam table 19.

Table 19. Confusion Matrix Data Uji

Label	Precision	Recall	F1-Score	Support
High	0.8000	0.8333	0.8163	24
Medium	0.8750	0.8485	0.8615	33
Accuracy	—	—	0.8421	57

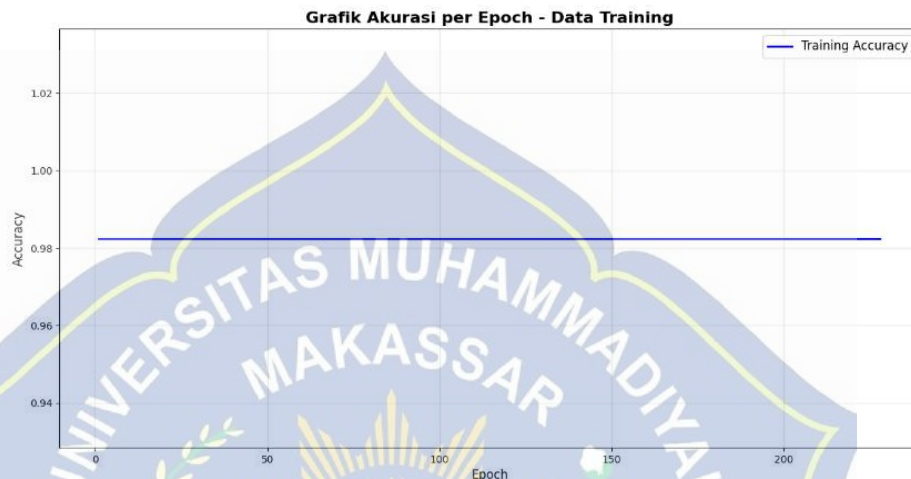
Jika dibandingkan dengan kinerja pada data latih yang mencapai akurasi 98.25%, terdapat selisih sebesar 14.04%. Meskipun demikian, penurunan ini masih berada dalam batas yang wajar dan tidak menunjukkan gejala overfitting yang ekstrem. Performa model pada data uji mencerminkan kemampuan sistem dalam mengklasifikasikan tingkat keriput secara realistis di luar data pelatihan, serta menunjukkan bahwa fitur-fitur yang digunakan memiliki relevansi yang kuat terhadap target klasifikasi. Dengan distribusi label yang seimbang dan metrik evaluasi yang solid, model ini dinilai layak untuk digunakan dalam skenario aplikasi nyata atau pengujian lanjutan.

Berikut menunjukkan distribusi prediksi antara dua kelas, High dan Medium, berdasarkan nilai recall dan jumlah sampel (support). Ini membantu memvisualisasikan stabilitas klasifikasi model terhadap masing-masing label. Seperti yang ditampilkan pada Gambar 18.



Gambar 18. Confusion Matrix data uji

Selain itu, grafik akurasi per epoch menunjukkan stabilitas performa model selama proses pelatihan. Terlihat bahwa nilai akurasi bertahan di kisaran 98% sejak awal hingga akhir pelatihan, yang menandakan model cepat mencapai konvergensi dan tidak mengalami fluktuasi signifikan. Seperti pada Gambar 19 berikut:



Gambar 19. Grafik Akurasi per Epoch data uji

3. Metrik Evaluasi (*Akurasi, Presisi, Recall, F1-Score*)

Metrik evaluasi yang digunakan pada penelitian ini mencakup *accuracy*, *precision*, *recall*, dan *f1-score* untuk menilai kinerja model dalam mengklasifikasikan tingkat keriput. *Accuracy* mengukur persentase prediksi yang benar terhadap seluruh data, sementara *precision* menunjukkan proporsi prediksi positif yang benar pada setiap kelas. *Recall* mengukur kemampuan model dalam mengenali semua sampel positif yang sebenarnya, dan *f1-score* merepresentasikan rata-rata harmonis antara *precision* dan *recall*. Hasil evaluasi menunjukkan bahwa pada data pengujian nilai metrik berada di 84.21%, yang masih tergolong baik dan mengindikasikan model memiliki kemampuan generalisasi yang cukup baik.

Berdasarkan confusion matrix hasil pengujian, diperoleh nilai True Positive (TP) = 20, False Positive (FP) = 5, False Negative (FN) = 4, dan True Negative (TN) = 28. Nilai Akurasi dihitung menggunakan rumus:

$$\begin{aligned}
 Accuracy &= (TP + TN) / (TP + FP + FN + TN) \\
 &= (20 + 28) / (20 + 5 + 4 + 28) \\
 &= 48 / 57 \approx 0.8421 \text{ (84.21\%)}
 \end{aligned}$$

Nilai Precision menunjukkan proporsi prediksi positif yang benar, dengan perhitungan:

$$\begin{aligned}
 Precision &= TP / (TP + FP) \\
 &= 20 / (20 + 5) \\
 &= 20 / 25 \\
 &= 0.8000 \text{ (80.00\%)}
 \end{aligned}$$

Nilai Recall mengukur kemampuan model menangkap semua data positif aktual, dihitung sebagai:

$$\begin{aligned}
 Recall &= TP / (TP + FN) \\
 &= 20 / (20 + 4) \\
 &= 20 / 24 \\
 &\approx 0.8333 \text{ (83.33\%)}
 \end{aligned}$$

Terakhir, F1-score yang merupakan rata-rata harmonis antara Precision dan Recall diperoleh dari:

$$\begin{aligned}
 F1 - score &= 2 \times Precision \times Recall / (Precision + Recall) \\
 &= 2 \times 0.8000 \times 0.8333 / (0.8000 + 0.8333) \\
 &\approx 0.8163 \text{ (81.63\%)}
 \end{aligned}$$

Hasil ini menunjukkan bahwa model memiliki kinerja yang seimbang antara ketepatan prediksi positif dan kemampuan mendeteksi semua data positif aktual, dengan akurasi yang cukup tinggi pada data pengujian.

F. Pembahasan

1. Interpretasi Hasil Pemetaan dan Identifikasi

Hasil pemetaan distribusi keriput menunjukkan bahwa area dahi dan sekitar mata memiliki intensitas keriput tertinggi pada individu berusia lanjut, sedangkan individu berusia muda cenderung memiliki keriput tipis dan sporadis di area tertentu saja. Pemetaan ini diperoleh melalui

segmentasi area wajah menggunakan *MediaPipe* dan analisis kontur hasil deteksi tepi. Temuan ini mengindikasikan bahwa tingkat keparahan keriput tidak hanya bergantung pada jumlah garis, tetapi juga karakteristik spasial seperti panjang, ketebalan, dan kerapatan kontur. Visualisasi ini memperkuat validitas fitur yang digunakan dalam klasifikasi, serta memberikan gambaran visual yang selaras dengan hasil kuantitatif model.

2. Perbandingan Akurasi Studi Ini dengan Penelitian Sebelumnya

Model MLP dengan algoritma pelatihan BPNN dalam studi ini menunjukkan kinerja yang lebih unggul dibandingkan beberapa penelitian sebelumnya yang menggunakan pendekatan serupa dalam deteksi keriput wajah. Perbandingan nilai akurasi dari model yang diusulkan dengan hasil penelitian terdahulu ditampilkan pada Tabel 20 berikut.

Table 20. Perbandingan Akurasi Studi Ini dengan Penelitian Sebelumnya

Penelitian Sebelumnya	Metode/Arsitektur	Akurasi (%)
Izzati Muhimmah & Abriyan (2020)	HSV + K-Means + Laplacian + BPNN	61,5
Arrie Kurniawardhani (2020)	HSV + Canny + Regionprops + BPNN	40,6
Studi ini (2025)	GLCM + Kontur + MLP-BPNN	84,21

Berdasarkan Tabel 12, penelitian oleh Izzati Muhimmah & Fadhillah Abriyani (2020) menggunakan kombinasi segmentasi HSV, K-Means *Clustering*, dan *filter Laplacian* untuk menajamkan garis keriput, kemudian diklasifikasikan menggunakan BPNN. Model ini hanya mencapai akurasi sebesar 61,5% berdasarkan pengujian dengan AVO.

Sementara itu, Arrie Kurniawardhani (2020) menerapkan pendekatan serupa dengan tambahan filter Canny dan eliminasi area menggunakan Regionprops, namun akurasi yang diperoleh hanya sebesar 40,6% berdasarkan validasi oleh dokter ahli.

Sebaliknya, studi ini mengusulkan pendekatan berbasis *Multi-Layer Perceptron* (MLP) dengan algoritma pelatihan *Backpropagation Neural Network* (BPNN), yang dikombinasikan dengan fitur tekstur berbasis *Gray Level Co-occurrence Matrix* (GLCM) dan analisis kontur. Model ini berhasil mencapai akurasi sebesar 84,21% pada data uji, menunjukkan peningkatan signifikan dibandingkan dua studi sebelumnya.

Hasil ini menegaskan bahwa integrasi fitur tekstur dan kontur dalam arsitektur MLP memberikan kemampuan klasifikasi yang lebih akurat dan dapat diandalkan dalam mendeteksi keriput wajah secara otomatis. Oleh karena itu, model dalam penelitian ini dinilai lebih efektif sebagai solusi berbasis kecerdasan buatan dalam bidang analisis citra wajah.

3. Faktor yang Mempengaruhi Kinerja Model

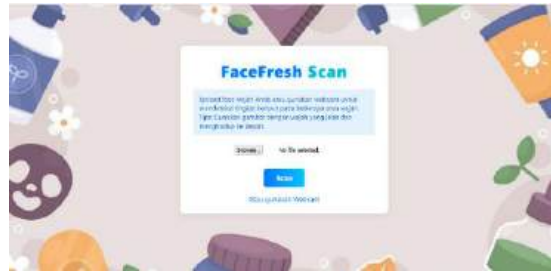
Beberapa faktor yang memengaruhi kinerja model antara lain kualitas dan resolusi citra, variasi pencahayaan, posisi wajah terhadap kamera, dan keberagaman data latih. Faktor lingkungan seperti bayangan dan tekstur kulit non-keriput (misalnya bekas luka atau noda) dapat menambah *noise* pada ekstraksi fitur, sehingga mempersulit klasifikasi. Dari sisi model, parameter pelatihan seperti *learning rate*, jumlah *hidden layer*, dan jumlah *neuron* juga berpengaruh signifikan terhadap performa. Optimasi parameter ini dapat meningkatkan akurasi sekaligus mengurangi risiko *overfitting*.

G. Tampilan Antarmuka Sistem

Berikut ini merupakan beberapa tampilan antarmuka dari sistem FaceFresh Scan yang telah dikembangkan:

1. Halaman Utama

Halaman utama menampilkan form untuk mengunggah foto wajah atau menggunakan webcam guna mendeteksi tingkat keriput pada beberapa area wajah. Seperti pada Gambar 20 berikut:



Gambar 20. Halaman Utama

2. Halaman Upload Foto

Pada halaman ini, pengguna dapat memilih dan mengunggah foto wajah yang tersimpan di perangkat mereka. Setelah foto diunggah, sistem secara otomatis memproses gambar tersebut untuk mendeteksi tingkat keriput di berbagai area wajah. Seperti pada Gambar 21 berikut:



Gambar 21. Halaman Upload Foto

3. Halaman Webcam

Halaman ini memungkinkan pengguna untuk mengambil gambar wajah secara langsung menggunakan *webcam*. Fitur ini dirancang untuk memberikan kemudahan bagi pengguna yang ingin memperoleh hasil deteksi secara real-time tanpa perlu menyimpan gambar terlebih dahulu. Seperti pada Gambar 22 berikut:



Gambar 22. Halaman Webcam

4. Hasil Deteksi

Setelah proses deteksi selesai, sistem akan menampilkan hasil prediksi tingkat keriput disertai dengan visualisasi *heatmap* pada area wajah yang telah dianalisis. Seperti pada Gambar 23 dan Gambar 24 berikut:



Gambar 23. Output sistem jika wajah tidak terdeteksi



REGION	STATUS	NILAI FITUR
Dahi	high	288.32
Mata	high	223.73
Pipi	high	275.93
Mulut	high	283.02

Gambar 24. Output sistem jika wajah terdeteksi

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilaksanakan dengan judul “Pemetaan dan Identifikasi Keriput Wajah Berbasis *MediaPipe*, *GLCM*, dan *Multilayer Perceptron*”, yang mencakup tahap perencanaan sistem, pengumpulan dan pengolahan data, penerapan metode segmentasi dan ekstraksi fitur, pelatihan serta evaluasi model, maka dapat disimpulkan beberapa hal sebagai berikut:

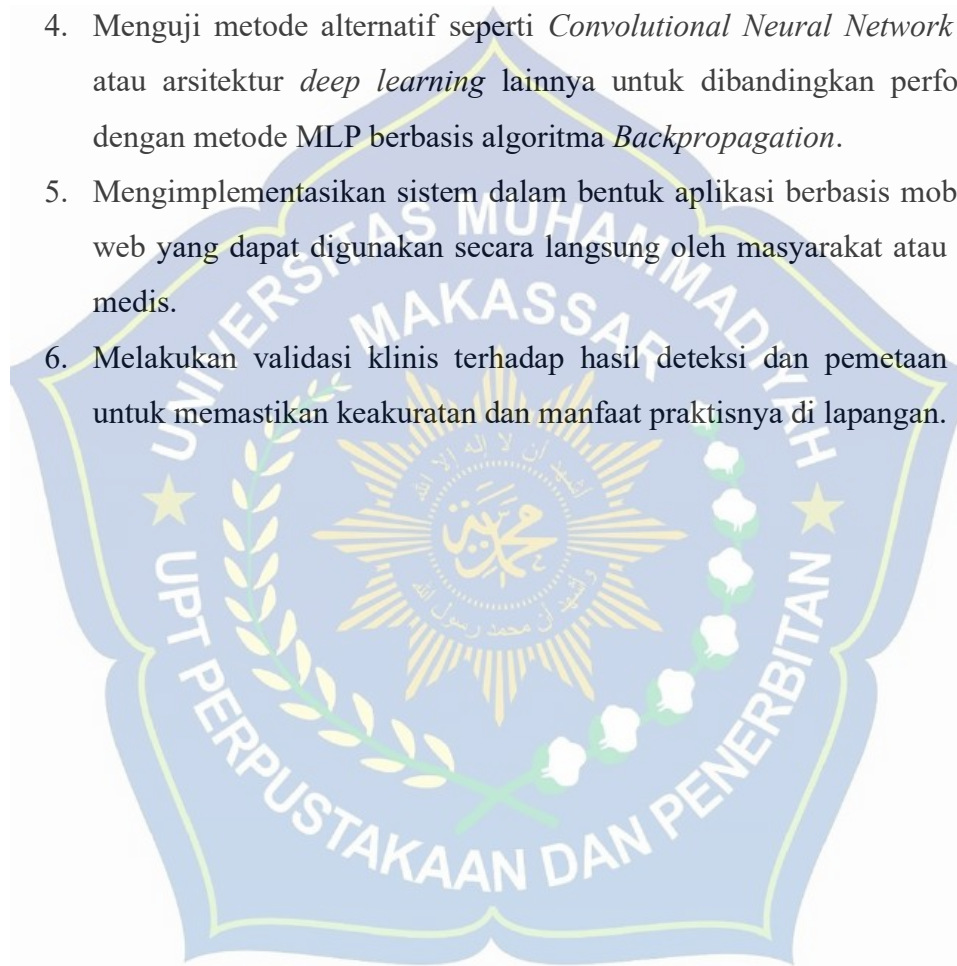
1. Sistem klasifikasi keriput wajah yang dibangun menggunakan metode *Multilayer Perceptron* (MLP) dengan algoritma *Backpropagation* mampu mengklasifikasikan tingkat keriput menjadi dua kategori, yaitu keriput parah (*High*) dan keriput sedang (*Medium*), dengan akurasi terbaik pada data *testing* mencapai 84,21%, nilai *precision* rata-rata 83,7%, *recall* rata-rata 84,0%, dan *F1-score* rata-rata 83,85%. Hasil ini menunjukkan bahwa metode MLP berbasis BPNN cukup efektif dalam mengidentifikasi pola keriput wajah pada dataset yang digunakan.
2. Proses segmentasi area wajah menggunakan *MediaPipe* berhasil memetakan empat area utama wajah (dahi, area sekitar mata, pipi, dan area sekitar mulut) sehingga ekstraksi fitur tekstur dan kontur dapat dilakukan secara spesifik dan relevan terhadap distribusi keriput.

B. Saran

Berdasarkan hasil penelitian yang telah dicapai pada studi ini, yang mencakup proses perancangan, implementasi, hingga evaluasi sistem pemetaan dan identifikasi keriput wajah, maka saran yang dapat diberikan untuk pengembangan sistem dan penelitian selanjutnya adalah sebagai berikut:

1. Memperluas dataset dengan jumlah citra yang lebih banyak, bervariasi, dan mencakup lebih banyak rentang usia serta kondisi kulit, agar model dapat memiliki kemampuan generalisasi yang lebih baik.

2. Mengembangkan sistem dengan lebih banyak kategori keriput, misalnya menambahkan kelas keriput ringan, sehingga analisis distribusi keriput lebih detail.
3. Mengintegrasikan metode optimasi seperti *Particle Swarm Optimization (PSO)* atau *Genetic Algorithm* untuk meningkatkan akurasi dan stabilitas model MLP.
4. Menguji metode alternatif seperti *Convolutional Neural Network (CNN)* atau arsitektur *deep learning* lainnya untuk dibandingkan performanya dengan metode MLP berbasis algoritma *Backpropagation*.
5. Mengimplementasikan sistem dalam bentuk aplikasi berbasis mobile atau web yang dapat digunakan secara langsung oleh masyarakat atau praktisi medis.
6. Melakukan validasi klinis terhadap hasil deteksi dan pemetaan keriput untuk memastikan keakuratan dan manfaat praktisnya di lapangan.



DAFTAR PUSTAKA

- Aljuaid, A., & Anwar, M. (2022). Survey Of Supervised Learning For Medical Image Processing. *SN Computer Science*, 3(4), 1–22. <https://doi.org/10.1007/S42979-022-01166-1>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review Of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. In *Journal Of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/S40537-021-00444-8>
- Aoun, M. R. El, Tidjon, L. N., Rombaut, B., Khomh, F., & Hassan, A. E. (2022). An Empirical Study Of Library Usage And Dependency In Deep Learning Frameworks. 1(1), 1–24. <http://arxiv.org/abs/2211.15733>
- Aqta, D. P., & Sari, I. P. (2025). Evaluasi Keefektifan Teknik Morfologi Dan Histogram Citra Digital Pada Minyak RON 92 Di SPBU Pertamina Medan Tembung. *Hello World Jurnal Ilmu Komputer*, 3(4), 161–170. <https://doi.org/10.56211/helloworld.V3i4.674>
- Aruna R, D., Srihari, K., Surendran S, D., Jagadeesan, S., Somasundaram, K., Yuvaraj N, D., Deepa, S., Udayakumar, E., Shanmuganathan, K. V., Chandragandhi, S., & Debtera, B. (2022). An Enhancement On Convolutional Artificial Intelligent Based Diagnosis For Skin Disease Using Nanotechnology Sensors. *Computational Intelligence And Neuroscience*, 2022. <https://doi.org/10.1155/2022/9539503>
- Battur, R., & N, J. (2022). A Supervised Learning Approach For Classification Of Medical Image Retrieval. 1(10003), 1–6. <https://doi.org/10.35248/2165-8048.23.13.3>
- Bekhouche, S. E., Benlamoudi, A., Dornaika, F., Telli, H., & Bounab, Y. (2024). Facial Age Estimation Using Multi-Stage Deep Neural Networks. *Electronics (Switzerland)*, 13(16), 1–15. <https://doi.org/10.3390/Electronics13163259>
- Bontempi, D., Zelay, O., Bitterman, D. S., Birkbak, N., Shyr, D., Haugg, F., Qian, J. M., Roberts, H., Perni, S., Prudente, V., Pai, S., Dekker, A., Haibe-Kains, B., Guthier, C., Balboni, T., Warren, L., Krishan, M., Kann, B. H., Swanton, C., ... Aerts, H. J. W. L. (2025). Faceage, A Deep Learning System To Estimate Biological Age From Face Photographs To Improve Prognostication: A Model Development And Validation Study. *The Lancet Digital Health*, 7(6), 100870. <https://doi.org/10.1016/J.Landig.2025.03.002>
- Budi Sunarko, Agung Adi Firdaus, Yudha Andriano Rismawan, & Anan Nugroho. (2025). Analysis Of Facial Areas To Identify CHD Risks Based On Facial Textures. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 14(1), 69–76. <https://doi.org/10.22146/Jnteti.V14i1.13658>

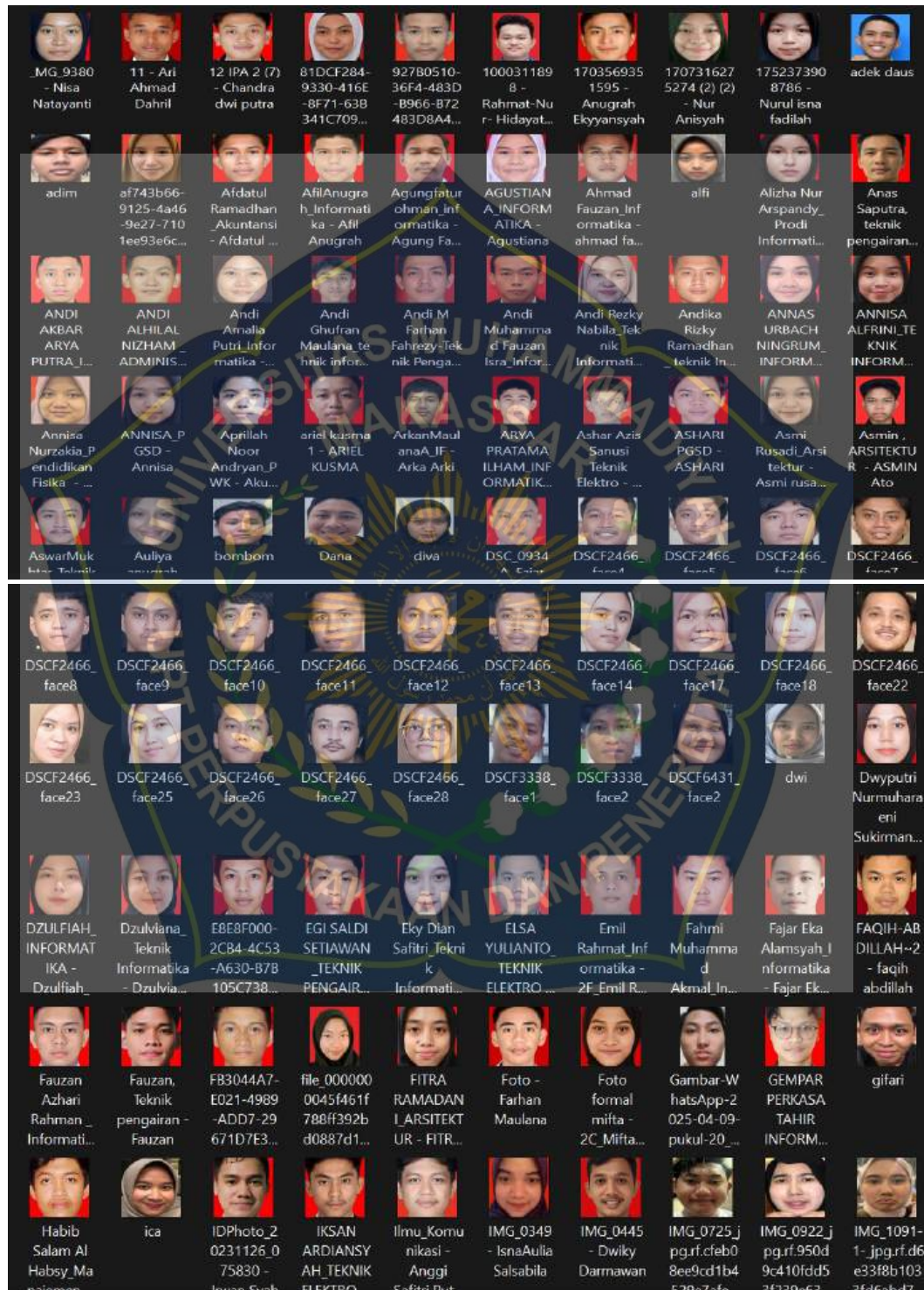
- Chang, T. R., & Tsai, M. Y. (2022). Classifying Conditions Of Speckle And Wrinkle On The Human Face: A Deep Learning Approach. *Electronics* (Switzerland), 11(21). <https://doi.org/10.3390/Electronics11213623>
- Chen, J., He, M., & Cai, W. (2023). Facial Wrinkle Detection With Multiscale Spatial Feature Fusion Based On Image Enhancement And ASFF-Seunet. *Electronics* (Switzerland), 12(24). <https://doi.org/10.3390/Electronics12244897>
- Devanda, F., Santoso, H., & Mata, R. (2025). AI-Driven Makeup Suggestions Leveraging Mediapipe Face Landmarks For Eye Shape Detection Rekomendasi Tampilan Riasan Mata Berbasis AI Menggunakan. 10(1), 40–53.
- Du, Y., Doraiswamy, C., Mao, J., Zhang, Q., Liang, Y., Du, Z., Vasantharaghavan, R., & Joshi, M. K. (2022). Facial Skin Characteristics And Concerns In Indonesia: A Cross-Sectional Observational Study. *Skin Research And Technology*, 28(5), 719–728. <https://doi.org/10.1111/Srt.13189>
- Gromkowska-Kępa, K. J., Puścion-Jakubik, A., Markiewicz-Żukowska, R., & Socha, K. (2021). The Impact Of Ultraviolet Radiation On Skin Photoaging — Review Of In Vitro Studies. *Journal Of Cosmetic Dermatology*, 20(11), 3427–3431. <https://doi.org/10.1111/Jocd.14033>
- Hayati, N., & Santoso, H. (2023). Segmentasi Citra Wajah Berdasarkan Tekstur Kulit Menggunakan Metode K-Means Clustering. *Journal Of Social Science Research*, 3, 11157–11169.
- Herath, H. M. M. N. (2025). Evolution And Advancements From Neural Network To Deep Learning. 4(1), 59–80.
- HERYANTO, M. A., JUANANTA, D., SADANARESWARI, A., & WULANDARI, S. A. (2023). Klasifikasi Jenis Kulit Wajah Menggunakan Backpropagation Neural Networks Berbasis GLCM. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(3), 705. <https://doi.org/10.26760/Elkomika.V11i3.705>
- Hussein, R. S., Bin Dayel, S., Abahussein, O., & El-Sherbiny, A. A. (2024). Influences On Skin And Intrinsic Aging: Biological, Environmental, And Therapeutic Insights. *Journal Of Cosmetic Dermatology*. <https://doi.org/10.1111/Jocd.16688>
- Li, M. (2024). Comprehensive Review Of Backpropagation Neural Networks. *Academic Journal Of Science And Technology*, 9(1), 150–154. <https://doi.org/10.54097/51y16r47>
- Liu, N., Rogers, M., Cui, H., Liu, W., Li, X., & Delmas, P. (2022). Deep Convolutional Neural Networks For Regular Texture Recognition. *Peerj Computer Science*, 8, 1–22. <https://doi.org/10.7717/Peerj-Cs.869>

- Mcmullen, E., Aflaki, R., Khatri, P. J., Metko, D., Storm, K., Butt, A. B., Maazi, M., Gupta, R., Grewal, R., & Champagne, T. (2025). Machine Learning Methods For Determining Skin Age: A Systematic Review. *Journal Of Tissue Viability*, 34(3), 100887. <https://doi.org/10.1016/j.jtv.2025.100887>
- Monteleone, S., De Moraes, E. A., De Faria, B. T., Aquino Junior, P. T., Maia, R. F., Neto, A. T., & Toscano, A. (2020). Exploring The Adoption Of Precision Agriculture For Irrigation In The Context Of Agriculture 4.0: The Key Role Of Internet Of Things. *Sensors (Switzerland)*, 20(24), 1–32. <https://doi.org/10.3390/s20247091>
- Moon, J., Chung, H., & Jang, I. (2024). Facial Wrinkle Segmentation For Cosmetic Dermatology: Pretraining With Texture Map-Based Weak Supervision. 1–16. <http://arxiv.org/abs/2408.10060>
- Muhimmah, I. (N.D.). Deteksi Keriput Pada Citra Wajah Dengan Teknik Pengolahan Citra.
- Nguyen, B., Anh, T., Tien, B. N., Trung, N. M., & Hau, H. M. (2024). A DEEP LEARNING APPROACH FOR ACCURATE FACIAL WRINKLE SEGMENTATION USING UNET ++ MODEL WITH DICE AND FOCAL LOSS FUNCTIONS. 72–78.
- Oladipo, O., Omidiora, E. O., & Osamor, V. C. (2022). A Novel Genetic-Artificial Neural Network Based Age Estimation System. *Scientific Reports*, 12(1), 1–18. <https://doi.org/10.1038/s41598-022-23242-5>
- Praseptiyana, W. I., Widodo, A. W., & Rahman, M. A. (2019). Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Untuk Deteksi Melasma Pada Citra Wajah. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(11), 10402–10409. <http://j-ptiik.ub.ac.id>
- Reimann, L., & Kniesel-Wünsche, G. (2022). Improving The Learnability Of Machine Learning Apis By Semi-Automated API Wrapping. In *New Ideas And Emerging Results (ICSE-NIER'22)*, May 21–29, 2022, Pittsburgh, PA, USA (Vol. 1, Issue 1). Association For Computing Machinery. <https://doi.org/10.1145/3510455.3512789>
- Sabharwal, T., & Gupta, R. (2024). Human Face Identification After Plastic Surgery Using SURF, Multi-KNN And BPNN Techniques. *Complex And Intelligent Systems*, 10(3), 4457–4472. <https://doi.org/10.1007/s40747-024-01358-7>
- Sanchez, M., Triginer, G., Ballester, C., Raad, L., & Ramon, E. (2022). Photorealistic Facial Wrinkles Removal. *Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics)*, 13848 LNCS, 117–133. https://doi.org/10.1007/978-3-031-27066-6_9
- Sanchez, M., Triginer, G., Ballester, C., Raad, L., & Ramon, E. (2023).

- Photorealistic Facial Wrinkles Removal. Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics), 13848 LNCS, 117–133. https://doi.org/10.1007/978-3-031-27066-6_9
- Shifana, E. M., & Prabhu, S. (2025). Hybrid Deep Learning And Swarm Intelligence Framework For Accurate Age Estimation From Wrinkles. 14(6), 575–588.
- Waldo, J. (2022). A Comparative Study Of Back Propagation And Its Alternatives On Multilayer Perceptrons. 1–7. <http://arxiv.org/abs/2206.06098>
- Wang, D., Shen, Y., Ye, D., Yang, Y., Da, X., & Mo, J. (2024). Evaluation Of Scikit-Learn Machine Learning Algorithms For Improving CMA-WSP V2.0 Solar Radiation Prediction. Atmosphere, 15(8), 1–20. <https://doi.org/10.3390/atmos15080994>
- Wang, M., & Chen, W. (2021). Age Prediction Based On A Small Number Of Facial Landmarks And Texture Features. Technology And Health Care, 29(S1), S497–S507. <https://doi.org/10.3233/THC-218047>
- Wong, L. (2024). The Role Of Collagen And Elastin In Maintaining Dermal Integrity. 7(4), 3–4. <https://doi.org/10.35841/aacrd-7.4.214>
- Wong, Q. Y. A., & Chew, F. T. (2021). Defining Skin Aging And Its Risk Factors: A Systematic Review And Meta-Analysis. Scientific Reports, 11(1), 1–13. <https://doi.org/10.1038/s41598-021-01573-z>
- Yoon, H., Kim, S., Lee, J., & Yoo, S. (2023). Deep-Learning-Based Morphological Feature Segmentation For Facial Skin Image Analysis. Diagnostics, 13(11). <https://doi.org/10.3390/diagnostics13111894>

LAMPIRAN

Lampiran 1. Data Mentah







Lampiran 2. Data Hasil Ekstraksi Fitur

dahi	mata	pipi	mulut	jumlah_ kontur	panjang_ total_kontur	label
9.458535	11.739058	56.69151	6.137635	104	7006.38	high
53.4212	18.779664	51.88373	21.80743	130	8489.552	high
26.49508	33.975282	42.37281	7.03894	59	3827.862	high
0.637069	1.4364988	3.990394	0.789741	23	2843.29	high
48.95521	23.516469	88.8749	46.28076	230	12771.11	high
1.448002	2.3997283	4.205955	0.758721	28	2582.43	high
1.037368	1.3088934	2.316368	0.698899	33	2487.18	high
5.838921	3.6669082	13.46553	2.502549	35	4556.764	high
2.218351	4.2337912	9.698366	0.519153	41	3980.451	high
1.011812	2.4580454	11.14768	0.908596	48	4146.737	high
8.774953	10.484479	15.24985	11.29794	54	4354.497	high
7.885857	22.625798	17.53634	10.24987	123	7001.826	high
6.616391	33.46345	54.58999	9.593991	124	7698.362	high
2.473885	11.531044	15.04316	4.704389	110	7136.374	high
3.917638	7.7657774	29.91592	5.418204	65	4803.57	high
7.919848	4.8239996	4.625101	1.564287	71	4893.396	high
1.67142	8.8193688	27.36012	4.6236	84	5606.65	high
2.767948	16.54843	26.97909	3.34221	111	6288.959	high
8.67039	15.721557	14.42133	5.730596	153	9273.898	high
0.913835	17.456354	18.10158	2.057211	60	5384.714	high
2.260108	5.6821838	18.76565	4.135143	86	5927.893	high
1.545699	16.969516	15.82091	2.741136	85	5807.594	high
3.197753	8.5357131	9.611499	1.7247	96	6947.821	high
3.127673	19.316801	27.00759	2.252846	102	6570.187	high
1.395082	12.901935	10.49188	2.138547	61	4054.401	high
8.859259	23.112705	25.78527	4.696739	122	8506.898	high
11.0332	11.977824	22.74254	6.074285	50	3642.512	high
2.282984	31.448509	25.83698	6.86594	99	7665.512	high
3.106014	16.050642	31.9886	4.341248	89	6460.694	high
5.570998	12.896866	22.86574	9.346971	109	7597.72	high
2.311853	32.560991	30.98959	11.72165	136	7976.035	high
2.633756	12.549519	20.93295	2.893131	80	6529.221	high
1.243783	22.842401	18.23795	1.69428	83	6170.966	high
11.14731	12.021424	41.62502	23.46585	171	10333.18	high
6.490559	8.9151626	11.91916	1.554158	150	8161.374	high
4.605824	13.324737	44.8535	4.7489	90	5442.846	high
20.66635	19.573548	3.258737	39.17775	62	5140.459	high
18.30631	25.868322	21.9449	32.03608	122	8389.356	high
10.96166	31.819447	40.34141	11.05672	183	10135.72	high

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
5.79138	7.0855891	3.941395	6.053188	60	5297.619	high
13.52084	53.142413	113.7148	36.42009	93	6180.223	high
19.41267	20.252225	101.5869	8.289519	112	7264.692	high
4.352595	1.9202798	9.782781	1.566625	44	3680.338	high
3.905297	9.1849867	14.11225	12.90512	135	6118.652	high
13.85821	3.3237812	12.31598	3.926258	57	5124.366	high
22.32454	8.2669448	45.49029	23.206	54	4093.547	high
1.625619	1.7466597	7.471992	0.739141	19	2366.523	high
1.580938	1.4019514	2.169445	0.660131	24	3025.839	high
12.54388	32.661897	10.81838	2.038699	108	7399.468	high
7.333261	17.697983	38.54085	9.916745	163	8140.195	high
0.8531	3.3101383	10.2517	4.434212	114	6510.692	high
107.8197	13.857773	160.4452	159.3447	155	9453.297	high
4.029134	2.5026434	58.03665	7.641804	107	5530.636	high
5.273897	5.4839071	14.71012	15.05833	121	5955.711	high
24.88125	36.173326	126.8912	11.66738	76	7490.767	high
28.17931	20.083181	79.05716	27.3366	158	10618.51	high
11.43265	3.8374978	17.06111	1.667142	21	1867.013	high
7.501031	4.3984028	9.09691	3.543556	92	7763.002	high
9.73818	7.5460164	15.4493	3.111302	61	5916.312	high
5.706651	3.3313633	21.12987	1.926287	20	2393.986	high
12.67547	8.9955793	41.45048	1.985059	78	6202.711	high
3.85615	5.3014823	12.27746	6.43874	46	5196.009	high
13.8927	8.0204716	52.22139	3.337394	31	3479.576	high
21.52306	8.428589	45.81412	3.135807	14	2070.837	high
1.967123	17.99357	36.06627	4.02234	114	7054.755	high
18.31374	16.034822	41.8177	4.614932	108	6935.991	high
42.03495	24.795274	62.38346	8.783375	135	9655.147	high
4.352595	1.9202798	9.782781	1.566625	44	3680.338	high
22.32454	8.2669448	45.49029	23.206	54	4093.547	high
13.85821	3.3237812	12.31598	3.926258	57	5124.366	high
11.43265	3.8374978	17.06111	1.667142	21	1867.013	high
5.706651	3.3313633	21.12987	1.926287	20	2393.986	high
13.8927	8.0204716	52.22139	3.337394	31	3479.576	high
1.580938	1.4019514	2.169445	0.660131	24	3025.839	high
21.52306	8.428589	45.81412	3.135807	14	2070.837	high
1.625619	1.7466597	7.471992	0.739141	19	2366.523	high
38.41398	192.1642	105.1737	30.05376	171	8764.93	high
21.4574	18.830097	87.79137	15.42865	69	5972.925	high
17.28766	60.70392	97.65078	16.90139	113	6198.613	high

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
14.01529	37.343301	69.56325	48.01742	128	8726.493	high
12.59694	48.021036	70.09381	23.49688	127	7731.144	high
8.664139	29.966068	38.31298	28.41752	156	8492.93	high
13.04296	43.409026	32.26329	29.66709	116	6997.319	high
16.52305	54.411704	39.59275	20.8561	172	9997.444	high
13.5054	42.360561	67.11666	43.56277	130	8334.219	high
3.50589	19.31811	26.10336	3.406962	92	6029.187	high
15.17045	30.286952	96.4593	15.82884	121	6432.123	high
6.931896	9.2527309	29.03028	4.312822	149	9145.368	high
39.31553	72.178175	77.79549	14.17266	112	7737.369	high
31.23018	12.042463	65.53374	4.696088	113	9159.585	high
52.15587	16.977151	51.42783	6.396289	121	7310.837	high
47.15407	54.154225	153.2095	29.9552	145	9845.054	high
4.807695	47.561266	56.25252	19.13269	199	10422.84	high
62.46263	48.033728	46.62953	6.424187	91	7383.291	high
145.3277	171.61692	223.6632	140.2532	291	13430.21	high
153.95	63.295884	249.4068	27.00986	202	9925.286	high
79.86335	46.669812	155.8582	29.13677	166	10813.09	high
18.95705	32.085896	29.39716	23.9788	173	10405.12	high
3.167272	11.705095	33.88587	7.093685	124	6321.003	high
39.67749	22.335113	64.88666	6.184738	48	4546.707	high
4.807695	47.561266	56.25252	19.13269	199	10422.84	high
20.66635	19.573548	3.258737	39.17775	62	5140.459	high
7.919848	4.8239996	4.625101	1.564287	71	4893.396	high
8.774953	10.484479	15.24985	11.29794	54	4354.497	high
11.73314	13.32736	41.4126	10.82829	133	10299.17	high
1.545699	16.969516	15.82091	2.741136	85	5807.594	high
5.79138	7.0855891	3.941395	6.053188	60	5297.619	high
9.170015	12.631757	14.58635	9.655586	191	12421.62	high
23.10905	27.616385	59.34368	8.097191	89	7331.546	high
1.756157	2.3801281	6.132869	0.738401	16	2658.562	high
9.10436	30.622961	22.30889	27.55411	180	9579.272	high
12.49017	84.071641	63.63018	19.74946	124	6964.059	high
8.501956	43.698221	62.93994	11.714	140	8129.805	high
19.58133	40.474973	43.47507	12.34147	146	7923.016	high
69.13099	49.272599	88.10739	49.40037	142	8344.336	high
31.54865	70.619797	85.88066	39.44656	151	9114.625	high
34.87672	50.750882	82.18418	76.46127	122	7363.57	high
48.43795	25.502601	134.5207	185.5336	189	10627.98	high
63.27051	63.734385	71.20093	23.19015	77	6530.135	high

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
21.68563	34.488671	52.52895	19.30017	133	7243.99	medium
24.89767	27.41463	2.47939	6.646195	67	4737.671	medium
43.26506	57.888944	2.559653	6.14617	112	6130.388	medium
82.86365	128.63335	6.264181	6.454516	106	5633.319	medium
17.63683	18.431514	5.814561	4.238876	59	4553.409	medium
27.67867	25.966121	1.664364	8.212888	91	4716.847	medium
10.91639	13.653838	22.82422	3.923429	53	5782.756	medium
15.56334	22.899753	2.452612	5.447058	71	4983.778	medium
10.60728	25.240261	32.42719	19.6499	79	5794.9	medium
8.882162	21.027699	15.84878	11.02086	185	8032.229	medium
24.15734	31.674635	7.358795	3.037715	84	5110.335	medium
6.061553	10.325946	3.551692	6.646886	75	5169.298	medium
11.91086	20.111856	2.389326	1.650191	70	5496.636	medium
5.753454	13.688764	24.59385	12.26733	167	8899.119	medium
14.84904	18.36349	1.388408	4.112098	68	4345.071	medium
9.294447	21.580404	18.61739	10.37489	109	6429.228	medium
33.10624	34.802675	3.293493	6.873613	46	4225.107	medium
4.443881	22.182816	26.42955	2.320349	56	4840.962	medium
44.71535	48.981865	4.747993	5.145678	77	5148.874	medium
15.71374	26.170319	3.252404	4.565616	65	4678.913	medium
7.178719	31.449169	9.790842	5.239778	82	6863.991	medium
24.74947	32.770519	2.958445	7.774731	89	5665.579	medium
23.7664	23.544515	1.741747	4.146419	63	4231.277	medium
6.481751	8.7431795	3.848645	5.12172	67	5142.198	medium
19.81015	24.014522	7.483312	6.777078	92	6945.892	medium
7.628355	9.4152943	1.743525	2.74054	51	4040.223	medium
52.16796	79.553498	12.71485	17.33831	177	7462.193	medium
39.45459	60.212726	24.28104	51.07466	140	6960.173	medium
36.90891	36.061218	34.51003	15.64748	92	6025.493	medium
30.56586	112.51015	95.72653	23.62015	87	4607.918	medium
19.06331	36.784502	42.69694	13.44102	101	5974.351	medium
28.86784	26.097493	3.853614	6.053327	71	5288.539	medium
9.726545	34.11901	53.84981	10.52981	97	6441.265	medium
50.4756	66.18645	4.855458	17.74149	83	5449.177	medium
23.09312	17.017045	2.69373	7.737451	53	4533.629	medium
40.65414	45.627769	24.04786	14.07028	244	11210.66	medium
4.061336	7.2983305	8.87351	25.4557	87	5912.545	medium
5.731031	7.776249	20.14534	5.705331	5	1731.072	medium
81.63165	106.77973	6.253988	22.93175	96	4730.825	medium
8.606188	11.89845	10.75383	7.878199	104	9393.76	medium

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
70.1453	63.122059	2.278244	4.278547	49	4238.46	medium
11.60461	10.146281	1.509817	2.001433	43	4116.495	medium
12.6548	15.551574	1.24162	3.265329	56	4973.546	medium
96.28992	59.336192	6.428607	12.4364	85	5915.013	medium
22.44872	38.072124	9.756936	10.64996	124	6425.049	medium
43.5664	33.101881	7.387061	13.60372	62	4245.987	medium
13.69429	29.043701	1.90954	4.325624	74	4483.982	medium
172.6722	183.53391	5.464143	17.93212	79	5150.962	medium
47.87682	60.816757	2.364859	9.765072	88	4565.651	medium
11.63652	16.347139	27.84153	9.124338	87	4796.095	medium
25.0185	41.077736	2.802871	11.60617	112	6380.909	medium
64.3079	106.27983	1.958706	3.659993	78	5276.477	medium
4.410654	20.140562	25.47055	3.584733	88	4415.627	medium
20.97549	24.682132	3.568565	4.915178	95	6250.958	medium
11.73293	103.06382	148.1354	28.89441	49	5426.099	medium
14.00746	22.246603	7.514698	5.558693	98	5488.879	medium
85.01429	115.05728	7.407403	12.98974	117	5775.604	medium
13.19342	12.952885	1.721627	1.635254	35	4078.353	medium
12.70496	21.023091	5.207403	11.56438	99	5302.513	medium
100.4356	165.17733	58.81262	12.56443	76	6748.377	medium
12.18013	29.072599	7.230646	6.559186	163	8048.786	medium
12.43149	15.393326	30.66703	9.431851	135	8934.202	medium
10.10123	23.471673	17.202	10.92181	121	6174.179	medium
25.13229	56.672773	4.420706	12.60705	99	6124.368	medium
43.81566	45.491221	3.166048	4.944484	77	5107.234	medium
82.86365	128.63335	6.264181	6.454516	106	5633.319	medium
26.29021	32.394796	4.411015	8.481181	114	7036.507	medium
30.37435	41.042342	13.24663	8.473055	66	5681.335	medium
32.91907	37.947862	2.221417	3.315639	74	4782.757	medium
12.12312	28.087372	4.355079	15.5263	141	6888.054	medium
17.41374	30.852536	14.2713	22.20404	139	6445.326	medium
20.43829	37.4926	12.96529	27.64907	111	6725.247	medium
10.45025	14.924883	4.689381	5.357509	70	6201.361	medium
12.38634	35.549442	6.158085	12.39869	159	7168.96	medium
28.58941	65.584796	15.0618	57.80283	138	7904.555	medium
56.85093	65.189507	13.7439	13.18668	108	6640.826	medium
19.91444	21.923698	7.297775	6.175223	105	7557.323	medium
3.937407	5.152844	2.110869	2.769444	35	3672.38	medium
7.741796	9.3111502	8.110572	5.015769	73	6135.001	medium
5.517616	10.341625	5.271209	39.31293	92	6416.594	medium

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
5.168218	7.7936653	7.438222	38.02683	70	5123.536	medium
10.31131	25.027783	59.35687	8.395724	72	6196.065	medium
31.83042	31.75624	6.05182	9.817693	87	6941.549	medium
13.64182	22.639692	39.25337	11.35384	82	7576.1	medium
8.801457	9.4470307	2.509765	2.494576	45	4848.254	medium
15.16985	19.797958	9.318581	38.96423	86	6229.731	medium
28.40568	16.453865	9.096779	2.084607	187	10078.18	medium
16.91666	16.536167	3.925081	6.642699	102	6077.023	medium
4.543263	10.59551	2.480329	8.14443	144	9897.976	medium
5.698449	10.347718	4.100012	5.800488	111	7254.906	medium
5.918952	10.174329	16.11231	33.19867	127	7510.794	medium
8.007642	8.8012694	5.548916	10.73523	124	6918.424	medium
11.08073	18.690009	3.724653	7.984873	117	5921.224	medium
9.812311	25.076419	19.89422	26.60415	222	9070.771	medium
11.7303	16.784998	5.279705	29.14197	110	6703.174	medium
9.149846	20.856136	27.34265	7.467202	51	4569.171	medium
21.2808	16.707901	2.907955	9.012268	103	6680.65	medium
15.33138	51.586648	52.51509	17.59463	114	5986.853	medium
5.043652	14.315516	4.363216	6.452906	81	5695.467	medium
51.379	50.265881	2.85862	9.810348	81	5135.815	medium
99.22631	129.41395	2.131463	6.20494	77	4645.284	medium
22.02212	30.15834	1.826171	2.690469	104	5251.376	medium
39.12625	53.770959	7.220695	3.343727	81	4754.847	medium
11.50793	19.003955	1.545207	3.393182	64	4818.256	medium
4.21592	6.4611527	2.087095	3.942516	67	4688.575	medium
33.54806	54.740509	16.45493	59.40079	194	8704.552	medium
21.11864	18.078713	2.003868	6.802624	59	4945.352	medium
11.39257	20.00783	4.575575	7.889005	109	5618.444	medium
27.38474	39.549359	15.4068	11.59806	94	5111.021	medium
69.56414	93.268224	7.778593	10.28417	90	6246.838	medium
11.607	9.128216	3.63968	4.977362	59	4730.239	medium
10.12825	16.367236	9.278314	7.473204	108	6085.346	medium
4.125431	17.182174	16.3419	3.538129	85	4323.379	medium
4.354265	7.7060625	5.17899	4.821174	67	4232.473	medium
5.916298	10.826504	5.452639	5.778285	62	4164.414	medium
7.298213	11.08535	5.873859	7.925469	96	6390.517	medium
16.51317	26.796707	2.346903	4.207939	65	4930.918	medium
32.23382	44.441182	6.879887	8.98092	96	6136.255	medium
41.54334	40.651915	4.843485	21.10898	88	6529.689	medium
26.29021	32.394796	4.411015	8.481181	114	7036.507	medium

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
28.02372	33.915064	5.464054	4.280609	92	5870.515	medium
62.2288	68.072787	7.641991	14.10059	105	5420.621	medium
34.51993	49.733457	1.519167	3.968278	92	5869.716	medium
27.63467	25.246173	1.884522	4.352841	74	4884.607	medium
5.580406	10.44041	12.74987	5.776467	53	4874.967	medium
29.34699	32.725378	26.46677	14.81932	146	9537.069	medium
70.75772	107.02519	15.759	62.01953	147	6864.662	medium
28.8188	13.087184	74.04002	13.57467	144	9508.679	medium
2.420399	8.9040711	16.26076	3.642406	55	5260.009	medium
10.46023	6.2958768	16.6433	5.227085	82	6899.888	medium
7.760073	12.347491	6.059972	9.002166	93	5620.692	medium
38.06781	35.537448	7.447333	8.032539	65	5070.673	medium
23.0154	23.213875	4.604499	13.06007	109	6675.483	medium
26.85907	40.456415	8.865549	9.352131	134	5648.614	medium
24.97312	33.168813	2.684985	8.079547	85	5708.81	medium
3.461096	7.6182863	14.54931	19.77662	108	6726.282	medium
6.748982	15.728588	32.05385	6.126588	105	5703.579	medium
44.71806	63.041408	6.342855	15.22393	96	6203.351	medium
31.29217	22.407149	2.619757	12.77314	92	6317.866	medium
11.13182	14.953258	2.088185	9.419105	101	6165.084	medium
37.64885	32.972261	7.161707	8.041497	113	6338.113	medium
84.70865	85.273124	78.84747	3.971953	74	5709.834	medium
7.4006	16.389378	8.79814	3.317956	154	8007.957	medium
5.825661	16.375115	4.795346	5.356539	67	5130.967	medium
39.39782	36.048612	23.68719	5.616877	46	5179.225	medium
100.4356	165.17733	58.81262	12.56443	76	6748.377	medium
4.689108	13.295383	36.05879	11.58309	161	8206.749	medium
16.02903	25.872687	2.850971	6.023731	72	5000.18	medium
75.19254	55.930971	4.763418	9.460037	76	4651.327	medium
3.012015	10.043256	3.335151	3.663704	78	4871.374	medium
20.19124	29.44797	34.55254	6.760417	53	4794.595	medium
18.99992	25.422309	3.415818	3.948397	77	6129.525	medium
91.04966	110.45352	238.6441	34.66629	110	7153.461	medium
83.78922	91.277439	220.9843	14.86254	189	11027.57	medium
16.08014	24.919807	25.60467	21.00041	146	7538.649	medium
10.81895	19.755756	29.56863	43.69038	163	9063.414	medium
13.70505	19.863816	23.46022	15.96734	50	3162.456	medium
5.422233	6.9589698	4.068623	3.859683	67	4042.323	medium
4.961971	4.8795735	5.092161	4.736401	84	4959.629	medium
80.90526	39.109943	214.136	59.69425	393	14532.42	medium

dahi	mata	pipi	mulut	jumlah_kontur	panjang_total_kontur	label
82.78604	44.409876	89.60734	16.14917	303	12549.62	medium
41.57255	83.948851	49.80234	19.20786	133	7168.132	medium
30.91046	21.076561	143.6963	9.807075	120	9224.612	medium
31.91695	26.012379	7.868571	23.63554	57	5651.66	medium
32.06587	34.040249	4.299272	13.23739	103	8371.7	medium
6.605048	10.329135	20.87457	6.095918	168	8380.33	medium

Lampiran 3. Source Code

```
from flask import Flask, render_template, request, redirect,
url_for, send_from_directory
import os
import cv2
import numpy as np
import mediapipe as mp
import joblib
from werkzeug.utils import secure_filename
```

app.py

```
# Konfigurasi upload folder
UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

# Load model dan preprocessing tools
model = joblib.load('model/bpnn_model.pkl')
try:
    scaler = joblib.load('model/scaler.pkl')
except:
    scaler = None
try:
    label_encoder = joblib.load('model/label_encoder.pkl')
    if label_encoder is not None:
        print(f"Loaded label encoder classes:
{label_encoder.classes_}") # Debug print
except:
    label_encoder = None

# Inisialisasi face mesh dari MediaPipe
```



```

mp_face_mesh = mp.solutions.face_mesh

# Fungsi untuk mengecek ekstensi file
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

# Fungsi ekstraksi fitur GLCM
def ekstrak_glmc_fitur(area):
    from skimage.feature import graycomatrix, graycoprops
    glcm = graycomatrix(area, distances=[1], angles=[0], levels=256,
symmetric=True, normed=True)
    contrast = graycoprops(glcm, 'contrast')[0,0]
    homogeneity = graycoprops(glcm, 'homogeneity')[0,0]
    energy = graycoprops(glcm, 'energy')[0,0]
    return (contrast + homogeneity + energy) // 3

# Fungsi prediksi keriput berdasarkan area wajah
def predict_regions(img):
    results = {}
    img_copy = img.copy()
    with mp_face_mesh.FaceMesh(static_image_mode=True,
max_num_faces=1, refine_landmarks=True,
min_detection_confidence=0.5) as face_mesh:
        rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        res = face_mesh.process(rgb)

        # Jika wajah tidak terdeteksi
        if not res.multi_face_landmarks:
            return None, img_copy

        # Landmark wajah
        face_landmarks = res.multi_face_landmarks[0].landmark
        img_h, img_w = img.shape[:2]

        # Koordinat area wajah
        atas = int(face_landmarks[10].y * img_h)
        bawah = int(face_landmarks[152].y * img_h)
        kiri = int(face_landmarks[234].x * img_w)
        kanan = int(face_landmarks[454].x * img_w)
        tinggi = bawah - atas
        h_bagian = tinggi // 4

        # Membuat masker wajah

```

```

        face_oval_idx = [10, 338, 297, 332, 284, 251, 389, 356, 454,
323, 361, 288, 397, 365, 379, 378, 400, 377, 152, 148, 176, 149,
150, 136, 172, 58, 132, 93, 234, 127, 162, 21, 54, 103, 67, 109]
        face_oval = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in face_oval_idx], np.int32)
        mask_face = np.zeros(img.shape[:2], dtype=np.uint8)
        cv2.fillPoly(mask_face, [face_oval], 255)
        kernel = np.ones((25,25), np.uint8)
        mask_face = cv2.erode(mask_face, kernel, iterations=1)

        # Hilangkan area mata, bibir, hidung agar tidak dihitung
        under_left_eye_idx = [33, 246, 161, 160, 159, 158, 157, 173,
133, 155, 154, 153, 145, 144, 163, 7]
        under_left_eye = np.array([(int(face_landmarks[i].x *
img_w), int(face_landmarks[i].y * img_h)) for i in
under_left_eye_idx], np.int32)
        cv2.fillPoly(mask_face, [under_left_eye], 0)

        under_right_eye_idx = [263, 466, 388, 387, 386, 385, 384,
398, 362, 398, 382, 381, 380, 374, 373, 249]
        under_right_eye = np.array([(int(face_landmarks[i].x *
img_w), int(face_landmarks[i].y * img_h)) for i in
under_right_eye_idx], np.int32)
        cv2.fillPoly(mask_face, [under_right_eye], 0)

        lips_idx = [61, 146, 91, 181, 84, 17, 314, 405, 321, 375,
291, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 185, 40, 39, 37,
0, 267, 269, 270, 409, 415, 310, 311, 312, 13, 82, 81, 42, 183, 78]
        lips = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in lips_idx], np.int32)
        cv2.fillPoly(mask_face, [lips], 0)

        left_eye_idx = [33, 246, 161, 160, 159, 158, 157, 173]
        right_eye_idx = [263, 466, 388, 387, 386, 385, 384, 398]
        left_eye = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in left_eye_idx], np.int32)
        right_eye = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in right_eye_idx], np.int32)
        cv2.fillPoly(mask_face, [left_eye], 0)
        cv2.fillPoly(mask_face, [right_eye], 0)

        kernel_eye = np.ones((13,13), np.uint8)
        mask_face = cv2.erode(mask_face, kernel_eye, iterations=1)

```

```

left_eyebrow_idx = [70, 63, 105, 66, 107]
right_eyebrow_idx = [336, 296, 334, 293, 300]
left_eyebrow = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in left_eyebrow_idx],
np.int32)
right_eyebrow = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in right_eyebrow_idx],
np.int32)
cv2.fillPoly(mask_face, [left_eyebrow], 0)
cv2.fillPoly(mask_face, [right_eyebrow], 0)

nostril_idx = [195, 5, 4, 45, 275, 440, 344, 278, 98, 327,
326, 2, 97, 326, 327, 2]
nostril = np.array([(int(face_landmarks[i].x * img_w),
int(face_landmarks[i].y * img_h)) for i in nostril_idx], np.int32)
cv2.fillPoly(mask_face, [nostril], 0)
kernel_nostril = np.ones((15,15), np.uint8)
mask_face = cv2.erode(mask_face, kernel_nostril,
iterations=1)

# Definisi area wajah untuk analisis
margin_x = 5
margin_y = 5
margin_dahi_top = 10
area_list = [
    (kiri + margin_x, atas + margin_y + margin_dahi_top,
kanan - margin_x, atas + h_bagian - margin_y, 'Dahi'),
    (kiri + margin_x, atas + h_bagian + margin_y, kanan -
margin_x, atas + 2*h_bagian - margin_y, 'Mata'),
    (kiri + margin_x, atas + 2*h_bagian + margin_y, kanan -
margin_x, atas + 3*h_bagian - margin_y, 'Pipi'),
    (kiri + margin_x, atas + 3*h_bagian + margin_y, kanan -
margin_x, bawah - margin_y, 'Mulut')
]

# List untuk menyimpan hasil fitur
fitur_list = []
kontur_list = []
panjang_kontur_list = []

# Ekstraksi fitur tiap area
for (x1, y1, x2, y2, label) in area_list:
    roi = img[y1:y2, x1:x2]
    mask_roi = mask_face[y1:y2, x1:x2]

```



```

# Validasi ukuran ROI
if roi.size == 0 or roi.shape[0] < 10 or roi.shape[1] <
10:
    fitur_list.append(0.0)
    kontur_list.append(0)
    panjang_kontur_list.append(0.0)
    continue

    gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY) if
len(roi.shape) == 3 else roi

    # Preprocessing berbeda untuk dahi
    if label == 'Dahi':
        clahe_dahi = cv2.createCLAHE(clipLimit=6.0,
tileGridSize=(32,32))
        gray_eq_dahi = clahe_dahi.apply(gray)
        adapt = cv2.adaptiveThreshold(gray_eq_dahi, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 5, 7)
        sobelx = cv2.Sobel(gray_eq_dahi, cv2.CV_64F, 1, 0,
ksize=5)
        sobely = cv2.Sobel(gray_eq_dahi, cv2.CV_64F, 0, 1,
ksize=5)
        sobel = cv2.convertScaleAbs(sobelx) +
cv2.convertScaleAbs(sobely)
        combined = cv2.add(adapt, sobel)
        canny_low, canny_high = 1, 6
        kernel_close = np.ones((7,7), np.uint8)
    else:
        clahe = cv2.createCLAHE(clipLimit=4.0,
tileGridSize=(32,32))
        gray_eq = clahe.apply(gray)
        blur = cv2.GaussianBlur(gray_eq, (3,3), 0)
        combined = cv2.adaptiveThreshold(blur, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 7, 5)
        canny_low, canny_high = 2, 10
        kernel_close = np.ones((3,3), np.uint8)

    # Ekstrak fitur GLCM
    if label == 'Dahi':
        fitur = ekstrak_glcm_fitur(gray_eq_dahi)
    else:
        fitur = ekstrak_glcm_fitur(gray_eq)
    fitur_list.append(fitur)

```

```

# Deteksi tepi dengan Canny
edges = cv2.Canny(combined, canny_low, canny_high)
edges = cv2.bitwise_and(edges, edges, mask=mask_roi)
edges = cv2.morphologyEx(edges, cv2.MORPH_CLOSE,
kernel_close, iterations=1)

# Thinning agar garis lebih tipis
try:
    from cv2.ximgproc import thinning
    edges = thinning(edges)
except Exception:
    pass

# Hitung jumlah kontur dan panjangnya
contours, _ = cv2.findContours(edges, cv2.RETR_LIST,
cv2.CHAIN_APPROX_NONE)
kontur_list.append(len(contours))
panjang_total = sum([cv2.arcLength(cnt, False) for cnt
in contours])
panjang_kontur_list.append(panjang_total)

# Gambar kontur
for cnt in contours:
    cnt_offset = cnt + np.array([[x1, y1]])
    cv2.drawContours(img_copy, [cnt_offset], -1, (255,
0, 0), 1)

# Jika fitur tidak lengkap
if len(fitur_list) != 4:
    for idx, (x1, y1, x2, y2, label) in
enumerate(area_list):
        results[label] = {'status': '-', 'nilai': 0.0}
else:
    # Gabungkan fitur
    jumlah_kontur = sum(kontur_list)
    panjang_total_kontur = sum(panjang_kontur_list)
    fitur_input = fitur_list + [jumlah_kontur,
panjang_total_kontur]
    X = np.array([fitur_input])

# Normalisasi jika scaler ada
if scaler is not None:
    X = scaler.transform(X)

```

```

# Prediksi model
try:
    pred = model.predict(X)[0]
    print(f"Prediksi model: {pred}") # Logging prediksi
model

if label_encoder is not None:
    try:
        if pred in label_encoder.classes_:
            status =
label_encoder.inverse_transform([pred])[0]
            status = status.lower()
            print(f"Prediksi status: {status}") #
Debug print
        else:
            status = str(pred).lower()
            print(f"Peringatan: Label prediksi tidak
dikenal oleh label encoder, menggunakan langsung: {status}")
    except Exception as e:
        print(f"Error saat inverse transform label:
{e}")
        status = str(pred).lower()
    else:
        status = str(pred)

    if not status or status == '-':
        status = 'tidak diketahui'
except Exception as e:
    print(f"Error saat prediksi: {e}")
    status = '-'

# Evaluasi tiap area
for idx, (x1, y1, x2, y2, label) in
enumerate(area_list):
    nilai_fitur = fitur_list[idx]
    jumlah_kontur_area = kontur_list[idx]
    panjang_kontur_area = panjang_kontur_list[idx]
    area_size = (x2 - x1) * (y2 - y1)
    density_kontur = panjang_kontur_area /
max(area_size, 1)
    if nilai_fitur > 20 or jumlah_kontur_area > 12 or
density_kontur > 0.008:
        area_status = 'high'

```



```

        else:
            area_status = 'medium'
            results[label] = {'status': area_status, 'nilai':
nilai_fitur}

    return results, img_copy

# Routing halaman utama
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files['file']
        if file.filename == '':
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            filepath = os.path.join(app.config['UPLOAD_FOLDER'],
filename)
            file.save(filepath)
            img = cv2.imread(filepath)
            results, img_with_line = predict_regions(img)
            print(f"DEBUG: results = {results}")
            output_filename = f"output_{filename}"
            output_path = os.path.join('output', output_filename)
            cv2.imwrite(output_path, img_with_line)
            return render_template('result.html', filename=filename,
results=results, output_filename=output_filename)
        return render_template('index.html')

# Routing file upload
@app.route('/uploads/<filename>')
def uploaded_file(filename):
    return send_from_directory(app.config['UPLOAD_FOLDER'],
filename)

# Routing file output
@app.route('/output/<filename>')
def output_file(filename):
    return send_from_directory('output', filename)

# Routing halaman webcam
@app.route('/webcam')

```

```
def webcam():
    return render_template('webcam.html')
```

Main program

```
if __name__ == '__main__':
    app.run(debug=True)
```

ekstraksi_fitur_revisi.py

```
import cv2
import numpy as np
from skimage.feature import graycomatrix, graycoprops

# Fungsi untuk ekstraksi fitur GLCM pada area tertentu
def ekstrak_glcml_fitur(area):
    """Ekstraksi fitur GLCM untuk area tertentu"""
    if area.size == 0: # Jika area kosong
        return 0.0

    # Konversi ke grayscale jika masih berwarna
    if len(area.shape) == 3:
        area = cv2.cvtColor(area, cv2.COLOR_BGR2GRAY)

    # Hitung matriks GLCM
    glcm = graycomatrix(area, distances=[1], angles=[0], levels=256,
        symmetric=True, normed=True)
    contrast = graycoprops(glcm, 'contrast')[0,0] # Kontras
    homogeneity = graycoprops(glcm, 'homogeneity')[0,0] # Homogenitas
    energy = graycoprops(glcm, 'energy')[0,0] # Energi
    return (contrast + homogeneity + energy) / 3 # Rata-rata fitur GLCM

# Fungsi untuk ekstraksi fitur dari seluruh area wajah
def ekstrak_fitur_keriput(img):
    """Ekstraksi fitur untuk semua area wajah yang dibutuhkan"""
    h, w = img.shape[:2]

    # Potong area wajah: dahi, mata, pipi, mulut
    dahi = img[30:80, 50:150]
    mata = img[70:110, 60:140]

    pipi_kiri = img[90:130, 30:70]
    pipi_kanan = img[90:130, 130:170]
```

```

pipi = np.concatenate([pipi_kiri, pipi_kanan], axis=1)

mulut = img[140:180, 70:130]

# Hitung fitur GLCM dari setiap area
fitur_dahi = ekstrak_glcml_fitur(dahi)
fitur_mata = ekstrak_glcml_fitur(mata)
fitur_pipi = ekstrak_glcml_fitur(pipi)
fitur_mulut = ekstrak_glcml_fitur(mulut)

# Deteksi tepi untuk menghitung jumlah dan panjang kontur
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
if len(img.shape) == 3 else img
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)
edges = cv2.Canny(img_blur, 30, 80)
contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
jumlah_kontur = len(contours) # Jumlah kontur
panjang_total_kontur = sum([cv2.arcLength(cnt, False) for cnt in
contours]) # Panjang total kontur

# Return fitur dalam bentuk list
return [fitur_dahi, fitur_mata, fitur_pipi, fitur_mulut,
jumlah_kontur, panjang_total_kontur]

# Import fungsi preprocessing untuk memuat dan memproses citra
from preprocessing import load_and_preprocess_images

# Eksekusi utama jika file dijalankan langsung
if __name__ == "__main__":
    import os
    import pandas as pd

    data = [] # List untuk menyimpan fitur
    base_dir = "dataset"
    kelas_list = ["high", "medium"] # Dua kelas keriput

    # Looping tiap kelas
    for kelas in kelas_list:
        folder = os.path.join(base_dir, kelas)
        label = kelas
        images, filenames = load_and_preprocess_images(folder) #
Load dataset
        for img, filename in zip(images, filenames):

```



```

    fitur = ekstrak_fitur_keriput(img) # Ekstrak fitur
    data.append({
        'dahi': fitur[0],
        'mata': fitur[1],
        'pipi': fitur[2],
        'mulut': fitur[3],
        'jumlah_kontur': fitur[4],
        'panjang_total_kontur': fitur[5],
        'label': label
    })

    # Simpan hasil ekstraksi ke file CSV
    os.makedirs(base_dir, exist_ok=True)
    df = pd.DataFrame(data)
    df.to_csv(os.path.join(base_dir, "keriput_dataset.csv"),
index=False)
    print(f"Ekstraksi selesai. Data disimpan di
{os.path.join(base_dir, 'keriput_dataset.csv')}")

```

preprocessing.py

```

import pandas as pd

# Path file CSV hasil ekstraksi fitur
csv_path = "d:\\deteksi_keriput\\dataset\\keriput_dataset.csv"

# Fungsi untuk memuat dataset dari CSV
def load_dataset_from_csv(csv_path):
    df = pd.read_csv(csv_path)

    # Ambil kolom fitur
    X = df[["dahi", "mata", "pipi", "mulut", "jumlah_kontur",
"panjang_total_kontur"]].values
    # Ambil kolom label
    y = df["label"].values

    return X, y

# Load dataset dan tampilkan informasi awal
X, y = load_dataset_from_csv(csv_path)
print(f"Jumlah data: {len(X)}")
print(f"Label unik: {set(y)}")
print("5 data pertama:")

```

```

print(X[:5], y[:5])

# Fungsi untuk load dataset sekaligus melakukan preprocessing
numerik
def load_and_preprocess_images(folder):
    """
    Load dataset from CSV dan lakukan preprocessing sederhana pada
    kolom numerik.
    Return X (fitur) dan y (label).
    """
    # Saat ini parameter folder tidak digunakan, path CSV sudah
    fixed
    csv_path = "d:\\deteksi_keriput\\dataset\\keriput_dataset.csv"
    df = load_dataset_from_csv(csv_path) # ⚠ df di sini sebenarnya
    berupa tuple (X, y), bukan DataFrame
    if df is None or df.empty:
        print("Dataset kosong atau tidak ditemukan.")
        return None, None

    # Import scaler untuk normalisasi
    from sklearn.preprocessing import StandardScaler
    scaler = StandardScaler()

    # Pilih kolom numerik
    numeric_cols = df.select_dtypes(include=['int64',
    'float64']).columns
    if not numeric_cols.empty:
        df[numeric_cols] = scaler.fit_transform(df[numeric_cols])

    # Pisahkan fitur dan label jika ada kolom label
    if 'label' in df.columns:
        X = df.drop(columns=['label'], errors='ignore')
        y = df['label']
    else:
        X = df
        y = None

    return X, y

```

split_dataset.py

```

import pandas as pd
from sklearn.model_selection import train_test_split
import os

# Fungsi untuk membagi dataset menjadi train dan test
def split_dataset(csv_path, train_ratio=0.8):

    # Membaca dataset dari file CSV
    df = pd.read_csv(csv_path)

    # Membagi dataset menjadi train dan test dengan stratifikasi
    # Label
    train_df, test_df = train_test_split(df, train_size=train_ratio,
    stratify=df['label'], random_state=42)

    # Menentukan lokasi penyimpanan file hasil split
    base_dir = os.path.dirname(csv_path)
    train_path = os.path.join(base_dir, 'keriput_train.csv')
    test_path = os.path.join(base_dir, 'keriput_test.csv')

    # Simpan data train dan test ke CSV
    train_df.to_csv(train_path, index=False)
    test_df.to_csv(test_path, index=False)

    # Informasi hasil split
    print(f"Dataset dibagi menjadi {train_ratio*100}% data pelatihan
    dan {(1-train_ratio)*100}% data pengujian.")
    print(f"Data pelatihan disimpan di: {train_path}")
    print(f"Data pengujian disimpan di: {test_path}")

# Eksekusi utama jika dijalankan langsung
if __name__ == "__main__":
    csv_path = "dataset/keriput_dataset.csv"
    split_dataset(csv_path)

```

train_bpnn_enhanced.py

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

```



```

from sklearn.preprocessing import LabelEncoder, StandardScaler
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
import os
import time
from datetime import datetime
import warnings
warnings.filterwarnings('ignore')

# -----
# Fungsi: Load dataset train & test, scaling, dan encoding label
# -----
def load_and_prepare_data():
    """Load and prepare data for training."""

    # Membaca dataset train dan test
    train_data = pd.read_csv("dataset/keriput_train.csv")
    test_data = pd.read_csv("dataset/keriput_test.csv")

    # Tampilkan distribusi label
    print('Distribusi label di dataset training:')
    print(train_data['label'].value_counts())
    print('\nDistribusi label di dataset testing:')
    print(test_data['label'].value_counts())

    # Tentukan fitur yang digunakan
    fitur_names = ['dahi', 'mata', 'pipi', 'mulut']
    if 'jumlah_kontur' in train_data.columns and
'panjang_total_kontur' in train_data.columns:
        fitur_names += ['jumlah_kontur', 'panjang_total_kontur']

    print(f"\nFitur yang digunakan: {fitur_names}")

    # Data training + augmentasi dengan menambahkan noise kecil
    X_train = []
    y_train = []
    for idx, row in train_data.iterrows():
        fitur = [row[nama] for nama in fitur_names]
        label = row['label']

        X_train.append(fitur)
        y_train.append(label)

```

```

        for _ in range(3):
            augmented = [f + np.random.normal(0, 0.01) for f in
fitur]

            X_train.append(augmented)
            y_train.append(label)

# Data testing
X_test = []
y_test = []
for idx, row in test_data.iterrows():
    fitur = [row[nama] for nama in fitur_names]
    label = row['label']
    X_test.append(fitur)
    y_test.append(label)

# Encode label ke bentuk numerik
label_encoder = LabelEncoder()
y_train_enc = label_encoder.fit_transform(y_train)
y_test_enc = label_encoder.transform(y_test)

# Konversi ke numpy array
X_train = np.array(X_train)
X_test = np.array(X_test)

# Normalisasi fitur dengan StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

return X_train_scaled, X_test_scaled, y_train_enc, y_test_enc,
label_encoder, scaler, fitur_names

# -----
# Fungsi: Uji berbagai konfigurasi hidden Layer MLPClassifier
# -----
def test_hidden_layer_configurations():
    """Test various hidden layer configurations."""

    # Daftar konfigurasi hidden layer yang diuji
    configurations = [
        (8, 8), (16, 16), (32, 32), (64, 64), (128, 128),
        (16, 8), (32, 8), (64, 8), (128, 8),
        (8, 16), (16, 32), (32, 64), (64, 128),
        (32, 16), (64, 32), (128, 64),

```

```

        (8, 32), (16, 64), (32, 128),
        (8, 4), (16, 4), (32, 4), (64, 4),
        (4, 8), (4, 16), (4, 32), (4, 64)
    ]

    print(f"Total konfigurasi yang akan diuji:
    {len(configurations)}")

    # Load data
    X_train, X_test, y_train, y_test, label_encoder, scaler,
    feature_names = load_and_prepare_data()

    results = []
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")

    # Buat folder output untuk hasil eksperimen
    os.makedirs("output", exist_ok=True)
    os.makedirs(f"output/experiments_{timestamp}", exist_ok=True)

    # Loop semua konfigurasi hidden Layer
    for idx, hidden_layers in enumerate(configurations):
        print(f"\n{'='*60}")
        print(f"Menguji konfigurasi {idx+1}/{len(configurations)}:
        {hidden_layers}")
        print(f"{'='*60}")

        start_time = time.time()

        try:
            # Inisialisasi model MLP
            mlp = MLPClassifier(
                hidden_layer_sizes=hidden_layers,
                max_iter=1000,
                random_state=42,
                early_stopping=True,
                validation_fraction=0.2,
                n_iter_no_change=50,
                verbose=False
            )

            # Training model
            mlp.fit(X_train, y_train)

            # Prediksi pada data test

```



```

y_pred = mlp.predict(X_test)

# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
training_time = time.time() - start_time

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Simpan plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
            yticklabels=label_encoder.classes_)
plt.title(f"Confusion Matrix - Hidden Layers:
{hidden_layers}\nAccuracy: {accuracy:.4f}")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.savefig(f"output/experiments_{timestamp}/confusion_m
atrix_{hidden_layers}.png")
plt.close()

# Simpan kurva Loss training
plt.figure(figsize=(10, 5))
plt.plot(mlp.loss_curve_)
plt.title(f"Training Loss - Hidden Layers:
{hidden_layers}")
plt.xlabel("Iterations")
plt.ylabel("Loss")
plt.grid(True)
plt.tight_layout()
plt.savefig(f"output/experiments_{timestamp}/training_lo
ss_{hidden_layers}.png")
plt.close()

# Simpan hasil
result = {
    'configuration': str(hidden_layers),
    'hidden_layers': hidden_layers,
    'accuracy': accuracy,
    'training_time': training_time,
    'iterations': len(mlp.loss_curve_),

```

```

        'final_loss': mlp.loss_curve_[-1] if mlp.loss_curve_
else None
    }

    results.append(result)

    print(f"Akurasi: {accuracy:.4f}")
    print(f"Waktu training: {training_time:.2f} detik")
    print(f"Iterasi: {result['iterations']}")
    print(f"Loss akhir: {result['final_loss']:.4f} if
result['final_loss'] else "Loss akhir: N/A")

    # Laporan klasifikasi
    print("\nClassification Report:")
    print(classification_report(y_test, y_pred,
target_names=label_encoder.classes_))

    except Exception as e:
        print(f"Error pada konfigurasi {hidden_layers}:
{str(e)}")
        result = {
            'configuration': str(hidden_layers),
            'hidden_layers': hidden_layers,
            'accuracy': 0.0,
            'training_time': 0.0,
            'iterations': 0,
            'final_loss': None,
            'error': str(e)
        }
        results.append(result)

    # Simpan hasil eksperimen ke CSV
    results_df = pd.DataFrame(results)
    results_df = results_df.sort_values('accuracy', ascending=False)
    results_df.to_csv(f"output/experiments_{timestamp}/hidden_layer_
results.csv", index=False)

    # Buat visualisasi ringkasan
    create_summary_visualization(results_df, timestamp)

    # Ambil model terbaik
    best_result = results_df.iloc[0]
    best_config = eval(best_result['configuration'])

```

```

print(f"\n{'='*60}")
print("KONFIGURASI TERBAIK DITEMUKAN!")
print(f"\n{'='*60}")
print(f"Konfigurasi terbaik: {best_config}")
print(f"Akurasi tertinggi: {best_result['accuracy']:.4f}")
print(f"Waktu training: {best_result['training_time']:.2f}
detik")

# Retrain model terbaik
print("\nMenyimpan model terbaik...")
best_mlp = MLPClassifier(
    hidden_layer_sizes=best_config,
    max_iter=1000,
    random_state=42,
    early_stopping=True,
    validation_fraction=0.2,
    n_iter_no_change=50
)

best_mlp.fit(X_train, y_train)

# Simpan model, scaler, dan label encoder
os.makedirs("model", exist_ok=True)
joblib.dump(best_mlp, f'model/bpnn_model_best_{timestamp}.pkl')
joblib.dump(scaler, f'model/scaler_best_{timestamp}.pkl')
joblib.dump(label_encoder,
f'model/label_encoder_best_{timestamp}.pkl')

# Simpan versi generik (untuk prediksi langsung)
joblib.dump(best_mlp, 'model/bpnn_model.pkl')
joblib.dump(scaler, 'model/scaler.pkl')
joblib.dump(label_encoder, 'model/label_encoder.pkl')

print(f"Model terbaik disimpan dengan timestamp: {timestamp}")
print("Model juga disimpan sebagai bpnn_model.pkl untuk
prediksi")

return results_df, timestamp

# -----
# Fungsi: Buat ringkasan visualisasi hasil eksperimen
# -----
def create_summary_visualization(results_df, timestamp):
    """Create summary visualizations of all experiments."""

```



```

# Plot perbandingan akurasi, waktu training, dan top 10
konfigurasi
plt.figure(figsize=(15, 8))

# Plot accuracy comparison
plt.subplot(2, 2, 1)
configurations = results_df['configuration'].astype(str)
accuracies = results_df['accuracy']
plt.bar(range(len(configurations)), accuracies)
plt.xlabel('Configuration Index')
plt.ylabel('Accuracy')
plt.title('Accuracy Comparison Across Configurations')
plt.xticks(range(len(configurations)), configurations,
rotation=45, ha='right')
plt.grid(True, alpha=0.3)
for i, v in enumerate(accuracies):
    plt.text(i, v + 0.01, f'{v:.3f}', ha='center', va='bottom',
fontsize=8)

# Plot top 10 best configurations
top_10 = results_df.head(10)
plt.subplot(2, 2, 2)
plt.barh(range(len(top_10)), top_10['accuracy'])
plt.yticks(range(len(top_10)), top_10['configuration'])
plt.xlabel('Accuracy')
plt.title('Top 10 Best Configurations')
plt.grid(True, alpha=0.3)

# Plot training time comparison
plt.subplot(2, 2, 3)
plt.bar(range(len(configurations)), results_df['training_time'])
plt.xlabel('Configuration Index')
plt.ylabel('Training Time (seconds)')
plt.title('Training Time Comparison')
plt.xticks(range(len(configurations)), configurations,
rotation=45, ha='right')
plt.grid(True, alpha=0.3)

# Scatter plot Accuracy vs Training Time
plt.subplot(2, 2, 4)
plt.scatter(results_df['training_time'], results_df['accuracy'],
c=range(len(results_df)), cmap='viridis', alpha=0.7)
plt.xlabel('Training Time (seconds)')

```

```

plt.ylabel('Accuracy')
plt.title('Accuracy vs Training Time')
plt.colorbar(label='Configuration Index')
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig(f"output/experiments_{timestamp}/hidden_layer_comparison_{timestamp}.png", dpi=300)
plt.close()

# Simpan Laporan ringkasan eksperimen
with
open(f"output/experiments_{timestamp}/summary_report_{timestamp}.txt", 'w') as f:
    f.write("LAPORAN HASIL EKSPERIMEN BPNN\n")
    f.write("=" * 50 + "\n\n")
    f.write(f"Total konfigurasi yang diuji: {len(results_df)}\n")
    f.write(f"Waktu eksperimen: {timestamp}\n\n")

    f.write("KONFIGURASI TERBAIK:\n")
    f.write("-" * 30 + "\n")
    best = results_df.iloc[0]
    f.write(f"Konfigurasi: {best['configuration']}\n")
    f.write(f"Akurasi: {best['accuracy']:.4f}\n")
    f.write(f"Waktu training: {best['training_time']:.2f} detik\n")
    f.write(f"Iterasi: {best['iterations']}\n\n")

    f.write("10 KONFIGURASI TERBAIK:\n")
    f.write("-" * 30 + "\n")
    for idx, row in results_df.head(10).iterrows():
        f.write(f"{idx+1}. {row['configuration']} - Akurasi: {row['accuracy']:.4f} - Waktu: {row['training_time']:.2f}s\n")

    f.write("\nSEMUA HASIL:\n")
    f.write("-" * 30 + "\n")
    for idx, row in results_df.iterrows():
        f.write(f"{row['configuration']} - Akurasi: {row['accuracy']:.4f} - Waktu: {row['training_time']:.2f}s\n")

# -----
# Main eksekusi eksperimen training BPNN
# -----

```

```

if __name__ == "__main__":
    print("Memulai eksperimen BPNN dengan berbagai konfigurasi
hidden layer...")
    results_df, timestamp = test_hidden_layer_configurations()

    print("\n" + "="*60)
    print("EKSPERIMEN SELESAI!")
    print("="*60)
    print(f"Hasil lengkap tersedia di:
output/experiments_{timestamp}/")
    print(f"Model terbaik tersimpan di:
model/bpnn_model_best_{timestamp}.pkl")
    print(f"Skala terbaik tersimpan di:
model/scaler_best_{timestamp}.pkl")
    print(f"Label encoder tersimpan di:
model/label_encoder_best_{timestamp}.pkl")

```

evaluate_model.py

```

import pandas as pd
import numpy as np
import joblib
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# -----
# 1. Load dataset hasil ekstraksi fitur
# -----
df = pd.read_csv('dataset/keriput_dataset.csv')

# Pastikan semua fitur yang dibutuhkan tersedia
required_features = ['dahi', 'mata', 'pipi', 'mulut',
'jumlah_kontur', 'panjang_total_kontur']
for col in required_features:
    if col not in df.columns:
        raise ValueError(f"Kolom '{col}' tidak ditemukan di dataset.
Pastikan ekstraksi fitur sudah benar.")

# Pisahkan fitur (X) dan Label (y)
X = df[required_features].values
y_true = df['label'].values

# -----
# 2. Load scaler (normalisasi) jika tersedia

```



```

# -----
try:
    scaler = joblib.load('model/scaler.pkl')
    X_scaled = scaler.transform(X)
except:
    scaler = None
    X_scaled = X

# -----
# 3. Load model BPNN hasil training
# -----
model = joblib.load('model/bpnn_model.pkl')

# -----
# 4. Load label encoder jika digunakan saat training
# -----
try:
    label_encoder = joblib.load('model/label_encoder.pkl')
    y_true_enc = label_encoder.transform(y_true)
except:
    label_encoder = None
    y_true_enc = y_true

# -----
# 5. Prediksi dengan model
# -----
y_pred = model.predict(X_scaled)

# Jika label encoder ada, kembalikan hasil prediksi ke bentuk label
# asli
if label_encoder is not None:
    y_pred_label = label_encoder.inverse_transform(y_pred)
    y_true_label = y_true
else:
    y_pred_label = y_pred
    y_true_label = y_true

# -----
# 6. Evaluasi model: Akurasi, Confusion Matrix, dan Classification
# Report
# -----
print('Akurasi:', accuracy_score(y_true_label, y_pred_label))
print('\nConfusion Matrix:')
print(confusion_matrix(y_true_label, y_pred_label))

```

```
print('\nClassification Report (precision, recall, f1-score):')
print(classification_report(y_true_label, y_pred_label, digits=4))
```

main.py

```
from preprocessing import load_and_preprocess_images_from_csv
from ekstraksi_fitur_revisi import ekstrak_fitur_keriput
import joblib
import pandas as pd
import numpy as np
import os

# -----
# 1. Load gambar & Label dari CSV
# -----
images, filenames, labels =
load_and_preprocess_images_from_csv('dataset/keriput_dataset/')

# Ekstraksi fitur untuk setiap gambar
fitur_data = [ekstrak_fitur_keriput(img) for img in images]

# -----
# 2. Load model, scaler, dan Label encoder
# -----
try:
    # Coba load model default
    model = joblib.load('model/bpnn_model.pkl')
    scaler = joblib.load('model/scaler.pkl')
    label_encoder = joblib.load('model/label_encoder.pkl')
    print("Model berhasil dimuat dari model/bpnn_model.pkl")
except FileNotFoundError:
    # Jika model default tidak ada, cari model terbaik (best)
    berdasarkan timestamp
    model_files = [f for f in os.listdir('model') if
f.startswith('bpnn_model_best_') and f.endswith('.pkl')]
    if model_files:
        latest_model = sorted(model_files)[-1] # Ambil model
        terbaru
        latest_timestamp = latest_model.replace('bpnn_model_best_',
        '').replace('.pkl', '')

        model = joblib.load(f'model/{latest_model}')
        scaler =
joblib.load(f'model/scaler_best_{latest_timestamp}.pkl')
```

```

        label_encoder =
joblib.load(f'model/label_encoder_best_{latest_timestamp}.pkl')
        print(f"Model terbaik berhasil dimuat: {latest_model}")
    else:
        raise FileNotFoundError("Tidak ada model yang ditemukan di
folder model/")

# -----
# 3. Preprocessing fitur
# -----
fitur_array = np.array(fitur_data)
fitur_scaled = scaler.transform(fitur_array)

# -----
# 4. Prediksi keriput dengan model
# -----
prediksi_encoded = model.predict(fitur_scaled)
prediksi = label_encoder.inverse_transform(prediksi_encoded)

# -----
# 5. Simpan hasil prediksi ke CSV
# -----
df_hasil = pd.DataFrame(fitur_data, columns=["dahi", "mata", "pipi",
"mulut", "jumlah_kontur", "panjang_total_kontur"])
df_hasil['prediksi_keriput'] = prediksi
df_hasil['nama_file'] = filenames

os.makedirs('output', exist_ok=True)
df_hasil.to_csv('output/hasil_prediksi.csv', index=False)

# -----
# 6. Informasi hasil prediksi
# -----
print("Hasil prediksi disimpan di output/hasil_prediksi.csv")
print(f"Total gambar diproses: {len(images)}")
print(f"Distribusi prediksi:
{df_hasil['prediksi_keriput'].value_counts()}")

```

model_utils.py

```

import joblib
import os
import numpy as np

```

```

# -----

```



```

# Fungsi untuk Load model, scaler, dan Label encoder
# -----
def load_model_components():
    """
    Load model, scaler, dan label encoder dengan fallback mechanism.
    Jika bpnn_model.pkl tidak ada, otomatis mencari model terbaik
    dengan timestamp.

    Returns:
        tuple: (model, scaler, label_encoder)
    """
    try:
        # Coba Load model default
        model = joblib.load('model/bpnn_model.pkl')
        scaler = joblib.load('model/scaler.pkl')
        label_encoder = joblib.load('model/label_encoder.pkl')
        print("Model berhasil dimuat dari model/bpnn_model.pkl")
        return model, scaler, label_encoder
    except FileNotFoundError:
        # Jika tidak ada, cari model terbaik berdasarkan timestamp
        model_files = [f for f in os.listdir('model') if
            f.startswith('bpnn_model_best_') and f.endswith('.pkl')]
        if model_files:
            latest_model = sorted(model_files)[-1] # Ambil model
            latest_timestamp =
            latest_model.replace('bpnn_model_best_', '').replace('.pkl', '')

            model = joblib.load(f'model/{latest_model}')
            scaler =
            joblib.load(f'model/scaler_best_{latest_timestamp}.pkl')
            label_encoder =
            joblib.load(f'model/label_encoder_best_{latest_timestamp}.pkl')
            print(f"Model terbaik berhasil dimuat: {latest_model}")
            return model, scaler, label_encoder
        else:
            raise FileNotFoundError("Tidak ada model yang ditemukan
            di folder model/")

# -----
# Fungsi untuk prediksi dengan model
# -----
def predict_with_model(features):
    """

```

Melakukan prediksi menggunakan model BPNN yang sudah diload.

Args:

features: array fitur (bisa 1 sampel atau banyak sampel)

Returns:

predictions: hasil prediksi dalam bentuk label asli
"""

```
model, scaler, label_encoder = load_model_components()

# Jika input hanya 1 sampel (1D array), reshape ke 2D
features_scaled = scaler.transform(
    np.array(features).reshape(1, -1) if
len(np.array(features).shape) == 1 else features
)

# Prediksi label encoded
predictions_encoded = model.predict(features_scaled)
# Decode ke label asli (misalnya "high", "medium")
predictions =
label_encoder.inverse_transform(predictions_encoded)

return predictions

# -----
# Fungsi untuk melihat model apa saja yang tersedia
# -----
def get_model_info():
    """
    Menampilkan daftar file model (.pkl) yang ada di folder 'model'.

    Returns:
        list: daftar nama file model
    """
    model_files = [f for f in os.listdir('model') if
f.endswith('.pkl')]
    print("Model yang tersedia:")
    for f in sorted(model_files):
        print(f" - {f}")

    return model_files
```

deteksi_kamera.py

```

import cv2
import numpy as np
import mediapipe as mp
import joblib

# Load label encoder jika tersedia
try:
    label_encoder = joblib.load('model/label_encoder.pkl')
except:
    label_encoder = None

# Load model utama
model = joblib.load('model/bpnn_model.pkl')

# Load scaler jika tersedia
try:
    scaler = joblib.load('model/scaler.pkl')
except:
    scaler = None

# Akses kamera (webcam)
cap = cv2.VideoCapture(0)

# Inisialisasi modul FaceMesh dari MediaPipe
mp_face_mesh = mp.solutions.face_mesh
mp_drawing = mp.solutions.drawing_utils

# Fungsi ekstraksi fitur GLCM
def ekstrak_glcml_fitur(area):
    import skimage.feature
    glcm = skimage.feature.graycomatrix(area, distances=[1],
    angles=[0], levels=256, symmetric=True, normed=True)
    contrast = skimage.feature.graycoprops(glcm, 'contrast')[0,0]
    homogeneity = skimage.feature.graycoprops(glcm,
    'homogeneity')[0,0]
    energy = skimage.feature.graycoprops(glcm, 'energy')[0,0]
    return (contrast + homogeneity + energy) / 3

# Fungsi menggambar area wajah + ekstraksi fitur
def draw_face_areas_landmarks(img, landmarks, img_w, img_h):

    # Tentukan batas area wajah berdasarkan Landmark
    atas = int(landmarks[10].y * img_h)
    bawah = int(landmarks[152].y * img_h)

```



```

kiri = int(landmarks[234].x * img_w)
kanan = int(landmarks[454].x * img_w)
tinggi = bawah - atas
h_bagian = tinggi // 4

# Daftar area wajah (dahi, mata, pipi, mulut)
area_list = [
    (kiri, atas, kanan, atas + h_bagian, (0,255,255), 'Dahi'),
    (kiri, atas + h_bagian, kanan, atas + 2*h_bagian, (0,255,0),
'Mata'),
    (kiri, atas + 2*h_bagian, kanan, atas + 3*h_bagian,
(255,0,255), 'Pipi'),
    (kiri, atas + 3*h_bagian, kanan, bawah, (0,0,255), 'Mulut')
]
fitur_list = []
import cv2

# Fungsi preprocessing ROI (CLAHE + grayscale)
def preprocess_roi(roi):
    if roi.size == 0 or roi.shape[0] < 10 or roi.shape[1] < 10:
        return None
    roi_resized = cv2.resize(roi, (200, 200))
    gray = cv2.cvtColor(roi_resized, cv2.COLOR_BGR2GRAY) if
len(roi_resized.shape) == 3 else roi_resized
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    gray_clahe = clahe.apply(gray)
    return gray_clahe

# Looping untuk setiap area (dahi, mata, pipi, mulut)
for (x1, y1, x2, y2, color, label) in area_list:
    cv2.rectangle(img, (x1, y1), (x2, y2), color, 2) #
Gambarkan kotak
    roi = img[y1:y2, x1:x2] # Potong ROI
    if roi.size == 0 or roi.shape[0] < 10 or roi.shape[1] < 10:
        print(f"ROI kosong untuk area {label}")
        fitur_list.append(None)
        continue
    preprocessed = preprocess_roi(roi)
    if preprocessed is None:
        print(f"Preprocessing gagal untuk area {label}")
        fitur_list.append(None)
        continue
    fitur = ekstrak_glcmm_fitur(preprocessed)
    print(f"Nilai fitur {label}: {fitur}")

```

```

        fitur_list.append(fitur)

    # Hitung kontur (jumlah & panjang total)
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) if
len(img.shape) == 3 else img
    img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)
    edges = cv2.Canny(img_blur, 30, 80)
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    jumlah_kontur = len(contours)
    panjang_total_kontur = sum([cv2.arcLength(cnt, False) for cnt in
contours])
    fitur_list.append(jumlah_kontur)
    fitur_list.append(panjang_total_kontur)

    # Prediksi dan tampilkan status di gambar
    for idx, (x1, y1, x2, y2, color, label) in enumerate(area_list):
        fitur = fitur_list[idx]
        if None in fitur_list: # Jika ada ROI gagal diproses
            status = '-'
            nilai = 0.0
        else:
            nilai = fitur
            X_pred = np.array([fitur_list])
            if scaler is not None:
                X_pred = scaler.transform(X_pred)
                pred = model.predict(X_pred)

            if label_encoder is not None:
                status = label_encoder.inverse_transform(pred)[0]
            else:
                status = str(pred[0])

        # Format Label prediksi agar Lebih rapi
        status = status.replace('_', ' ')
        status = ' '.join([w.capitalize() for w in
status.split()])
        print(f"Prediksi area {label}: {status} (fitur:
{nilai})")

    # Tampilkan hasil prediksi di layar
    text = f"{label}: {status} ({nilai:.2f})"
    font = cv2.FONT_HERSHEY_SIMPLEX
    font_scale = 1.1

```

```

thickness = 3

cv2.putText(img, text, (x1+8, y1+34), font, font_scale,
(0,0,0), thickness+2, cv2.LINE_AA)
cv2.putText(img, text, (x1+5, y1+30), font, font_scale,
(0,255,255), thickness, cv2.LINE_AA)

# Jalankan FaceMesh dan Looping kamera
with mp_face_mesh.FaceMesh(static_image_mode=False, max_num_faces=1,
refine_landmarks=True, min_detection_confidence=0.5) as face_mesh:
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        img_h, img_w = frame.shape[:2]
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = face_mesh.process(rgb)
        if results.multi_face_landmarks:
            for face_landmarks in results.multi_face_landmarks:
                draw_face_areas_landmarks(frame,
face_landmarks.landmark, img_w, img_h)
        cv2.imshow('Deteksi Keriput (Tekan q untuk keluar)', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()

```


Lampiran 4. Permohonan Data Penelitian Kepada Ketua Program Studi Informatika.

Nomor : Makassar, 1 Dzulhijah 1446 H
Lamp : - 28 Mei 2025 M
Hal : Permohonan Data Penelitian

Kepada Yang Terhormat,
Ketua Program Studi Informatika
Di -
Tempat

Assalamualaikum Warahmatullahi Wabarakatuh

Dengan Rahmat Allah SWT, Semoga aktivitas kita bernilai ibadah di Sisi-Nya. Dalam rangka penyelesaian Tugas Akhir pada Studi Informatika dengan judul "Analisis Distribusi Keriput Wajah Berbasis Segmentasi Area Spesifik Menggunakan Metode BPNN untuk Estimasi Penuaan Personalized" bersama ini kami sampaikan mahasiswa.

Stambuk	Nama
105841102421	Magfiratul Jannah

Sehubungan dengan hal tersebut, maka kami memohon dibuatkan surat pengantar pada instansi di bawah ini:

Nama Instansi : CitraCosmetic
Alamat : Jl. Boulevard No.5 Kel Blok A2, Masale, Kec. Panakkukang, Kota Makassar.

Demikian surat kami atas perhatian dan kerja samanya kami hanturkan banyak terima kasih.
Jazakumullah Khaeran Kasiran,
Wassalamu Alaikum Warahmatullahi wabarakatuh.

Yang Mengusulkan:

Magfiratul Jannah
105841102421

Gedung Menara Ibra Lantai 3
Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
Web: <http://cs7.teknik.unismuh.ac.id/>, e-mail: teknik@unismuh.ac.id, pengairan@unismuh.ac.id

   Kampus Merdeka

Lampiran 5. Surat Permohonan Data Penelitian ke LP3M UNISMUH Makassar

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

Nomor : 035/05/IF/C.4-VI/V/46/2025
 Lamp. : -
 Hal : Permohonan Data Penelitian

Makassar, 4 Dzulhijjah 1446 H
 31 Mei 2025 M

Kepada yang Terhormat,
Ketua LP3M Unismuh Makassar
 Di -
 Tempat

Assalamu 'Alaikum Warahmatullahi Wabarakatuh

Dengan Rahmat Allah SWT, Semoga aktivitas kita bermilai ibadah di Sisi-Nya. Dalam rangka penyelesaian Tugas Akhir pada Program Studi Informatika dengan judul "Analisis Distribusi Wajah Berbasis Segmentasi Area Spesifik Menggunakan Metode BPNN". Bersama ini kami sampaikan mahasiswa

Stambuk	Nama
105.84.11024.21	Magfiratul Jannah

Sehubungan dengan hal tersebut, maka kami memohon dibuatkan surat pengantar pada instansi di bawah ini:

Nama Instansi : Citra Cosmetic
 Alamat : Jln. Boulevard No.5 Blok A2, Masale, Kec. Panakkukang, Kota Makassar
 Demikian surat kami atas perhatian dan kerja samanya kami haturkan banyak terima kasih.
Jazakumullahi Khairan Katsiran
Wassalamu 'Alaikum warahmatullahi Wabarakatuh

Ketua Program Studi
 Informatika,

 Muhaimin A.M Havat, S.Kom., M.T.
 NBM 0931087901

Tembusan:

1. Dekan Fakultas Teknik
2. Arsip

Gedung Menara Iqra Lantai 3
 Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
 Web: <https://teknik.unismuh.ac.id/>, e-mail: teknik@unismuh.ac.id, pengajaran@unismuh.ac.id

Kampus Merdeka

Lampiran 6. Surat Rekomendasi Penelitian dari LP3M

 **MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH**
UNIVERSITAS MUHAMMADIYAH MAKASSAR
LEMBAGA PENELITIAN PENGEMBANGAN DAN PENGABDIAN KEPADA MASYARAKAT
Jl. Sultan Alauddin No. 259 Telp.866972 Fax (0411)865588 Makassar 90221 e-mail :dp3m@unismuh.ac.id

Nomor : 7157/05/C.4-VIII/VI/1446/2025
Lamp : 1 (satu) Rangkap Proposal
Hal : Permohonan Izin Penelitian

03 June 2025 M
07 Dzulhijjah 1446

Kepada Yth,
Bapak Gubernur Prov. Sul-Sel
Cq. Kepala Dinas Penanaman Modal & PTSP Provinsi Sulawesi Selatan
di -
Makassar

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Berdasarkan surat Dekan Fakultas Teknik Universitas Muhammadiyah Makassar, nomor: 035/05/IF/C.4-VI/V/46/2025 tanggal 31 Mei 2025, menerangkan bahwa mahasiswa tersebut di bawah ini :

Nama : **MAGFIRATUL JANNAH**
No. Stambuk : **10584 1102421**
Fakultas : **Fakultas Teknik**
Jurusan : **Informatika**
Pekerjaan : **Mahasiswa**
Bermaksud melaksanakan penelitian/pengumpulan data dalam rangka penulisan Skripsi dengan judul :

"ANALISIS DISTRIBUSI WAJAH BERBASIS SEGMENTASI AREA SPESIFIK MENGGUNAKAN METODE BPNN"

Yang akan dilaksanakan dari tanggal 10 Juni 2025 s/d 10 Agustus 2025.

Sehubungan dengan maksud di atas, kiranya Mahasiswa tersebut diberikan izin untuk melakukan penelitian sesuai ketentuan yang berlaku.
Demikian, atas perhatian dan kerjasamanya diucapkan Jazakumullahu khaeran

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Ketua LP3M,


Dr. Muh. Arief Muhsin, M.Pd.
NBM 1127761

Lampiran 7. Surat Izin Penelitian dari PTSP


PEMERINTAH PROVINSI SULAWESI SELATAN
DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU
Jl. Bougenville No.5 Telp. (0411) 441077 Fax. (0411) 448936
Website : <http://simap-new.sulselprov.go.id> Email : ptsp@sulselprov.go.id
Makassar 90231

Nomor : 12369/S.01/PTSP/2025	Kepada Yth.
Lampiran : -	Walikota Makassar
Perihal : <u>Izin penelitian</u>	

di-
Tempat

Berdasarkan surat Ketua LP3M UNISMUH Makassar Nomor : 7157/05/C.4-VIII/VI/1446/2025 tanggal 03 Juni 2025 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

N a m a	: MAGFIRATUL JANNAH
Nomor Pokok	: 105841102421
Program Studi	: Informatika
Pekerjaan/Lembaga	: Mahasiswa (S1)
Alamat	: Jl. Sit Alauddin No 259, Makassar

PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI, dengan judul :

" ANALISIS DISTRIBUSI KERIPUT WAJAH BERBASIS SEGMENTASI AREA SPESIFIK MENGGUNAKAN METODE BACKPROPAGATION NEURAL NETWORK (BPNN) UNTUK ESTIMASI PENUAAN PERSONALIZED "

Yang akan dilaksanakan dari : Tgl. **10 Juni s/d 10 Juli 2025**

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami **menyetujui** kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar
Pada Tanggal 10 Juni 2025

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU
SATU PINTU PROVINSI SULAWESI SELATAN



ASRUL SANI, S.H., M.Si.
Pangkat : PEMBINA TINGKAT I
Nip : 19750321 200312 1 008

Tembusan Yth

1. Ketua LP3M UNISMUH Makassar di Makassar;
2. Peringgal.

Lampiran 8. Hasil Turnitin

 **MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH**
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat Kantor: Jl. Sultan Alauddin No.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Magfiratul Jannah
Nim : 105841102421
Program Studi, Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	2%	10 %
2	Bab 2	0%	25 %
3	Bab 3	0%	15 %
4	Bab 4	9%	10 %
5	Bab 5	5%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 27 Agustus 2025
Mengetahui,
Kepala UPT- Perpustakaan dan Penerbitan,


Nursimah S. Hum, M.P.
NBM: 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Bab I Magfiratul Jannah

105841102421

by Tahap Tutup

The logo of Universitas Muhammadiyah Makassar is a blue shield-shaped emblem. It features a central yellow sunburst with Arabic calligraphy. The text 'UNIVERSITAS MUHAMMADIYAH' is written in a semi-circle at the top, and 'MAKASSAR' is at the bottom. On the left side, 'UPT PUSTAKAAN DAN PENERBITAN' is written vertically. The shield is flanked by two yellow stars and a green laurel wreath.

Submission date: 27-Aug-2025 01:21PM (UTC+0700)
Submission ID: 2736028477
File name: BAB_1_-_2025-08-27T131921.248.docx (58.55K)
Word count: 1603
Character count: 10915

Bab I Magfiratul Jannah 105841102421

ORIGINALITY REPORT

2%	2%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Islam Indonesia	2%
	Student Paper	

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off



Bab II Magfiratul Jannah

105841102421

by Tahap Tutup

Submission date: 27-Aug-2025 01:22PM (UTC+0700)
Submission ID: 2736028646
File name: BAB_2_-_2025-08-27T131919.921.docx (182.77K)
Word count: 4140
Character count: 27601

Bab II Magfiratul Jannah 105841102421

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off



Bab III Magfiratul Jannah

105841102421

by Tahap Tutup

Submission date: 27-Aug-2025 01:26PM (UTC+0700)
Submission ID: 2736030276
File name: BAB_3_-_2025-08-27T131919.921.docx (1.17M)
Word count: 2685
Character count: 17425

Bab III Magfiratul Jannah 105841102421

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off



Bab IV Magfiratul Jannah

105841102421

by Tahap Tutup

The logo of Universitas Muhammadiyah Makassar is a blue shield-shaped emblem. It features a central yellow sunburst with Arabic calligraphy. The text 'UNIVERSITAS MUHAMMADIYAH' is written in a semi-circle at the top, and 'MAKASSAR' is in the center. The bottom part of the shield contains the text 'PERPUSTAKAAN DAN PENERBITAN'.

Submission date: 27-Aug-2025 01:35PM (UTC+0700)
Submission ID: 2736032955
File name: BAB_4_-_2025-08-27T131919.821.docx (3.7M)
Word count: 4572
Character count: 29069

Bab IV Magfiratul Jannah 105841102421

ORIGINALITY REPORT

9%	4%	6%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Middlesex University Student Paper	1%
2	Submitted to Konsorsium Perguruan Tinggi Swasta Indonesia II Student Paper	1%
3	repository.its.ac.id Internet Source	<1%
4	Diah Fatma Sjoraida, Bucky Wibawa Karya Guna, Dudi Yudhakusuma. "Analisis Sentimen Film Dirty Vote Menggunakan BERT (Bidirectional Encoder Representations from Transformers)", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2024 Publication	<1%
5	eprints.uny.ac.id Internet Source	<1%
6	Muhammad Khatama Insani, Dwi Budi Santoso. "Perbandingan Kinerja Model Pre-Trained CNN (VGG16, RESNET, dan INCEPTIONV3) untuk Aplikasi Pengenalan Wajah pada Sistem Absensi Karyawan", Jurnal Indonesia : Manajemen Informatika dan Komunikasi, 2024 Publication	<1%
7	Submitted to Universitas Muslim Indonesia Student Paper	<1%

- 8 Ismail Mohidin, Andi Mariani, Imaniar Abdullah PajiriPajiri. "APLIKASI E-VOTING PEMILIHAN KEPALA DESA DI KECAMATAN BOTUPINGGE BERBASIS WEB", Jurnal Teknologi Informasi Indonesia (JTII), 2017
Publication <1 %
- 9 Doly Ilham Saputra Huta Julu, Dewi Nurdiyah. "KLASIFIKASI SAMPAH ORGANIK DAN NON ORGANIK MENGGUNAKAN TRANSFER LEARNING", Jurnal Transformatika, 2025
Publication <1 %
- 10 Muhammad Saharullah Raiya, Muhammad Raihan Putra Khamil, Nur Fadillah, Rizal Adi Saputra. "Implementation of Long Short-Term Memory Algorithm on the Issue of Single Tuition Fee Increase on Student Interest in Studying", Jurnal Informatika Terpadu, 2025
Publication <1 %
- 11 brightideas.houstontx.gov
Internet Source <1 %
- 12 Nur Halizah Alfajr, Sofi Defiyanti. "PREDIKSI PENYAKIT JANTUNG MENGGUNAKAN METODE RANDOM FOREST DAN PENERAPAN PRINCIPAL COMPONENT ANALYSIS (PCA)", Jurnal Informatika dan Teknik Elektro Terapan, 2024
Publication <1 %
- 13 Panji Novantara, Risteruw Leonardo Firmansyah, Marrilyn Arismawati. "Deteksi Hama Penyakit Daun Padi Dengan Menggunakan Teknik Optimasi Deep Learning Convolutional Neural Network", bit-Tech, 2025 <1 %

14	Submitted to Sekolah Teknik Elektro & Informatika Student Paper	<1 %
15	garuda.kemdikbud.go.id Internet Source	<1 %
16	jurnal.unimor.ac.id Internet Source	<1 %
17	Anasthasya Averina, Helen Hadi, Joko Siswantoro. "Analisis Sentimen Multi-Kelas Untuk Film Berbasis Teks Ulasan Menggunakan Model Regresi Logistik", Teknika, 2022 Publication	<1 %
18	Suharyadi Pancono, Narwikant Indroasyoko, Asep Irfan Setiawan. "Pemantauan dan Deteksi Penyakit Daun Tomat Berbasis IoT dan CNN dengan Aplikasi Android", Indonesian Journal of Computer Science, 2024 Publication	<1 %
19	Submitted to Universitas Prima Indonesia Student Paper	<1 %
20	docplayer.info Internet Source	<1 %
21	fadiyni.blogspot.com Internet Source	<1 %
22	ojs.unud.ac.id Internet Source	<1 %
23	pemrogramanmatlab.com Internet Source	<1 %

repository.ub.ac.id

24	Internet Source	<1 %
25	Siti Hajar, Murinto Murinto, Anton Yudhana. "Identifikasi Jenis Daun untuk Ecoprint Menggunakan Metode Convolutional Neural Network", Jurnal Sains dan Informatika, 2025 Publication	<1 %
26	Submitted to Universitas Budi Luhur Student Paper	<1 %
27	faqs.tips Internet Source	<1 %
28	jurnal.uns.ac.id Internet Source	<1 %
29	Dhyanna Lisa Rahmadona Putri, Supatman . "KLASIFIKASI CITRA JENIS HIJAB MENGUNAKAN DENSENET-121", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1 %
30	digilib.uns.ac.id Internet Source	<1 %
31	dokumen.tips Internet Source	<1 %
32	es.scribd.com Internet Source	<1 %
33	v4nired.wordpress.com Internet Source	<1 %

Bab V Magfiratul Jannah

105841102421

by Tahap Tutup



Submission date: 27-Aug-2025 01:37PM (UTC+0700)
Submission ID: 2736033815
File name: BAB_5_-_2025-08-27T131918.273.docx (39.19K)
Word count: 330
Character count: 2263

Bab V Magfiratul Jannah 105841102421

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

www.slideshare.net

Internet Source

5%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off

