

Peningkatan Akurasi *DeepMelaNet* Melalui Penghapusan Garis Rambut dan Peningkatan Kontras Citra

<http://dx.doi.org/10.28932/jutisi.v12i2.13632>

Riwayat Artikel

Received: 27 Oktober 2025 | Final Revision: 23 Juli 2026 | Accepted: 23 Juli 2026

Creative Commons License 4.0 (CC BY – NC)



Nurvianto Akbar Ikhsani ^{#1}, Yufis Azhar ^{#2}

[#] Informatika, Universitas Muhammadiyah Malang
Jl. Raya Tlogomas No.246, Kota Malang, 65144, Indonesia

¹akbar.ikhsani123@webmail.umm.ac.id

²yufis@umm.ac.id

✉Corresponding author: akbar.ikhsani123@gmail.com

Abstrak — Melanoma adalah salah satu jenis kanker kulit yang paling berbahaya, berasal dari sel melanosit yang memproduksi melanin sebagai pigmen warna kulit. Penyakit ini dapat berkembang secara agresif dan berisiko tinggi menyebar ke organ tubuh lain jika tidak dideteksi sejak dini. Diagnosis yang akurat dapat mencegah perkembangan penyakit dan meminimalisir risiko kematian. Penelitian ini bertujuan untuk mengoptimalkan hasil klasifikasi melanoma berbasis citra dengan memanfaatkan model *DeepMelaNet* melalui integrasi dua teknik *preprocessing* utama, yaitu penghapusan garis rambut dan peningkatan kontras dengan metode *CLAHE*. Permasalahan kualitas citra dermoskopi yang kerap tidak konsisten menjadi hambatan dalam identifikasi detail klinis yang relevan, sehingga teknik *preprocessing* diperlukan untuk meningkatkan generalisasi dan reliabilitas model. Penelitian ini juga menerapkan beberapa augmentasi kemudian diproses lebih lanjut dengan eksperimen klasifikasi berbasis *DeepMelaNet*. Evaluasi dilakukan pada berbagai skenario kombinasi *preprocessing* untuk menilai dampak terhadap akurasi klasifikasi. Temuan utama menunjukkan bahwa urutan penerapan peningkatan kontras diikuti penghapusan garis rambut serta pengaturan *learning rate* secara optimal mampu meningkatkan akurasi validasi hingga 96,5%, melampaui pendekatan mandiri *DeepMelaNet* dan kombinasi lainnya sekaligus menghasilkan performa pelatihan yang paling stabil. Penelitian ini menegaskan bahwa pemilihan dan penyesuaian strategi *preprocessing* berperan krusial dalam meningkatkan akurasi dan generalisasi model *deep learning* untuk tugas klasifikasi melanoma, sehingga metode ini dapat direkomendasikan sebagai standar dalam analisis citra medis dermoskopi berbasis *deep learning*.

Kata kunci— *CLAHE*; *DeepMelaNet*; *EfficientNet*; Penghapusan Garis Rambut, Peningkatan Kontras Citra.

Enhancing DeepMelaNet Accuracy Through Hairline Removal and Image Contrast Enhancement

Abstract — Melanoma is one of the most dangerous types of skin cancer, originating from melanocytes that produce melanin as skin pigment. This disease can develop aggressively and has a high risk of spreading to other organs if not detected early. Accurate diagnosis makes it possible to prevent this disease and avoid death from melanoma. This study aims to optimize image-based melanoma

classification results by utilizing the DeepMelaNet model through the integration of two main preprocessing techniques, namely hair line removal and contrast enhancement using the CLAHE method. The problem of inconsistent dermoscopy image quality is an obstacle in identifying relevant clinical details, so preprocessing techniques are needed to improve the generalization and reliability of the model. This study also applies several augmentations which are then further processed with DeepMelaNet based classification experiments. Evaluations are carried out on various preprocessing combination scenarios to assess the impact on classification accuracy. The main findings show that the sequence of applying contrast enhancement followed by hairline removal and optimal learning rate adjustment can improve validation accuracy to 96.5%, surpassing the standalone DeepMelaNet approach and other combinations while producing the most stable training performance. This study confirms that the selection and adjustment of preprocessing strategies play a crucial role in improving the accuracy and generalization of deep learning models for melanoma classification tasks, so that this method can be recommended as a standard in deep learning based dermoscopy medical image analysis.

Keywords— CLAHE; Contrast Enhancement; DeepMelaNet; EfficientNet; Hair Removal.

I. PENDAHULUAN

Kulit manusia, sebagai organ terluas tubuh dan juga sebagai organ vital, terdiri dari tiga lapisan utama yaitu *epidermis* (lapisan terluar pelindung), *dermis* (tempat kelenjar keringat dan pembuluh darah), dan *subkutis* (jaringan lemak pengatur suhu). Di dalam kulit terdapat sel Melanosit yang berada di jaringan epidermis berfungsi sebagai produksi melanin yang melindungi DNA dari radiasi UV, namun paparan sinar matahari berlebihan atau mutasi genetik dapat menyebabkan transformasi ganas sel ini menjadi melanoma sebagai kanker kulit paling agresif yang menyumbang 75% kematian terkait kanker kulit [1]. Melanoma merupakan jenis kanker kulit yang berbahaya, dalam hal ini jenis yang paling umum muncul adalah melanoma ganas dan berbagai jenis Karsinoma Sel Basal (*Basal Cell Carcinoma/BCC*). Dengan bentuk *BCC* non-melanoma tumbuh sebagai beragam nodul kecil. Sedangkan Karsinoma Sel Skuamosa (*SCC*) memiliki bentuk seperti *BCC* akan tetapi dapat menyebabkan bercak merah bersisik, radang terbuka, pendarahan tidak teratur, dan benjolan kulit [2].

Menurut *International Agency for Research on Cancer (IARC)*, pada tahun 2022 diperkirakan sebanyak 330.000 kasus terkait dengan melanoma dan hampir 60.000 orang meninggal akibat penyakit tersebut [3]. Dan juga menurut *Skin Cancer Foundation* kanker kulit adalah kanker paling umum yang ada di Amerika Serikat dan di seluruh dunia, diperkirakan di tahun 2025 sebanyak 212.200 kasus terdiagnosis di Amerika Serikat, dari jumlah tersebut sebanyak 107.240 kasus akan bersifat *in-situ* (noninvasif), terbatas pada epidermis (lapisan atas kulit), dan 104.960 kasus akan bersifat invasif, menembus epidermis ke lapisan kedua kulit (*dermis*). Dari kasus invasif, 60.550 kasus akan dialami oleh pria dan 44.410 kasus akan dialami oleh wanita [4]. Melanoma terjadi karena perkembangan pada sel-sel yang memproduksi pigmen melanin di kulit dan menyebar ke bagian tubuh lainnya, sehingga melanoma dianggap sebagai jenis kanker kulit yang paling berbahaya dan serius karena cenderung cepat menyebar ke organ lain di dalam tubuh jika tidak segera ditangani secara dini [5]. Deteksi dini sangat penting mengingat penyebaran melanoma yang cepat dan laju pertumbuhannya yang tinggi, yang terutama menyerang punggung, lengan, kaki, dan wajah. Faktor risiko yang dapat diidentifikasi meliputi paparan sinar UV, banyak tahi lalat, kulit putih yang mudah terbakar, dan riwayat keluarga yang menderita penyakit ini [6]. Oleh karena itu deteksi dini menjadi kunci utama dalam upaya penanganan dan peningkatan harapan hidup pasien melanoma [7].

Penanganan melanoma secara manual oleh tenaga medis, baik melalui pemeriksaan fisik langsung maupun dermatoskopi, masih menghadapi tantangan besar berupa ketidakakuratan diagnosis akibat pola lesi yang kompleks dan sangat bervariasi. Penggunaan metode tradisional dalam mendeteksi melanoma, seperti pemeriksaan visual berbasis kriteria ABCDE (asimetri, batas, warna, diameter, evolusi). Penentuan dalam hal ini sangat bergantung pada pengalaman dan subjektivitas dokter kulit. Hal ini berisiko mengarah pada keterlambatan diagnosis dan pengobatan, yang sangat berpengaruh pada prognosis pasien [8]. Dalam dunia diagnosis medis yang mengandalkan citra, ketelitian menjadi landasan utama. Diagnosis dini dan akurat sangatlah krusial, namun proses manual yang ada saat ini seringkali padat karya dan rentan terhadap variabilitas. Implementasi CNN dan *Transfer Learning* terbukti meningkatkan sensitivitas dan akurasi diagnosis, terutama untuk membedakan lesi *benign* dan *malignant* dengan perbedaan visual sangat tipis [9].

Tantangan ini diperparah oleh masalah fundamental yang terletak pada kualitas citra itu sendiri. Citra medis seringkali memiliki variasi kualitas yang tinggi, dengan gangguan seperti *noise* dan rentang kontras yang lebar menjadi masalah umum. Faktor-faktor seperti perbedaan kondisi pencahayaan saat pengambilan citra dapat secara langsung berkontribusi pada masalah kontras ini. Kualitas citra yang tidak konsisten ini menciptakan rintangan yang signifikan, tidak hanya bagi para ahli medis tetapi juga bagi model *deep learning*. Fitur-fitur penting yang menandakan suatu penyakit bisa menjadi tidak jelas, sehingga menyulitkan model untuk melakukan identifikasi yang andal. Kehadiran rambut dalam gambar dermatoskopi menyebabkan terhalangnya informasi mengenai tekstur dan batas lesi [10]. Bentuk detail kritis, seperti 'batas lesi' (*lesion boundaries*), bisa menjadi kabur dan sulit dibedakan. Oleh karena itu, meningkatkan kualitas citra bukan lagi pilihan, melainkan sebuah keharusan. Teknik morfologi untuk *hair removal* secara signifikan meningkatkan deteksi lesi pada sistem *deep learning*, karena *noise* seperti garis rambut dapat mengaburkan batas lesi dan menurunkan akurasi segmentasi serta

klasifikasi otomatis [11]. Dengan adanya Perkembangan teknologi kecerdasan buatan, khususnya *deep learning*, telah membuka peluang besar dalam otomatisasi analisis citra dermoskopi untuk deteksi melanoma.

Hingga saat ini, berbagai studi telah dilakukan untuk mengembangkan metode dalam optimasi klasifikasi melanoma, dengan 2 jenis melanoma yaitu *Benign* (melanoma jinak) dan *Malignant* (melanoma ganas). Metode metode ini memanfaatkan tahap *preprocessing* dan beberapa jenis arsitektur untuk meningkatkan klasifikasi terhadap jenis melanoma. Hasil penelitian ini menunjukan akurasi yang bervariasi tergantung pada metode yang digunakan.

Penelitian oleh Dartiko [12], menggunakan *convolutional neural network (CNN)* dengan metode *hybrid preprocessing* yang terdiri dari *Contrast Limited Adaptive Histogram Equalization (CLAHE)*, *morphological closing* dan *median filter*. guna mengatasi masalah kontras citra yang rendah dan menghapus *noise* yang biasanya terdapat pada *epidermis* kulit berupa rambut-rambut halus. Hasil penelitian tersebut menunjukkan bahwa model *CNN* yang diintegrasikan dengan *hybrid preprocessing* berhasil mencapai nilai akurasi, presisi, *Recall*, dan *F1-Score* sebesar 78,19%.

Penelitian lainnya dilakukan oleh Al Huda et al. [5]. Menggunakan model bernama *DeepMelaNet* yang dibangun dengan mengadopsi arsitektur *EfficientNet* yang telah dimodifikasi dan dioptimalkan untuk analisis citra dermoskopi. Dengan menggunakan teknik *preprocessing* seperti augmentasi data untuk Menambahhkan variasi dalam *dataset* yang digunakan. Pada pengujian menggunakan *dataset* besar, *DeepMelaNet* berhasil mencapai akurasi hingga 93,4%.

Penelitian lainnya juga dilakukan oleh Kaur et al. [7]. Penelitian ini menggabungkan dua metode *preprocessing* yaitu teknik *Hair Removal* dan *Contrast Enhancement* menggunakan *context aggregation network*. Akan tetapi selama langkah penghilangan rambut, kontras dan resolusi citra kulit dapat terpengaruh secara signifikan. Hal ini disebabkan oleh rekonstruksi piksel pada citra keluaran, sehingga penerapan metode pasca-pemrosesan menjadi krusial untuk meningkatkan kembali kualitas citra. Setelah *preprocessing*, model ini melanjutkan ke tahap segmentasi lesi. Proses ini memanfaatkan *atrous Convolutional Neural Network (ACNN)*. Tahap terakhir adalah klasifikasi, yang menggunakan *novel deep convolutional neural network (N-DCNN)* yang dirancang khusus. Hasil penelitian menunjukkan bahwa model terintegrasi yang diusulkan ini berhasil meningkatkan akurasi klasifikasi hingga mencapai 93,4%.

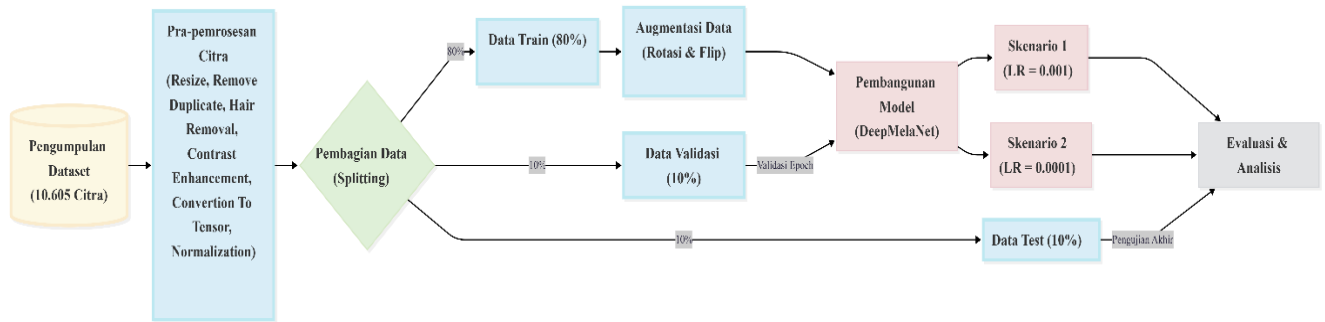
Namun, meskipun berbagai pendekatan telah menunjukkan hasil yang menjanjikan, tantangan signifikan masih ada, terutama dalam menjaga keseimbangan antara konsistensi akurasi dan efisiensi komputasi. Kompleksitas visual pada citra lesi kulit, seperti adanya artefak rambut atau kontras yang rendah, menuntut penanganan yang tepat agar fitur ekstraksi berjalan optimal. Selain itu, penggunaan arsitektur model yang kompleks sering kali memerlukan sumber daya komputasi yang besar. Oleh karena itu, diperlukan metode dengan optimasi arsitektur yang lebih efisien dan ringan, namun tetap mampu mempertahankan akurasi tinggi dalam mendiagnosis awal melanoma.

Penelitian ini mengusulkan sistem optimasi terhadap klasifikasi melanoma dengan menggunakan teknik *preprocessing* dan model *DeepMelaNet* pada dua skenario. Metode yang diajukan melibatkan enam tahap utama: pengumpulan *dataset*, *image preprocessing*, *splitting data*, augmentasi data, Pembangunan model, dan evaluasi. Penelitian ini memanfaatkan *Melanoma Skin Cancer Dataset of 10000 Images* yang diakses melalui website *Kaggle* [13]. Secara historis, *dataset* ini merupakan kompilasi citra dermoskopi yang ditarik dan digabungkan dari berbagai direktori berbeda di dalam repositori resmi *ISIC (International Skin Imaging ColLABoration) Melanoma Skin Cancer* dan terbagi ke dalam 2 kelas *dataset* yaitu *Benign* dan *Malignant*. Dengan demikian, batasan pada penelitian ini hanya melakukan klasifikasi terhadap 2 kelas yaitu *Benign* dan *Malignant* saja. Sebelum menerapkan tahap *preprocessing* lainnya citra diterapkan dua teknik *preprocessing* awal yaitu *Hair removal* dan *Contrast Enhancement* untuk menghapus *noise* yang ada pada *epidermis* kulit dan meningkatkan kualitas citra, setelah melakukan *preprocessing* awal dilanjutkan ke tahap *splitting data* dan transformasi menggunakan *PyTorch*. Proses klasifikasi menggunakan arsitektur dari model *DeepMelaNet* yang telah di bangun sebelumnya untuk mendapatkan tingkat akurasi berdasarkan proses klasifikasi biner. Pada skenario pertama membandingkan setiap metode *preprocessing* utama dengan parameter yang sudah ditentukan, pada skenario kedua melakukan perbandingan yang sama terhadap setiap metode *preprocessing* utama akan tetapi yang menjadi pembeda adalah parameter yang akan digunakan nantinya

Tujuan penelitian ini adalah untuk mengembangkan metode yang dapat meningkatkan akurasi dalam klasifikasi melanoma serta efisiensi performa komputasi di banding dengan metode yang ada. Diharapkan metode yang diberikan dapat memberikan kontribusi yang signifikan terhadap diagnosa medis yang akan membantu kedepannya.

II. METODE

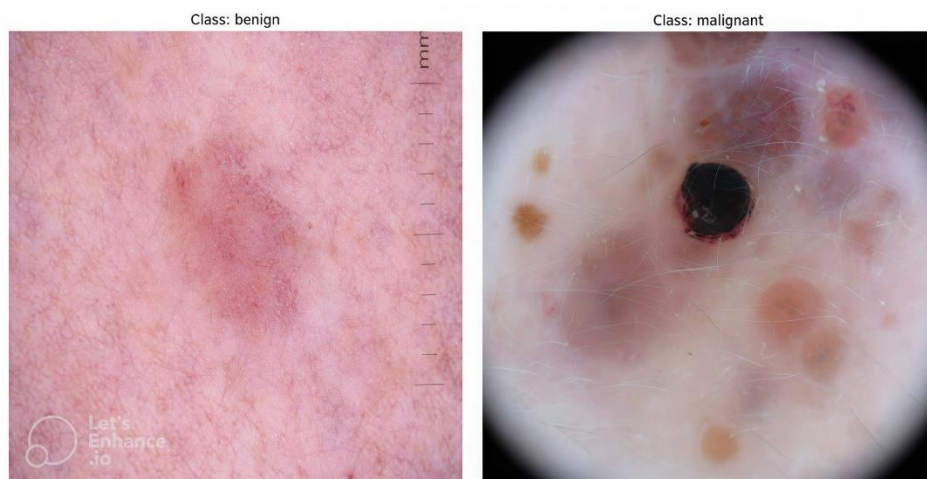
Dalam penelitian ini menggunakan beberapa metode yang memiliki sejumlah tahapan dimulai dari pengumpulan *dataset*, *image preprocessing*, *splitting data*, augmentasi data, pembangunan model, dan evaluasi. Dari keseluruhan tahapan tersebut harus dilakukan secara berurutan, keseluruhan alur dapat dilihat pada Gambar 1.



Gambar 1. Alur tahapan penelitian

A. Pengumpulan Dataset

Tahapan awal dalam penelitian ini yaitu melakukan pengambilan *dataset* yang berasal dari *website* yaitu *Kaggle* dengan nama *dataset Melanoma Skin Cancer Dataset of 10000 Images* [13]. *Dataset* ini berisikan 10.605 data citra yang mencakup berbagai kondisi kulit, didalam *dataset* dibagi Kembali menjadi 2 kategori utama yaitu *benign* (jinak) dan *maglinant* (ganas), dengan pembagian isi dari kategori tersebut adalah 5.105 data citra untuk kategori *malignant* dan 5.500 data citra untuk kategori *benign*. Untuk contoh data citra dapat dilihat pada Gambar 2.



Gambar 2. Data citra *Benign* dan *Malignant*

B. Image Preprocessing

Tahap *Image Preprocessing* (Pra-pemrosesan Citra) diimplementasikan sebagai langkah krusial untuk memastikan standarisasi *dataset* dan membersihkan data dari anomali yang berpotensi mengkompromikan proses pelatihan model. Implementasi ini mengikuti alur sistematis yang mencakup: (1) *Image ReSize*, (2) *Remove Duplicate*, (3) *Hair Removal*, (4) *Contrast Enhancement*, (5) Konversi citra ke format *tensor*, dan (6) *Normalization*.

1) *Image ReSize*: Proses ini dilakukan untuk menyeragamkan dimensi dari semua data citra yang telah diperoleh, dengan tujuan agar model dapat menetima input dengan ukuran yang konsisten dan proses komputasi menjadi lebih efisien. Tahapan ini mengubah ukuran gambar menjadi resolusi 112x112 piksel.

2) *Remove Duplicate*: File duplikat sering kali menjadi sebuah gangguan yang didapati dalam pelatihan model, sehingga penghapusan file duplikat dilakukan untuk mendapatkan *dataset* yang bersih dan sempurna, dengan melakukan pembacaan pada hal unik pada citra, apabila terdapat nilai yang sama maka akan dianggap sama dan dilakukan penghapusan salah satu dari file yang memiliki nilai yang sama.

3) *Hair Removal*: Prosedur penghapusan artefak rambut (*Hair Removal*) diimplementasikan untuk mengeliminasi *noise* dan meningkatkan kualitas *dataset*. Proses ini diawali dengan konversi citra dari *RGB* ke *Grayscale* untuk menyederhanakan operasi. Selanjutnya, teknik *Morphological Closing* (kombinasi *Dilation* dan *Erosion*) diterapkan: *Dilation* melebarkan piksel kulit yang lebih terang untuk menutupi garis rambut yang gelap, diikuti oleh *Erosion* yang mengembalikan ukuran objek

namun tetap mempertahankan area rambut yang telah tertutup. Garis rambut kemudian diisolasi melalui operasi pengurangan citra (*image subtraction*) antara citra hasil *closing* dengan citra *Grayscale* asli. Citra hasil isolasi ini selanjutnya diproses melalui Binarisasi (dengan *threshold* 25) untuk menciptakan *hair mask biner* (0 untuk latar belakang, 1 untuk rambut). Masker ini disempurnakan menggunakan *Dilation* tambahan untuk menutup celah kecil. Terakhir, teknik *Inpainting* digunakan untuk merekonstruksi area yang ditandai oleh *hair mask* dengan menganalisis dan mereplikasi tekstur serta warna dari piksel kulit di sekitarnya, sehingga menghasilkan citra yang bersih dan alami.

4) *Contrast Enhancement*: Implementasi teknis dari *Contrast Enhancement* (Peningkatan Kontras) diawali dengan mengonversi citra masukan dari ruang warna *BGR* ke *LAB*, Setelah dikonversi, citra segera dipisahkan menjadi *L-channel*, *A-channel*, dan *B-channel*, di mana fokus manipulasi hanya ditujukan pada *L-channel*. Sebuah objek *Contrast Limited Adaptive Histogram Equalization (CLAHE)* kemudian diinisiasi dengan parameter *clipLimit* diatur ke 2.0 dan *tileGridSize* ke (8, 8), yang bertujuan melakukan ekuilisasi histogram secara adaptif pada grid lokal sekaligus membatasi amplifikasi *noise*, Algoritma *CLAHE* ini lantas diterapkan secara eksklusif pada *L-channel*, dan *L-channel* hasil modifikasi tersebut digabungkan kembali dengan *A-channel* dan *B-channel* orisinal yang tidak diubah. Sebagai langkah final, citra *LAB* gabungan dikonversi kembali ke ruang warna *BGR*, sehingga menghasilkan citra akhir dengan kontras yang telah ditingkatkan secara signifikan namun tetap mempertahankan fidelitas warna aslinya.

5) *Konversi citra ke format tensor*: Proses ini digunakan untuk mempermudah langkah berikutnya dalam menjalankan proses *trainingnya*, dikarenakan penggunaan *PyTorch* dalam model yang dibangun perubahan citra menjadi *tensor* diperlukan, dengan menggunakan *library transform* yang nantinya akan mengubah format citra menjadi *tensor* dengan perbandingan 0 hingga 1.

6) *Normalization*: Dalam proses ini citra yang telah dirubah menjadi *tensor* dilakukan normalisasi dengan nilai *mean* [0,485, 0,456, 0,406] dan deviasi [0,229, 0,224, 0,225], penggunaan nilai tersebut sebagai standar nilai yang dipakai dalam penggunaan *dataset* besar seperti *ImageNet* sehingga penggunaan proses normalisasi dengan nilai yang disebutkan sebagai syarat untuk model dapat berjalan secara efektif.

C. *Splitting data*

Pada proses *splitting data*, *dataset* awal yang telah dikumpulkan akan diacak (*shuffle*) untuk menghindari bias data, selanjutnya data dibagi menjadi 80% untuk data *train*, 10% untuk data *validation*, dan 10% untuk data *test*. Dengan rasio ini, proporsi data cukup besar untuk melatih model, namun juga cukup data untuk validasi dan pengujian guna mencegah *overfitting* serta menjaga keakuratan evaluasi performa model.

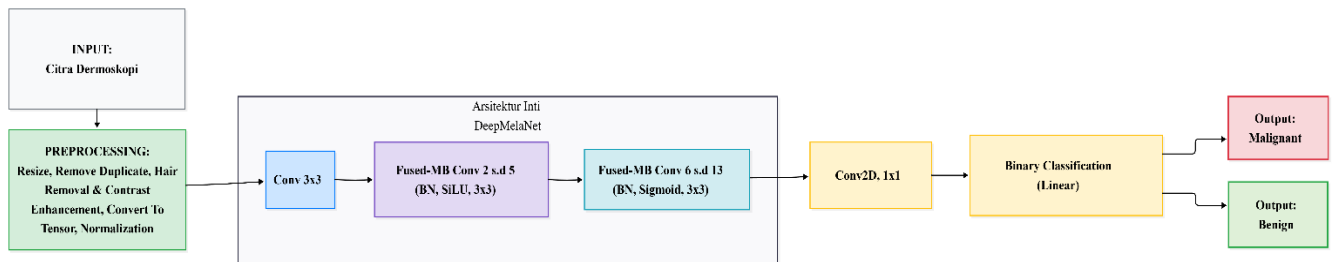
D. *Augmentasi Data*

Augmentasi dilakukan untuk meningkatkan jumlah dan variasi data latih, dengan memodifikasi data asli melalui beberapa *transformasi*. Teknik ini sangat penting untuk meningkatkan generalisasi model dan mengurangi risiko *overfitting*, terutama jika jumlah data asli terbatas.

Setelah data di proses pada *splitting data*, augmentasi dilakukan pada data latih menggunakan *library torchvision*. Teknik yang digunakan pada proses ini mencakup *Random rotation* dan *Random Flipping* dengan ketentuan, untuk *Random rotation* dilakukan sebesar 20° mengikuti arah jarum jam dan untuk *Random Flipping* menggunakan probabilitas 30% untuk keseluruhan proses yang ada pada data latih.

E. *Pembangunan Model*

Arsitektur *DeepMelaNet* dikembangkan dari arsitektur *EfficientNet*, dengan beberapa modifikasi untuk mengoptimalkan biaya akses memori. Salah satu modifikasi yang utama adalah penggunaan rasio ekspansi yang lebih kecil untuk *MBConv* (*Mobile Inverted Bottleneck Convolution*), rasio ekspansi yang lebih kecil cenderung lebih disukai karena menghasilkan akses memori yang lebih rendah, selain itu *DeepMelaNet* mengutamakan ukuran *kernel* yang lebih kecil yaitu (3x3). Namun, untuk mengompresi penurunan bidang penerimaan yang disebabkan oleh ukuran *kernel* yang lebih kecil, lapisan tambahan di tambahkan ke dalam jaringan. Untuk pelatihan model diinisiasi pada platform *website Kaggle*, dengan menggunakan *kernel GPU T4 Kaggle* yang didukung oleh *GPU Tels T4 Nvidia*. Landasan komputasi untuk penelitian ini dibangun menggunakan *PyTorch* versi 2.2.1. untuk bentuk arsitektur pada *DeepMelaNet* akan di tampilkan pada Gambar 3.



Gambar 3. Arsitektur DeepMelaNet

Pada arsitektur *DeepMelaNet* terdapat beberapa blok yang memiliki lapisan dan nilai yang berbeda, secara detail isi tersebut akan di jelaskan sebagai berikut:

- 1) *Conv 3x3*: blok ini bekerja sebagai bagian awal untuk menerima input dari data citra, dengan menggunakan *layer* konvolusi dengan ukuran 3x3 sebagai penerima awal dengan *output 32 channel*.
- 2) *Fused-MB Conv 2*: blok ini memiliki 2 lapisan, dengan lapisan awal menggunakan *layer* konvolusi dengan *filter 3x3*, *batch* normalisasi serta *SiLU* menghasilkan *output 32 channel* dengan dimensi *spatial* sebesar 128 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 1x1* dan *batch* normalisasi menghasilkan *output 128 channel* serta dimensi *spatial 64 channel*.
- 3) *Fused-MB Conv 3*: sama seperti blok sebelumnya yang berisikan 2 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 3x3*, *batch* normalisasi serta *SiLU* menghasilkan *output 64 channel* dengan dimensi *spatial* sebesar 256 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 3x3* dan *batch* normalisasi menghasilkan *output 256 channel* serta dimensi *spatial 64 channel*.
- 4) *Fused-MB Conv 4*: blok berikut berisi 2 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 3x3*, *batch* normalisasi serta *SiLU* menghasilkan *output 64 channel* dengan dimensi *spatial* sebesar 256 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 1x1* dan *batch* normalisasi menghasilkan *output 256 channel* serta dimensi *spatial 96 channel*.
- 5) *Fused-MB Conv 5*: blok berikut berisi 2 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 3x3*, *batch* normalisasi serta *SiLU* menghasilkan *output 96 channel* dengan dimensi *spatial* sebesar 384 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 1x1* dan *batch* normalisasi menghasilkan *output 384 channel* serta dimensi *spatial 96 channel*.
- 6) *Fused-MB Conv 6*: berbeda dengan blok sebelumnya, blok ini berisikan 3 lapisan dengan lapisan awal menggunakan *layer* konvolusi dengan *filter 1x1*, *batch* normalisasi serta *SiLU* menghasilkan *output 96 channel* dengan dimensi *spatial* sebesar 384 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 3x3* dan *batch* normalisasi menghasilkan *output 768x768*. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 48 dan kemudian di perluas kembali menjadi 768. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter 1x1* dengan *output 768* dan dimensi *spatial* sebesar 192 *channel*.
- 7) *Fused-MB Conv 7*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 1x1*, *batch* normalisasi serta *SiLU* menghasilkan *output 192 channel* dengan dimensi *spatial* sebesar 768 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 3x3* dan *batch* normalisasi menghasilkan *output 768x768*. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 48 dan kemudian diperluas kembali menjadi 768. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter 1x1* dengan *output 768* dan dimensi *spatial* sebesar 192 *channel*.
- 8) *Fused-MB Conv 8*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 1x1*, *batch* normalisasi serta *SiLU* menghasilkan *output 192 channel* dengan dimensi *spatial* sebesar 1152 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 3x3* dan *batch* normalisasi menghasilkan *output 1152x1152*. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 48 dan kemudian diperluas kembali menjadi 1152. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter 1x1* dengan *output 1152* dan dimensi *spatial* sebesar 224 *channel*.
- 9) *Fused-MB Conv 9*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter 1x1*, *batch* normalisasi serta *SiLU* menghasilkan *output 224 channel* dengan dimensi *spatial* sebesar 1344 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter 3x3* dan *batch* normalisasi menghasilkan *output 1344x1344*. Berhubungan dengan

ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 56 dan kemudian diperluas kembali menjadi 1344. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter* 1x1 dengan *output* 1344 dan dimensi *spatial* sebesar 224 *channel*.

10) *Fused-MB Conv 10*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter* 1x1, *batch* normalisasi serta *SiLU* menghasilkan *output* 224 *channel* dengan dimensi *spatial* sebesar 1344 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter* 3x3 dan *batch* normalisasi menghasilkan *output* 1344x1344. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 56 dan kemudian diperluas kembali menjadi 1344. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter* 1x1 dengan *output* 1344 dan dimensi *spatial* sebesar 384 *channel*.

11) *Fused-MB Conv 11*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter* 1x1, *batch* normalisasi serta *SiLU* menghasilkan *output* 384 *channel* dengan dimensi *spatial* sebesar 2304 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter* 3x3 dan *batch* normalisasi menghasilkan *output* 2304x2304. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 96 dan kemudian diperluas kembali menjadi 2304. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter* 1x1 dengan *output* 2304 dan dimensi *spatial* sebesar 384 *channel*.

12) *Fused-MB Conv 12*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter* 1x1, *batch* normalisasi serta *SiLU* menghasilkan *output* 384 *channel* dengan dimensi *spatial* sebesar 2304 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter* 3x3 dan *batch* normalisasi menghasilkan *output* 2304x2304. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 96 dan kemudian diperluas kembali menjadi 2304. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter* 1x1 dengan *output* 2304 dan dimensi *spatial* sebesar 640 *channel*.

13) *Fused-MB Conv 13*: blok berikut berisi 3 lapisan, lapisan awal menggunakan *layer* konvolusi dengan *filter* 1x1, *batch* normalisasi serta *SiLU* menghasilkan *output* 640 *channel* dengan dimensi *spatial* sebesar 3840 *channel*, lapisan kedua juga menggunakan *layer* konvolusi dengan *filter* 3x3 dan *batch* normalisasi menghasilkan *output* 3840x3840. Berhubungan dengan ini, lapisan *Squeeze Excitation* digunakan untuk mengurangi *channel* menjadi 160 dan kemudian diperluas kembali menjadi 3840. Akhirnya dilanjutkan dengan operasi konvolusi menggunakan *filter* 1x1 dengan *output* 3840 dan dimensi *spatial* sebesar 640 *channel*.

14) *Fused-MB Conv 14*: berbeda dengan blok lainnya, blok berikut merupakan *layer* klasifikasi terakhir yang dimodifikasi untuk memiliki *output* jaringan tunggal. Modifikasi berikut bertujuan untuk mengadaptasi model untuk klasifikasi biner, di mana *output* mewakili probabilitas bahwa suatu citra termasuk dalam kelas *malignant* dan *benign*.

Dalam tahap melakukan *training* terhadap model yang telah dibangun, perlunya beberapa skenario untuk dilakukan demi menemukan perbandingan mana yang cocok dengan tingkat optimasi terhadap akurasi yang dihasilkan. proses ini memerlukan konfigurasi awal yang cermat sebelum *training* dan *validation* dimulai. Fokus utamanya adalah pada empat komponen strategis berikut:

1) *Transformasi Data*: Data *training* dan *validation* melalui augmentasi: rotasi acak (-20 hingga 20 derajat), pencerminan horizontal/vertikal (probabilitas 0,3), dan pengubahan ukuran menjadi 112x122 (menggunakan *antialiasing* dan *center crop*). Citra (format *PIL*) diubah menjadi *tensor PyTorch*, kemudian dinormalisasi menggunakan *mean* [0,485, 0,456, 0,406] dan standar deviasi [0,229, 0,224, 0,225] *ImageNet*.

2) *Epoch*: Proses *training* diatur sebanyak 30 *epoch* untuk menyeimbangkan pelatihan dan efisiensi komputasi. Setelah setiap *epoch*, akurasi dan *loss* dicatat. Model kemudian beralih ke mode evaluasi (gradien dinonaktifkan) untuk menghemat memori dan mempercepat proses.

3) *Pemilihan Learning rate*: *Learning rate (LR)* awal ditetapkan sebesar 0,001. Proses ini menggunakan *scheduler* "ReduceLROnPlateau", yang akan menurunkan *LR* secara dinamis jika *validation loss* tidak menunjukkan perbaikan. Tujuannya adalah untuk mencegah *overfitting* dan meningkatkan performa.

4) *Training dan Validation Loop*: Sebuah algoritma disiapkan untuk menjalankan proses *training* dan validasi di setiap *epoch*. Selama *training*, model memproses *batch* data dan memperbarui parameter. Setelah selesai, model dievaluasi pada set validasi. *Validation loss* yang dihasilkan dari proses ini digunakan untuk mengatur *scheduler learning rate*. Keseluruhan proses akan diilustrasikan pada *pseudocode* berikut:

```
1) Input: model, trainLoader, valLoader, optimizer, criterion, scheduler, epochs
2) Output: trainLosses, trainAccs, valLosses, valAccs, valPrecisions, valRecalls, valF1s
3) Initialize scaler for mixed-Precision training
```

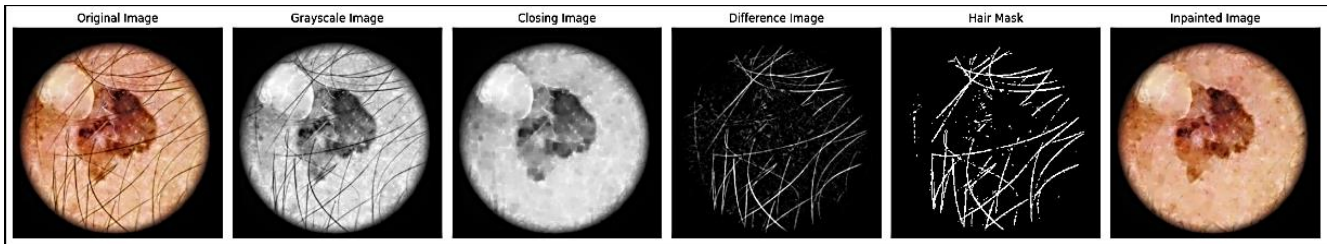
```
4) Initialize metric lists: trainLosses, trainAccs, valLosses, etc.
5) for epoch = 1 to epochs do
6)   Set model to training mode
7)   Initialize running Loss  $\leftarrow 0$ , correct_train  $\leftarrow 0$ 
8)   for each input, labels in trainLoader do
9)     Move inputs, labels to Device
10)    Zero gradients: optimizer.zero_grad()
11)    With autocast enabled:
12)      Outputs  $\leftarrow$  model(inputs)
13)      loss  $\leftarrow$  criterion(outputs, labels)
14)      Backpropagate and update model:
15)        scaler.scale(loss), backward(),
16)        scaler.step(optimizer)
17)      Update running loss and correct predictions
18)    end for
19)    Calculate and store trainLoss and trainAcc
20)    Print training metrics
21)    Set model to evaluation mode
22)    Initialize validation metrics
23)    With gradient disabled:
24)      for each input, Labels in valLoader
25)        Move inputs, Labels to device
26)        outputs  $\leftarrow$  model(inputs), compute loss
27)        Update validation loss and store predictions
28)      end for
29)      Calculate and store validation metrics: avgLoss, accuracy, Precision, etc.
30)      Print validation metrics
31)      Update scheduler based on validation Loss
32)    end for
```

Proses *training* diawali dengan fase inisialisasi yang cermat, di mana semua komponen prasyarat didefinisikan, termasuk arsitektur model, *dataloader* untuk *training* dan validasi, fungsi *loss (criterion)*, *optimizer*, *scheduler*. Selain itu, *scaler* untuk pelatihan *mixed-Precision* juga disiapkan guna meningkatkan efisiensi komputasi. Inti dari proses ini adalah perulangan (*loop*) utama yang dieksekusi berdasarkan jumlah *epoch* yang telah ditentukan, di mana setiap *epoch* mencakup dua fase utama. Pertama, fase pelatihan, yang mengonfigurasi model ke mode *training*, di sini, untuk setiap *batch* data, gradien di-nolkan, *forward pass* dijalankan untuk prediksi, loss dihitung, *backpropagation* dieksekusi (diskalakan untuk stabilitas), dan *optimizer* memperbarui bobot model.

Kedua, fase validasi, di mana model dialihkan ke mode evaluasi dan perhitungan gradien dinonaktifkan untuk mengurangi beban komputasi. Pada tahap ini, model dievaluasi pada data validasi, dan serangkaian metrik kinerja komprehensif, seperti *loss*, akurasi, presisi, *Recall*, dan *F1-Score* dihitung. Sebagai penutup setiap *epoch*, *scheduler* menyesuaikan *learning rate* secara dinamis berdasarkan *validation loss*, dan siklus ini dilakukan berulang untuk menyempurnakan konvergensi model.

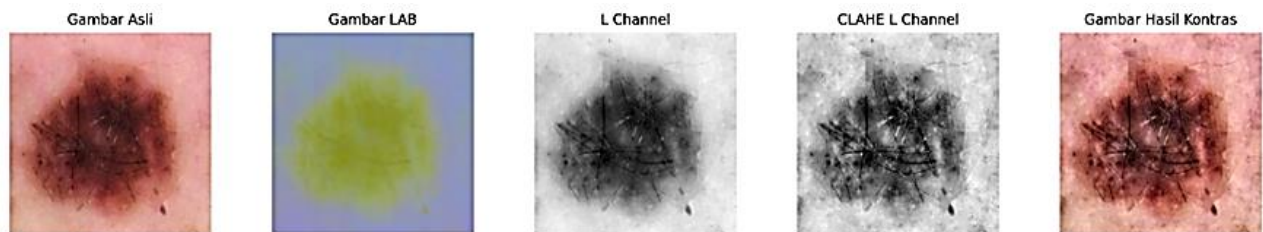
III. HASIL DAN PEMBAHASAN

Penelitian ini diawali dengan melakukan pengumpulan *dataset*, dengan keseluruhan isi *dataset* berjumlah 10.605 dan terbagi menjadi 5.105 untuk data citra *malignant* dan 5.500 untuk data citra *benign*, dilanjutkan dengan *image preprocessing*, langkah pertama dengan mengubah keseluruhan data citra menjadi 112x112 piksel, setelah mengubah ukuran tersebut masuk ke dalam tahap penghapusan *file* duplikat untuk melihat apabila terdapat data citra yang memiliki duplikat akan langsung dihapus, tahap berikutnya masuk ke dalam tahap penghapusan garis rambut, tahap ini dimulai dengan mengubah format citra dari *RGB* menjadi *Grayscale* untuk menyederhanakan citra. Selanjutnya, dilakukan proses *morphological closing* yang menggabungkan operasi dilasi dan erosi. Dilasi berfungsi melebarkan piksel terang (kulit) untuk menutupi piksel gelap (rambut), diikuti erosi yang mengembalikan ukuran piksel ke semula namun tanpa memunculkan kembali piksel rambut yang sudah tertimpa. Setelah itu, rambut diisolasi dengan mengurangi citra hasil *morphological closing* dengan citra *Grayscale* asli, sehingga garis rambut terdeteksi sebagai warna hitam. Proses dilanjutkan dengan binarisasi menggunakan nilai ambang batas "25" untuk menghasilkan "*hair mask*". *Hair mask* ini kemudian dipertebal menggunakan operasi dilasi agar celah kecil tertutup sempurna. Terakhir, dilakukan proses *Inpainting* (melukis ulang), di mana area yang ditandai putih pada *hair mask* dianalisis dan diisi ulang dengan piksel baru yang serasi dengan tekstur kulit di sekitarnya. Untuk hasilnya dapat dilihat pada Gambar 4.



Gambar 4. Proses Penghapusan Garis Rambut

Setelah dilakukannya tahap penghapusan garis rambut yang di tampilkan pada Gambar 4. Dilanjutkan dengan tahap *preprocessing* berikutnya yaitu peningkatan kontras citra, Tahap peningkatan kontras citra dilakukan untuk memperbaiki kontras yang berkurang akibat proses penghapusan garis rambut sebelumnya. Proses ini diawali dengan mengonversi format citra dari *BGR* (Biru, Hijau, Merah) menjadi format warna *LAB*, yang memisahkan *Lightness* (kecerahan) dari kanal warna. Pemisahan ini memungkinkan manipulasi kontras dan ketajaman tanpa merusak warna asli citra. Selanjutnya, dilakukan penajaman citra untuk menonjolkan detail, serta diterapkan *CLAHE* (*Contrast Limited Adaptive Histogram Equalization*). Penggunaan *CLAHE* bertujuan meningkatkan kontras secara signifikan namun tetap terlihat alami karena penyesuaiannya bersifat adaptif di seluruh bagian citra. Setelah tahap peningkatan kecerahan dan detail selesai, citra yang masih dalam format *LAB* dikembalikan ke format *BGR* agar dapat disimpan dan digunakan dalam format standar. Hasil dari proses ini dapat dilihat pada Gambar 5.



Gambar 5. Proses Peningkatan Kontras Citra.

Tahap konversi citra menjadi *tensor* dilakukan karena model *deep learning* dalam penelitian ini dibangun menggunakan PyTorch, yang memerlukan data masukan dalam format *tensor*. Proses ini bertujuan untuk mempermudah jalannya proses *training*. Dalam praktiknya, digunakan *library transform* dari PyTorch yang akan mengubah format citra, seperti PIL, menjadi *tensor* dengan rentang nilai pikselnya diskalakan antara 0 hingga 1. Setelah citra diubah menjadi *tensor*, proses dilanjutkan dengan normalisasi. Tahap ini menstandarisasi nilai piksel *tensor* menggunakan nilai *mean* [0,485, 0,456, 0,406] dan standar deviasi [0,229, 0,224, 0,225]. Nilai-nilai ini adalah standar yang umum digunakan pada *dataset* besar seperti *ImageNet*. Penggunaan normalisasi dengan nilai standar ini dianggap sebagai syarat agar model dapat berjalan secara efektif.

Setelah data citra menyelesaikan seluruh tahap pra-pemrosesan, langkah selanjutnya adalah pembagian *dataset* (*splitting data*)¹. Proses ini krusial untuk mengevaluasi performa model secara objektif. *Dataset* dibagi menjadi tiga bagian dengan proporsi 80% untuk data pelatihan (*train*), 10% untuk data validasi (*validation*), dan 10% untuk data pengujian (*test*). Berdasarkan total 10.605 data citra, pembagian ini menghasilkan 9.605 data latih, 1.000 data validasi, dan 1.000 data uji. Untuk menghindari bias, *dataset* diacak (*shuffle*) terlebih dahulu sebelum dibagi.

Selanjutnya, hanya pada data pelatihan (9.605 citra) diterapkan teknik augmentasi data. Augmentasi adalah teknik untuk meningkatkan jumlah dan variasi data latih dengan memodifikasinya. Tujuannya adalah untuk memberikan lebih banyak variasi data yang akan dipelajari oleh model, yang sangat penting untuk meningkatkan kemampuan generalisasi model dan mengurangi risiko overfitting. Dalam penelitian ini, teknik augmentasi yang digunakan meliputi Random Rotation (rotasi acak) sebesar 20 derajat dan *Random Flipping* (pencerminan acak) dengan probabilitas 30%.

Setelah *dataset* selesai dipersiapkan melalui tahap pra-pemrosesan dan pembagian data, fokus beralih pada persiapan dan pelatihan model arsitektur "*DeepMelaNet*". Sebelum memulai proses *training*, sebuah desain eksperimen dirancang untuk mengevaluasi performa model secara komprehensif. Eksperimen ini dibagi menjadi dua skenario pelatihan utama. Salah satunya perbedaan antara kedua skenario ini adalah nilai *learning rate* (*LR*).

Di dalam setiap skenario pelatihan, dilaksanakan sejumlah pengujian untuk mengevaluasi performa masing-masing metode pra-pemrosesan yang telah dikembangkan. Pengujian ini dirancang dengan tujuan untuk memberikan gambaran yang komprehensif mengenai efektivitas setiap metode yang diterapkan dan untuk mengidentifikasi konfigurasi yang paling optimal. Secara spesifik, penelitian ini membandingkan empat variasi metode, yakni: pertama, penerapan Penghapusan Garis Rambut secara tunggal. Kedua, penerapan Peningkatan Kontras Citra secara Tunggal. Ketiga, Metode Gabungan yang

mengkombinasikan Penghapusan Garis Rambut diikuti Peningkatan Kontras Citra; dan keempat, Metode Gabungan Terbalik. Metode terbalik ini, yang mendahulukan Peningkatan Kontras Citra sebelum Penghapusan Garis Rambut, diimplementasikan dengan hipotesis bahwa mendapatkan detail yang lebih mendalam dari data citra terlebih dahulu dapat berkontribusi pada proses penghapusan gangguan (*noise*) yang lebih efektif.

Untuk menyajikan perbandingan komprehensif dan memfasilitasi analisis hasil, performa model dari kedua skenario pelatihan dengan skenario pertama (*LR* 0,001) dan skenario kedua (*LR* 0,0001) serta keempat metode pra-pemrosesan telah dikonsolidasikan. Perbandingan rinci metrik evaluasi utama, yang mencakup akurasi, presisi, *Recall*, dan *F1-Score*, dari seluruh konfigurasi pada parameter pengujian tersebut disajikan secara lengkap pada Tabel 1.

TABEL 1
PARAMETER SKENARIO PERTAMA

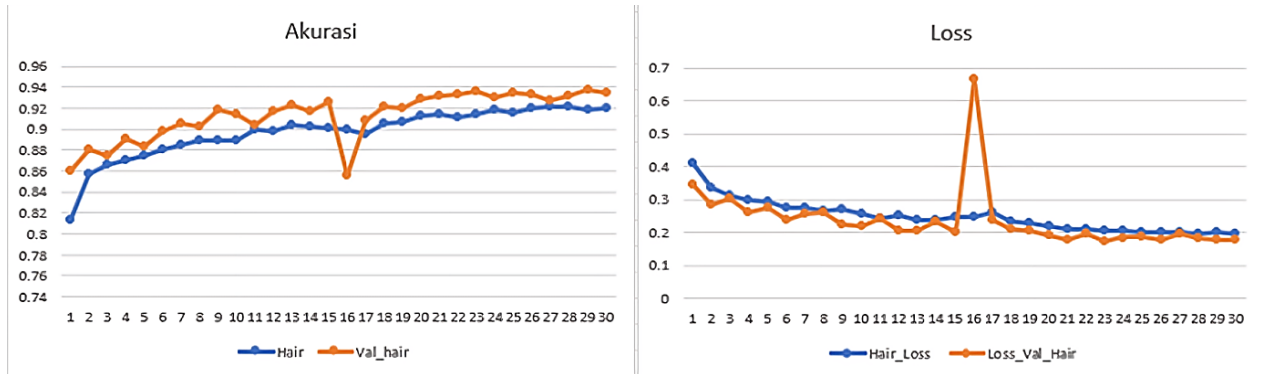
Parameter	Nilai Parameter
Epoch	30
Batch Size	32
Learning rate	0,001
Optimizer	Adam
Scheduler	ReduceLROnPlateau
Loss Function	Binary Cross-Entropy with Logit Loss

Pengujian Skenario pertama, dengan konfigurasi parameter yang dirinci pada Tabel 1, memperlihatkan variabilitas hasil yang signifikan. Perbedaan ini, yang utamanya tercermin pada beragamnya nilai akurasi validasi, menjadi krusial untuk menganalisis dan membandingkan performa optimasi antar model. Seluruh data hasil eksperimen Skenario pertama telah dirangkum pada Tabel 2.

TABEL 2
HASIL KLASIFIKASI SKENARIO PERTAMA

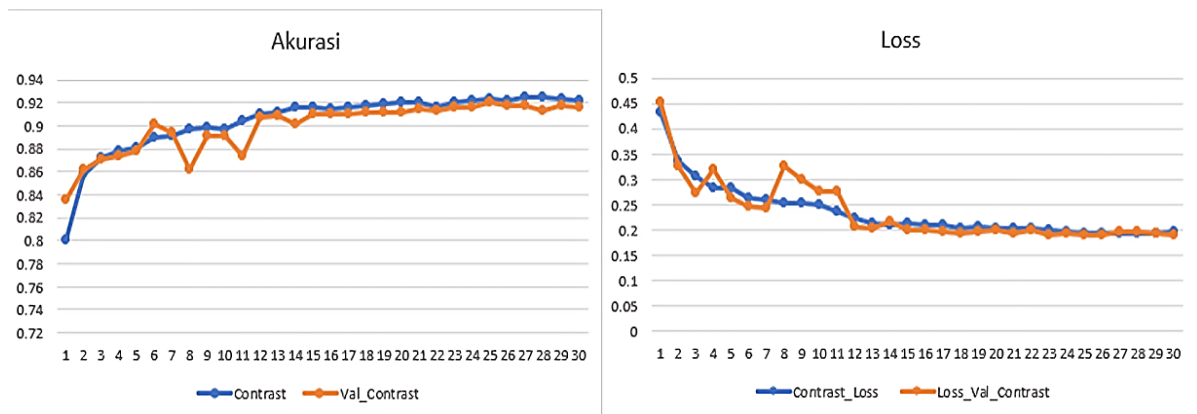
Methods	Result				
	Accuracy	Precision	Recall	F1-Score	Average Time per-Epoch
Hair Removal	93,7%	96,9%	88,4%	92,4%	336,6s
Contrast Enhancement	92%	95,9%	90,4%	93,1%	319,5s
Hair Removal + Contrast Enhancement	92,3%	95,2%	87,2%	97,7%	329,2s
Contrast Enhancement + Hair Removal	92,5%	94,7%	89,6%	98,1%	356,1s

Skenario pelatihan pertama dijalankan dengan parameter utama *learning rate (LR)* sebesar 0,001, jumlah *Epoch* 30, dan *Batch Size* 32. Secara keseluruhan, hasil dari keempat metode pra-pemrosesan dalam skenario ini menunjukkan masalah pada stabilitas pelatihan dan performa yang beragam. Skenario pelatihan pertama, yang dijalankan menghasilkan tingkat akurasi yang bervariasi di keempat metode pengujian.



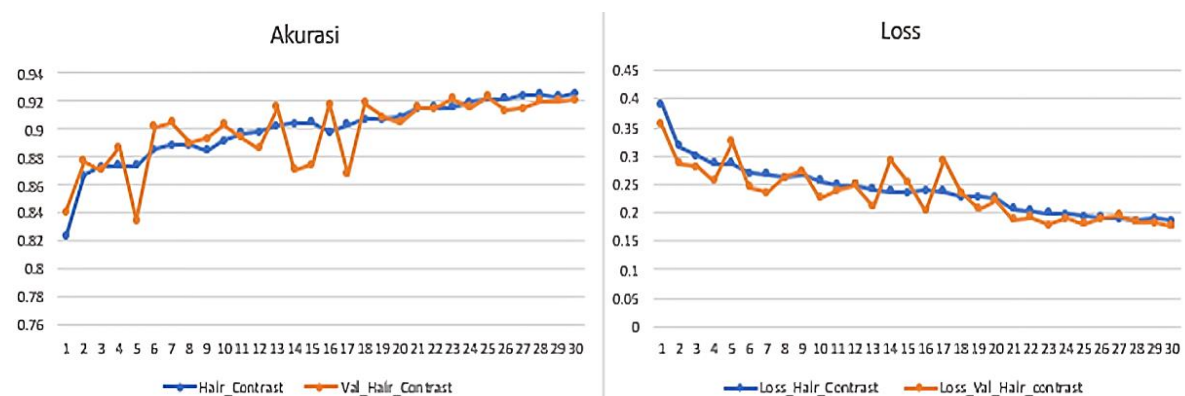
Gambar 6. Hasil DeepMelaNet menggunakan Hair Removal

Metode dengan *Hair Removal* pada Gambar 6 menunjukkan performa yang cukup baik dan relatif stabil, meskipun mengalami lonjakan pada *epoch* ke-15. Metode ini berhasil mencapai akurasi validasi tertinggi dalam skenario ini, yaitu 93,7% pada *epoch* ke-29. Dengan performa yang didapatkan pada *Precision* sebesar 96,9%, *Recall* 88,4%, dan *F1-Score* 92,4%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 336,6 detik.



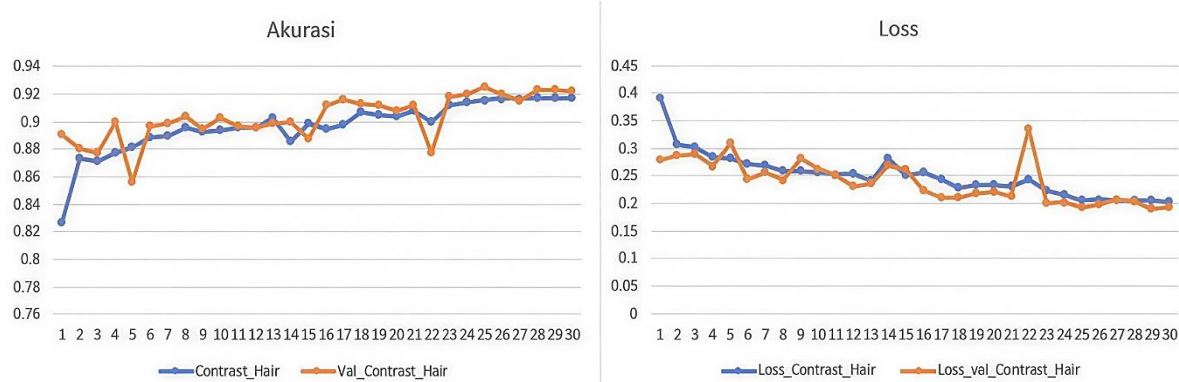
Gambar 7. Hasil DeepMelaNet dengan Contrast Enhancement

DeepMelaNet dengan *Contrast Enhancement* pada Gambar 7 menunjukkan hasil yang baik, namun kurva pelatihannya menampilkan fluktuasi yang cukup terlihat pada *epoch-epoch* awal sebelum akhirnya stabil setelah *epoch* ke-15. Akurasi validasi tertinggi yang dicapai adalah 92%. Dengan performa yang didapatkan pada *Precision* sebesar 95,9%, *Recall* 90,4%, dan *F1-Score* 93,1%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 319,5 detik.



Gambar 8. Hasil DeepMelaNet dengan metode ke-3

DeepMelaNet dengan Metode Gabungan (*Hair Removal* dan *Contrast Enhancement*) pada Gambar 8 menunjukkan performa yang "unik", ditandai dengan fluktuasi yang beragam dan signifikan pada kurva validasi hingga *epoch* ke-21. Performa ini dinilai sangat tidak efisien dan berisiko, yang diakibatkan oleh nilai *learning rate* awal yang terlalu tinggi. Model ini baru stabil setelah scheduler menurunkan *LR*. Akurasi validasi tertinggi mencapai 92,3%, Dengan perfoma yang didapatkan pada *Precision* sebesar 95,2%, *Recall* 87,2%, dan *F1-Score* 91%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 329,2 detik.



Gambar 9. Hasil *DeepMelaNet* metode ke-4

DeepMelaNet dengan Metode Gabungan Terbalik (Peningkatan Kontras Citra dan *Hair Removal*) pada Gambar 9 menunjukkan tingkat ketidakstabilan yang tinggi dengan pola kurva seperti gigi gergaji (*sawtooth pattern*), sehingga dianggap tidak dapat diandalkan dengan parameter ini. Meskipun akurasi validasinya mencapai 92,5%, Dengan perfoma yang didapatkan pada *Precision* sebesar 94,7%, *Recall* 89,6%, dan *F1-Score* 92%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 356,1 detik.

Skenario pelatihan kedua ditunjukan pada Tabel 3 dijalankan dengan perubahan utama yaitu pada *learning rate* (*LR*) yang diturunkan menjadi 0,0001, sementara jumlah *Epoch* (30) dan *Batch Size* (32) tetap sama. Penyesuaian *LR* ini terbukti krusial, karena secara keseluruhan, hasil dari keempat metode dalam skenario ini menunjukkan konvergensi yang jauh lebih baik, kurva pelatihan yang lebih stabil, dan performa *False Negative* (*FN*) yang menurun drastis dibandingkan skenario pertama.

TABEL 3
PARAMETER SKENARIO KEDUA

	Nilai
Epoch	30
Batch Size	32
Learning rate	0,0001
Optimizer	Adam
Scheduler	ReduceLROnPlateau
Loss Function	Binary Cross-Entropy with Logit Loss

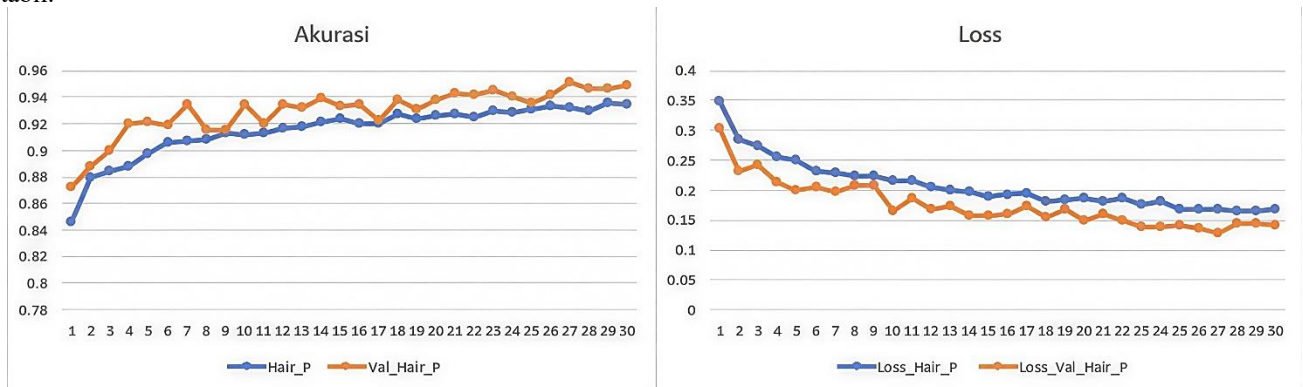
Pengujian Skenario 2 dirancang identik dengan skenario sebelumnya, baik dalam prosedur maupun penggunaan empat metode yang ada. Fokus skenario ini adalah modifikasi parameter *learning rate* untuk mengevaluasi dampaknya terhadap performa model. Rincian hasil perbandingan dari keseluruhan metode pada skenario ini disajikan pada Tabel 4.

TABEL 4
HASIL KLASIFIKASI SKENARIO KEDUA

Methods	Result				
	Accuracy	Precision	Recall	F1-Score	Average Time per-Epoch
Hair Removal	94,6%	97,4%	91,6%	94,4%	295,6s

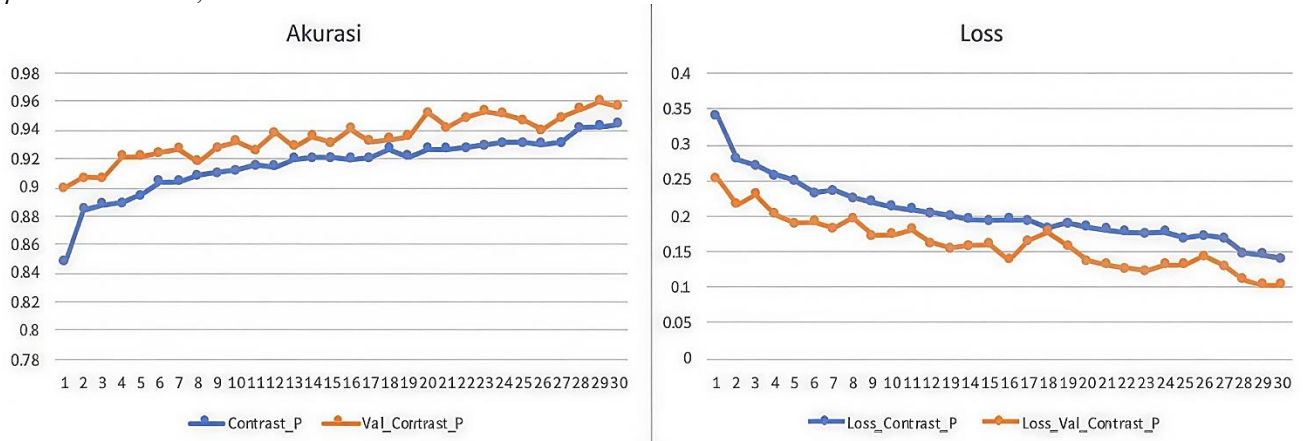
Methods	Result				
	Accuracy	Precision	Recall	F1-Score	Average Time per-Epoch
Contrast Enhancement	95,7%	97,9%	93,4%	95,6%	353,8s
Hair Removal + Contrast Enhancement	94,9%	96,8%	92,8%	94,7%	256,6s
Contrast Enhancement + Hair Removal	96,6%	98,7%	94,4%	96,5%	317,3s

Skenario pelatihan kedua dijalankan dengan perubahan utama pada *learning rate* (*LR*) yang diturunkan menjadi 0,0001, sementara jumlah *Epoch* (30) dan *Batch Size* (32) tetap sama. Penyesuaian *LR* ini terbukti krusial, karena secara keseluruhan, hasil dari keempat metode dalam skenario ini menunjukkan konvergensi yang jauh lebih baik, kurva pelatihan yang lebih stabil.



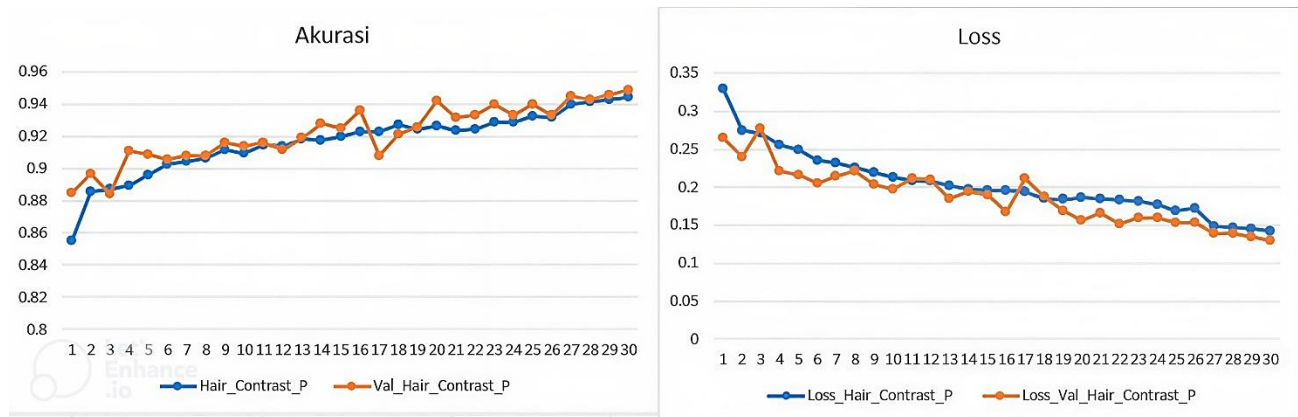
Gambar 10. Hasil Metode Pertama Skenario Kedua

Gambar 10 menggunakan *DeepMelaNet* dengan Penghapusan Garis Rambut menunjukkan peningkatan performa yang signifikan berkat perubahan *LR*. Kurva pelatihan menjadi jauh lebih stabil, tanpa fluktuasi berlebihan, yang mengindikasikan konvergensi yang lebih baik. Model ini berhasil mencapai akurasi validasi yang tinggi, yaitu 95,1%. Dengan performa yang didapatkan pada *Precision* sebesar 97,4%, *Recall* 91,6%, dan *F1-Score* 94,4%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 295,6 detik.



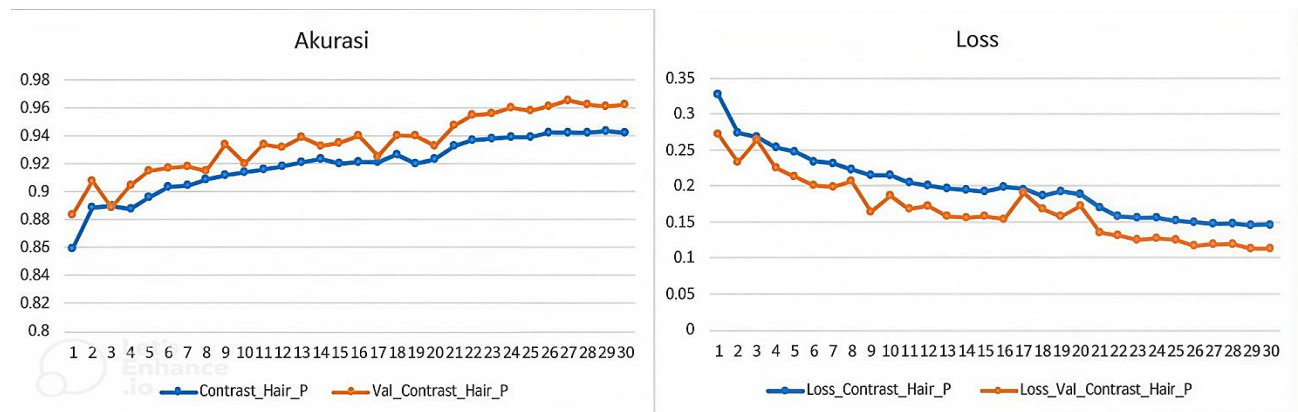
Gambar 11. Hasil Metode Kedua Skenario Kedua

DeepMelaNet dengan Peningkatan Kontras Citra pada Gambar 11 menunjukkan kinerja yang sangat baik dan stabil, dengan kurva pelatihan yang konsisten tanpa fluktuasi berlebih, menandakan model tidak mengalami *overfitting*. Akurasi validasi yang dicapai adalah 96%. Dengan performa yang didapatkan pada *Precision* sebesar 97,9%, *Recall* 93,4%, dan *F1-Score* 95,6%, disertai dengan rata-rata waktu pelatihan per-*Epochs* selama 353,8 detik.



Gambar 12. Hasil Metode Ketiga Skenario Kedua

Merujuk pada Gambar 12, implementasi *DeepMelaNet* dengan Metode Gabungan (*Hair Removal* dan *Contrast Enhancement*) menampilkan performa kurva validasi yang jauh lebih superior dibandingkan skenario pertama. Meski masih menyisakan sedikit fluktuasi merupakan hal yang sangat wajar, lonjakan ekstrem tak lagi terlihat, sehingga membuktikan stabilitas model yang sangat baik. Secara impresif, model ini sukses menembus akurasi validasi hingga 94,9%. Kinerja gemilang ini turut diperkuat oleh perolehan *Precision* sebesar 96,8%, *Recall* 92,8%, dan *F1-Score* 94,7%, dengan rata-rata waktu pelatihan yang terbilang efisien di angka 265,6 detik per *epoch*.



Gambar 13. Hasil Metode Keempat Skenario Kedua

Merujuk pada Gambar 13, implementasi *DeepMelaNet* dengan Metode Gabungan Terbalik (*Contrast Enhancement* yang disusul *Hair Removal*) mencatatkan lonjakan performa paling tajam, sekaligus mengukuhkannya sebagai konfigurasi terbaik dari keseluruhan pengujian. Proses pelatihannya terbukti tidak hanya stabil dan efisien, tetapi juga memamerkan kemampuan generalisasi yang sangat memukau. Model pamungkas ini sukses menorehkan akurasi validasi tertinggi di angka 96,5%. Pencapaian luar biasa ini makin disempurnakan oleh perolehan metrik *Precision* yang nyaris sempurna sebesar 98,7%, *Recall* 94,4%, dan *F1-Score* 96,5%, dengan rata-rata waktu pelatihan memakan 317,3 detik per *epoch*.

Berdasarkan hasil pengujian terhadap model *DeepMelaNet* dengan empat metode pra-pemrosesan yang berbeda dan dua skenario *learning rate* (*LR*), kesimpulan utamanya adalah Skenario Pelatihan Kedua (*LR* 0,0001) secara konsisten menghasilkan performa yang lebih stabil dan superior dibandingkan Skenario Pertama (*LR* 0,001). Penyesuaian *LR* ini terbukti mampu menghasilkan konvergensi yang lebih baik dan mengurangi fluktuasi selama pelatihan. Dari seluruh metode yang diuji, konfigurasi yang memberikan hasil paling optimal dan superior adalah "metode gabungan terbalik" (*Contrast Enhancement* diikuti oleh *Hair Removal*) pada skenario kedua. Kombinasi ini terbukti efektif dalam meningkatkan kualitas data, yang menghasilkan proses pelatihan yang stabil dan kemampuan generalisasi yang lebih unggul. Model dengan konfigurasi terbaik ini berhasil mencapai akurasi validasi tertinggi 96,5%, nilai presisi 98,74%, nilai *Recall* 94,4%, dan nilai *F1-Score* 96,52%. Hasil *Confusion Matrix*-nya menunjukkan 492 *True Negative*, 473 *True Positive*, 8 *False Positive*, dan hanya 27 *False Negative*. Pencapaian ini sangat signifikan karena model berhasil menemukan keseimbangan optimal antara presisi dan *Recall*, serta mampu meminimalkan kedua jenis kesalahan (*FP* dan *FN*) secara simultan. Dengan jumlah kasus

ganas yang terlewatkan (FN) yang sangat rendah, model ini dapat dianggap sebagai versi yang paling andal, robust, dan memiliki potensi besar untuk implementasi klinis

Meskipun arsitektur *DeepMelaNet* memiliki kompleksitas yang cukup tinggi dengan total 295,1 juta parameter, model ini terbukti tetap efisien saat diimplementasikan pada perangkat *GPU*. Hal ini dibuktikan dari waktu pelatihan rata-rata yang hanya memakan waktu 317,3 detik per *epoch* (pada konfigurasi terbaik dengan *Learning rate* 0,0001). Konfigurasi ini tidak hanya mencapai akurasi tertinggi (96,5%), tetapi juga menunjukkan stabilitas komputasi yang memadai untuk penggunaan tingkat klinis tanpa kendala akses memori (*out-of-memory*).

IV. SIMPULAN

Penelitian ini sukses membuktikan bahwa optimasi *DeepMelaNet* melalui integrasi teknik *preprocessing* berupa *Contrast Enhancement* dan *Hair Removal* mampu meningkatkan akurasi klasifikasi melanoma secara signifikan. Berdasarkan hasil eksperimen, Skenario Pelatihan Kedua dengan *learning rate* 0,0001 terbukti menjadi konfigurasi paling stabil dan superior dibandingkan skenario pertama karena menghasilkan konvergensi yang jauh lebih baik. Metode "Gabungan Terbalik", yang mendahulukan peningkatan kontras sebelum penghapusan garis rambut, dinobatkan sebagai teknik terbaik dengan perolehan akurasi validasi mencapai 96,5%, *Precision* 98,7%, *Recall* 94,4%, serta *F1-Score* 96,5%. Pencapaian ini sangat krusial bagi implementasi medis karena model berhasil meminimalkan jumlah *False Negative* hingga hanya 27 kasus, di mana performa tersebut dicapai dengan rata-rata waktu pelatihan yang efisien, yakni sebesar 317,3 detik per *epoch*.

Meskipun hasil yang didapatkan sudah sangat memuaskan, pengembangan berkelanjutan tetap diperlukan untuk memperkuat reliabilitas model di lingkungan medis yang lebih luas. Rencana penelitian lanjutan akan difokuskan pada ekspansi *dataset* dengan cakupan citra kanker kulit yang lebih bervariasi guna meningkatkan kemampuan generalisasi model. Selain itu, penerapan metode *cross-validation* serta eksplorasi optimasi *hyperparameter* dan teknik *ensemble* seperti *stacking* disarankan untuk terus mendongkrak performa serta mengurangi kelemahan individual model. Terakhir, pengujian menyeluruh yang mematuhi aspek legal dan regulasi medis harus dilakukan sebagai langkah final sebelum sistem klasifikasi otomatis ini benar-benar siap diimplementasikan untuk membantu diagnosis pakar secara nyata.

DAFTAR PUSTAKA

- [1] H. Naseri and A. A. Safaei, "Diagnosis and prognosis of melanoma from dermoscopy images using machine learning and deep learning: a systematic literature review," *BMC Cancer*, vol. 25, no. 1, 2025.
- [2] P. Thapar, M. Rakhra, M. Alsaadi, A. Quraishi, A. Deka, and J. V. N. Ramesh, "A hybrid Grasshopper optimization algorithm for skin lesion segmentation and melanoma classification using deep learning," *Healthcare Analytics*, vol. 5, no. 100326, 2024.
- [3] I. A. for R. on C. IARC, "Skin cancer." Accessed: Jan. 27, 2026. [Online]. Available: <https://www.iarc.who.int/cancer-type/skin-cancer/>
- [4] S. C. Foundation, "Skin Cancer Facts & Statistics." Accessed: Jan. 27, 2026. [Online]. Available: <https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/>
- [5] M. S. Al Huda, T. E. Shrestha, A. Hossain, N. Bin Sharif, M. A. Ali, and T. I. Erdei, "DeepMelaNet: Advancing Melanoma Stage Classification in Skin Cancer Diagnosis," *Engineering, Technology and Applied Science Research*, vol. 15, no. 1, pp. 19627–19635, 2025.
- [6] R. Sabir and T. Mehmood, "Classification of melanoma skin Cancer based on Image Data Set using different neural networks," *Sci. Rep.*, vol. 14, no. 29704, 2024.
- [7] R. Kaur, H. GholamHosseini, and M. Lindén, "Advanced Deep Learning Models for Melanoma Diagnosis in Computer-Aided Skin Cancer Detection," *Sensors*, vol. 25, no. 3, 2025.
- [8] S. I. Abir *et al.*, "Deep Learning-Based Classification of Skin Lesions: Enhancing Melanoma Detection through Automated Preprocessing and Data Augmentation," *Journal of Computer Science and Technology Studies*, vol. 6, no. 5, 2024.
- [9] M. Kreouzi *et al.*, "Deep Learning for Melanoma Detection: A Deep Learning Approach to Differentiating Malignant Melanoma from Benign Melanocytic Nevi," *Cancers (Basel)*, vol. 17, no. 1, 2025.
- [10] D. Bardou, H. Bouaziz, L. Lv, and T. Zhang, "Hair removal in dermoscopy images using variational autoencoders," *Skin Research and Technology*, vol. 28, no. 3, pp. 445–454, 2022.
- [11] S. Mustafa, A. Jaffar, M. Rashid, S. Akram, and S. M. Bhatti, "Deep learning-based skin lesion analysis using hybrid ResUNet++ and modified AlexNet-Random Forest for enhanced segmentation and classification," *PLoS One*, vol. 20, no. 1, 2025.
- [12] F. Dartiko, R. J. Pradana, R. E. Sari, W. Syahputra, and W. K. Oktoeberza, "Klasifikasi Kanker Kulit Berbasis CNN dengan Metode Hybrid Preprocessing," *Medika Teknika : Jurnal Teknik Elektromedik Indonesia*, vol. 5, no. 2, pp. 124–132, 2024.
- [13] M. H. Javid, "Melanoma Skin Cancer Dataset of 10000 Images." Accessed: Jan. 27, 2026. [Online]. Available: <https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images>