

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Tipe Kulit

Dr. Leslie Boumann dalam bukunya yang berjudul “Cosmetic Dermatology: Principles and Practice, Second Edition” (Baumann et al., 2009) menjelaskan bahwa tipe kulit manusia terbagi menjadi tiga kategori, yaitu:

1) Kulit Berminyak

Kulit berminyak ditandai dengan produksi sebum yang berlebihan yang membuat kulit tampak berkilau atau berminyak, terutama di area T (dahi, hidung, dan dagu) seperti contoh pada gambar 2.1. Kulit berminyak sering memiliki pori – pori yang membesar dan lebih rentan terhadap masalah seperti jerawat dan komedo akibat berlebihan minyak yang menyumbat pori – pori. Meskipun demikian, kulit berminyak cenderung mengalami penuaan lebih lambat dan memiliki perlindungan alami terhadap garis halus dan kerutan. Berdasarkan (Halodoc, 2026) kandungan perawatan yang cocok untuk tipe kulit ini antara lain seperti Niacinamide, Salicyld acid dan centella asiatica.



Gambar 2. 1 Kulit Berminyak
Sumber: (Baumann et al., 2009)

2) Kulit Kering

Kulit kering adalah kondisi dimana kulit kekurangan kelembapan, yang dapat menyebabkan rasa ketat, kasar, dan mengelupas. Kulit kering seringkali memiliki penampilan yang kusam dan dehidrasi. Orang dengan kulit kering mungkin mengalami ketidaknyamanan dan peningkatan sensitifitas, yang terkadang dapat menyebabkan iritasi. Beberapa faktor yang mempengaruhi kondisi ini termasuk lingkungan, faktor genetik, penuaan, dan kurangnya hidrasi yang memadai. Kandungan yang cocok untuk tipe kulit ini berdasarkan (Alodokter, 2025) adalah Niacinamide, Vit C dan eksfoliasi ringan seperti AHA. Untuk lebih jelasnya dapat dilihat pada gambar 2.2



Gambar 2. 2 Kulit Kering
Sumber: (Baumann et al., 2009)

3) Kulit Sensitif

Kulit sensitif seperti ilustrasi pada gambar 2.3 adalah tipe kulit yang cenderung bereaksi lebih cepat terhadap berbagai faktor eksternal, seperti produk perawatan kulit, cuaca, dan polusi. Kulit ini sering kali mengalami gejala seperti kemerahan, gatal, iritasi, atau rasa terbakar. Beberapa penyebab kulit

sensitif meliputi faktor genetik, kondisi lingkungan, serta penggunaan produk yang mengandung bahan kimia keras. Kulit sensitif sendiri terbagi menjadi beberapa tipe, tipe yang pertama atau yang disebut sebagai tipe S1 ditandai dengan komedo dan jerawat. Tipe yang kedua atau yang disebutnya sebagai S2 ditandai dengan reaksi kulit terhadap alergen tertentu yang bisa menyebabkan gatal atau ruam. Tipe yang ketiga atau yang disebut sebagai S3 ditandai dengan kemerahan yang persisten dan pembuluh darah yang terlihat di wajah. Tipe yang keempat atau yang biasa disebut dengan S4 ditandai dengan kondisi skin barrier yang rusak yang menyebabkan kulit mengalami flare-up. Berdasarkan (Halodoc, 2023), kandungan yang dianjurkan untuk tipe kulit ini adalah eksfoliasi seperti Polyhydroxy Acid (PHA), Niacinamide, Squalane dan ekstrak bunga chamomile.



Gambar 2. 3 Kulit Sensitif tipe S1
Sumber: (Baumann et al., 2009)

2.1.2 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Ichwan & Hakiky, 2011).

2.1.3 Deep Learning

Menurut Rosebrock (2017) *Deep Learning* merupakan sub bidang pembelajaran mesin yang juga merupakan sub bidang dari *Artificial Intelligence* (AI). Tujuan utama dari AI sendiri adalah untuk menyediakan sekumpulan algoritma dan juga teknik yang bisa digunakan menyelesaikan masalah yang secara intuitif dan hampir otomatis dilakukan oleh manusia, tetapi juga sangat menantang bagi komputer.

Secara umum, arsitektur *Deep Learning* terdiri atas beberapa komponen utama, yaitu sebagai berikut:

1) *Neural Networks*

Merupakan struktur komputasi yang terinspirasi dari cara kerja otak manusia, yang terdiri dari kumpulan neuron buatan yang saling terhubung dan bekerja untuk memproses melalui pembelajaran pola dari data.

2) *Layer*

Terdiri dari tiga lapisan, yaitu:

- *Input Layer*, merupakan lapisan untuk menerima data mentah.
- *Hidden Layer*, merupakan lapisan untuk melakukan komputasi kompleks untuk mengekstrak fitur.
- *Output Layer*, merupakan lapisan untuk memberikan hasil akhir.

3) *Activation Function*

Merupakan fungsi non – linear yang digunakan untuk membantu model mempelajari pola data kompleks dari data. Beberapa Activation Function yang umum digunakan antara lain Rectified Linear Unit (ReLU), Sigmoid, dan Tanh.

4) *Loss Function*

Merupakan fungsi untuk mengukur seberapa baik model untuk mengukur kesalahan antara hasil prediksi model dengan nilai yang sebenarnya, contoh Loss Function yang umum digunakan antara lain Mean Squared Error dan Cross-Entropy Loss

5) *Backpropagation*

Algoritma untuk memperbarui bobot dalam jaringan saraf melalui penurunan gradien.

Deep Learning memiliki beberapa jenis arsitektur utama yang digunakan

sesuai dengan karakteristik data, yaitu:

- 1) *Convolutional Neural Networks (CNN)*, digunakan untuk pengolahan gambar.
- 2) *Recurrent Neural Networks (RNN)*, digunakan untuk data sequential.
- 3) *Generative Adversarial Networks (GANs)*, digunakan untuk menghasilkan data baru yang menyerupai data training.
- 4) *Transformers*, digunakan untuk memproses bahasa alami (NLP), seperti dalam model GPT.

Menurut (Géron, 2019) dalam bukunya yang berjudul “*Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*” Deep learning sendiri juga memiliki beberapa tahapan yang penting agar model bekerja secara optimal. Tahapan tersebut meliputi proses pengumpulan data, persiapan data, pelatihan model, hingga evaluasi dan implementasi model.

Adapun tahapan umum dalam pengembangan model Deep Learning adalah:

- 1) Akuisisi Data dan Dataset

Tahapan awal dimuali dengan mengumpulkan dataset yang relevan sesuai dengan kebutuhan. Dataset yang digunakan dapat berupa gambar, teks, audio ataupun jenis data lainnya yang memungkinkan model untuk bisa belajar.

- 2) Pemisahan Dataset

Setelah mengumpulkan dataset, tahapan selanjutnya adalah memisahkan dataset menjadi tiga bagian, yaitu data untuk *training, testing dan validation*.

- Training data digunakan untuk melatih model.
- Validation data digunakan untuk melakukan evaluasi terhadap performa model selama model dilatih.
- Testing data digunakan untuk menguji kemampuan model untuk memprediksi setelah proses pelatihan selesai.

Pembagian dataset dilakukan berdasarkan rasio tertentu sesuai kebutuhan.

3) *Pre – processing* Data

Tahapan ini merupakan proses data dipersiapkan untuk pelatihan model. Pre – processing dapat berupa normalisasi data, *resizing image*, augmentasi data, peningkatan kontras, menghapus data yang tidak sesuai dan berbagai macam teknik yang lainnya untuk meningkatkan kualitas data.

4) Perancangan model

Tahapan ini adalah penentuan arsitektur model yang akan digunakan sesuai dengan karakteristik data dan tujuan penelitian.

5) Optimasi model dan *Hyperparameter Tuning*

Pada tahap ini dilakukan perubahan pada *hyperparameter* untuk memperoleh performa model yang optimal. Adapun parameter terkait meliputi *learning rate*, jumlah *epoch*, *batch size*, pemilihan *optimizer*, *activation function* dan *loss function*. Proses ini dilakukan untuk meningkatkan akurasi model dan mencegah terjadinya *overfitting* atau *underfitting*

6) Evaluasi Model

Tahapan ini dilakukan untuk melakukan evaluasi pada model menggunakan *testing data*. Evaluasi dapat dilakukan menggunakan beberapa matrix, seperti *accuracy*, *precision*, *recall*, *f1-score*, dan *Area Under Curve* (AUC). Selain itu, dapat digunakan juga *confusion matrix* untuk menganalisis kesalahan klasifikasi yang telah dilakukan model. Tahapan ini bertujuan untuk memastikan model mampu melakukan prediksi yang baik pada data yang baru.

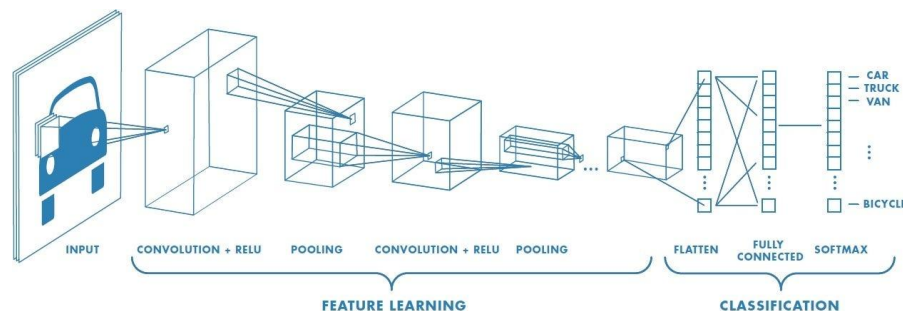
7) *Deployment*

Setelah model dirasa memiliki performa yang baik, model dapat diimplementasikan kedalam aplikasi atau sistem agar dapat digunakan langsung oleh pengguna. Untuk *mobile*, model dapat dikonversi kedalam format TensorFlow Lite (.tflite) agar kompatibel dengan perangkat android.

2.1.4 Convolutional Neural Network (CNN)

Dalam bukunya yang berjudul “Deep Learning” yang ditulis oleh GoodFellow Ian et al. (2016), Convolutional Neural Network atau CNN didefinisikan sebagai sebuah jaringan saraf yang menggantikan lapisan fully-connected dengan lapisan convolution tertentu untuk setidaknya satu dari lapisan dalam jaringan tersebut. CNN dirancang untuk secara otomatis mendeteksi pola dan fitur dalam data masukan terutama data gambar dengan menerapkan operasi konvolusi. Hal ini memungkinkan CNN untuk mengambil hierarki spasial dan mengurangi jumlah parameter dibandingkan dengan lapisan fully-connected. Sehingga membuat CNN sangat cocok untuk melakukan klasifikasi gambar dan juga mendeteksi sebuah objek.

Gambar arsitektur dasar dari Convolutional Neural Network (CNN) dapat dilihat pada Gambar 2.4 (<https://bit.ly/4v6ZCBR>)



Gambar 2. 4 Arsitektur Dasar *Convolutional Neural Network* (CNN)

Pada gambar 2.4 dapat dilihat bahwa proses CNN dimulai dari memasukkan gambar (*input image*) yang selanjutnya diproses melalui beberapa tahapan, yang meliputi convolution layer, fungsi aktivasi ReLU, pooling layer, flatten, fully connected layer, hingga menghasilkan output klasifikasi menggunakan *Softmax*.

Berikut penjelasan masing – masing tahapan tersebut:

1) Input Layer

Lapisan ini merupakan lapisan untuk menerima data sebagai masukan model.

2) Convolution Layer

Lapisan ini berfungsi untuk mengekstraksi fitur penting dari data dengan menggunakan filter atau kernel tertentu. Lapisan ini nantinya akan mempelajari tepi, tekstur, warna dan juga objek dari data yang

digunakan.

3) Activation Function

Lapisan ini berperan untuk memberikan sifat non-linear pada model sehingga model dapat mempelajari pola yang lebih kompleks.

4) Pooling Layer

Lapisan ini digunakan untuk mengurangi dimensi *feature map* tanpa melewatkan informasi yang penting dari data. Proses ini membantu mencegah terjadinya *overfitting*.

5) Flatten Layer

Lapisan ini berfungsi untuk menjadikan data yang sudah di ekstraksi yang berbentuk dua dimensi menjadi satu dimensi agar dapat diproses pada tahap klasifikasi.

6) Fully Connected Layer

Lapisan ini berfungsi untuk melakukan klasifikasi berdasarkan fitur yang sudah dipelajari. Neuron – neuron yang ada pada lapisan ini saling terhubung untuk menghasilkan prediksi akhir.

7) Output Layer

Lapisan ini merupakan lapisan akhir pada CNN, lapisan ini nantinya akan menghasilkan prediksi akhir model.

2.1.5 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Ichwan & Hakiky, 2011).

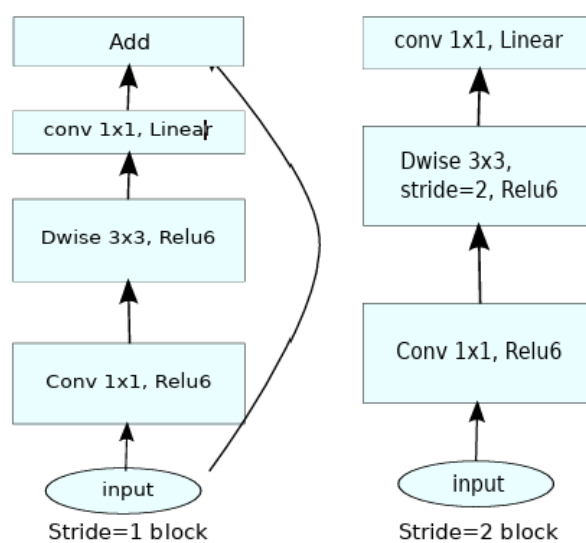
2.1.6 MobileNetV2

Menurut (Saiwaeo et al., 2023) *MobileNetV2* adalah arsitektur yang inovatif dan dirancang khusus untuk memenuhi kebutuhan klasifikasi citra serta ekstraksi

fitur yang efisien dan ringan pada perangkat mobile maupun *embedded system*. Arsitektur ini dikembangkan dari MobileNet generasi pertama oleh Google, dengan menghadirkan berbagai pembaruan untuk menambah performa dan efisiensi model.

Arsitektur ini menggunakan *depth – wise separable convolution*, yang merupakan teknik yang membagi proses konvolusi tradisional menjadi dua proses terpisah, yaitu *depth – wise convolution* dan *point – wise convolution*. Dengan begitu, MobileNetV2 mampu mengurangi kompleksitas komputasi dan jumlah parameter secara signifikan tanpa mengurangi tingkat akurasi. *depth – wise convolution* berfungsi untuk menyaring spasial secara independen pada setiap masukan, sedangkan *point – wise convolution* digunakan untuk menggabungkan hasil penyaringan tersebut.

Pada MobileNetV2 ini, diperkenalkan konsep *inverted residual block* yang dapat meningkatkan kemampuan representasi jaringan sekaligus meminimalkan biaya komputasi. dalam *inverted residual block*, digunakan struktur *bottleneck* yang melakukan kompresi atau pengurangan dimensi pada *feature map* masukan menggunakan konvolusi berukuran 1x1 yang dilanjutkan dengan *depth – wise convolution*. Setelahnya, dimensi diperluas kembali menggunakan konvolusi 1x1. *Bottleneck* juga dapat mengurangi kebutuhan komputasi serta dapat membantu aliran informasi dalam jaringan berlangsung lebih efisien. Arsitektur dari MobileNetV2 dapat dilihat pada gambar 2.5.



Gambar 2. 5 Arsitektur MobileNetV2

2.1.7 Tensorflow

Menurut Géron (2019) dalam bukunya yang berjudul “*Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*” TensorFlow adalah suatu *library* yang sangat bagus digunakan untuk komputasi numerik dan sangat cocok untuk *machine learning* dengan skala yang besar. *Library* ini dikembangkan oleh Google Brain yang dirilis pada bulan November 2015 dan saat ini menjadi *library deep learning* yang sangat banyak digunakan. *Library* ini sering digunakan bersama dengan arsitektur seperti *Convolutional Neural Networks* (CNN) untuk klasifikasi citra.

2.1.8 Confusion Matrix

Confusion matrix merupakan sebuah rangkuman prediksi dari masalah klasifikasi. *Confusion Matrix* diterapkan dengan tujuan untuk melakukan analisa tentang sebaik apa model klasifikasi yang digunakan dalam mengetahui data yang berbeda kelas (Fatmawati & Narti, 2022). Tabel *Confusion Matrix* dapat dilihat pada tabel 2.1

Tabel 2.1 Tabel Confusion Matrix

		Nilai Sebenarnya	
		Positif	Negatif
Nilai Prediksi	Positif	True Positive (TP)	False Negative (FN)
	Negatif	False Positive (FP)	True Negative (TN)

Tabel ini memiliki empat nilai, yaitu:

- 1) *True Positive* (TP) merupakan jumlah nilai positif yang di prediksi benar.
- 2) *True Negative* (TN) merupakan jumlah nilai negatif yang di prediksi benar.
- 3) *False positive* (FP) merupakan jumlah nilai positif yang di prediksi salah.
- 4) *False Negative* (FN) merupakan jumlah nilai negatif yang di prediksi salah.

Setelah hasil prediksi didapat, maka dilakukan perhitungan untuk akurasi, presisi, *recall*, spesifisitas dan *F1 – Score*. Perhitungan tersebut dapat dicari dengan menggunakan persamaan – persamaan berikut:

a) Akurasi

Akurasi menampilkan rasio jumlah prediksi yang benar terhadap total jumlah prediksi. Akurasi memberikan gambaran umum terhadap seberapa baik model melakukan klasifikasi secara keseluruhan. Akurasi didapatkan dengan menggunakan persamaan 2-1.

$$\text{Akurasi} = \frac{(TP + TN)}{TP + TN + FP + FN} \times 100\% \quad 2-1$$

b) Presisi

Presisi mengukur proporsi prediksi *True Positive* (TP) dari seluruh prediksi positif yang dibuat oleh model. Presisi penting untuk memahami seberapa banyak dari hasil positif yang benar – benar relevan. Presisi didapatkan dengan menggunakan persamaan 2-2.

$$\text{Presisi} = \frac{TP}{TP + FP} \times 100\% \quad 2-2$$

c) Recall

Recall atau sensitifitas mengukur proporsi prediksi *True Positive* (TP) dari seluruh prediksi positif. *Recall* penting untuk mengetahui seberapa baik model dapat menangkap semua kondisi yang positif. Recall didapatkan dengan menggunakan persamaan 2-3.

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad 2-3$$

d) *F1 – Score*

F1 – Score digunakan untuk mengevaluasi kerja model, terutama jika dataset yang digunakan tidak seimbang. Nilai dari *F1 – Score* sendiri berkisar antara 0 – 1. Dimana jika hasil perhitungan *F1 – Score* menunjukkan atau mendekati angka 1, maka performa model dapat dikatakan sempurna. Tetapi, jika menunjukkan angka 0, maka performa belum sempurna. *F1 – Score* didapatkam dengan menggunakan persamaan 2-4.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad 2-4$$

2.1.9 Area Under Curve

Area Under Curve (AUC) adalah ukuran luasan area yang terletak dibawah curva ROC. AUC mengukur luas dibawah kurva *Receiver Operating Chacarteristic* (ROC) yang menunjukkan *trade – off* antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Nilai AUC sendiri berkisar antara 0 sampai dengan 1, dimana:

- $AUC \leq 0.5$, menunjukkan model tidak lebih baik dari tebakan acak
- $AUC \geq 0.5$, menunjukkan model bekerja dengan sempurna dalam membedakan antara kelas positif dan kelas negatif.

True Positive Rate (TPR) dan *False Positive Rate* (FPR) pada ROC bisa didapatkan menggunakan persamaan 2-6 dan 2-7 berikut:

$$TPR = \frac{TP}{TP + FN} \quad 2-5$$

$$TFR = \frac{FP}{FP + TN} \quad 2-6$$

Lalu, untuk menghitung AUC, digunakan persamaan 2-8 berikut:

$$AUC = \int_0^1 TPR(FPR), dFPR \quad 2-7$$

2.1.10 Functional Testing

Functional Testing adalah salah satu jenis dari *Blackbox testing* yang merupakan metode pengujian perangkat lunak yang berfokus pada perilaku eksternal dari perangkat lunak. *Functional Testing* bertujuan untuk memastikan bahwa semua fungsi yang diharapkan dari sistem berfungsi dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan. Dalam *Functional Testing*, pemilihan *test case* didasarkan pada spesifikasi kebutuhan yang sedang diuji (Nidhra, 2012).

2.1.11 User Acceptance Testing (UAT)

User Acceptance Testing merupakan pengujian yang dilakukan oleh end user yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. Acceptance testing merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian black box untuk menguji sistem terhadap spesifikasinya. User Acceptance Testing menguji yang dilakukan oleh pengguna sistem. Hasil dari pengujian dapat dijadikan bukti bahwa sistem dapat membantu para pengguna. User Acceptance Testing dilakukan pada pengembangan perangkat lunak bertujuan untuk memastikan sistem memenuhi kebutuhan sebenarnya dari pengguna, bukan hanya spesifikasi sistem. (Wahyudi & Alameka, 2023).

2.2 Penelitian Terdahulu

Penelitian oleh Firaz Bintang Nusantara & Dwi Atmaja, (2018) dengan judul "Penggunaan Metode GLCM dan SVM untuk Klasifikasi Citra Kulit" dengan 2 kelas tipe kulit yaitu kering dan berminyak. menunjukkan bahwa parameter terbaik pada pengujian *Gray Level Co-occurrence Matrix* (GLCM) adalah arah 45° dengan nilai jarak sebesar 1 piksel. Kombinasi parameter ciri orde dua berupa energi dan homogenitas menghasilkan akurasi tertinggi, yaitu sebesar 88,89%. Pada pengujian

kernel trick menggunakan *polynomial*, *linear*, dan *RBF kernel*, hasil menunjukkan bahwa *kernel* yang paling sesuai adalah *kernel polynomial* dengan *kernel option* bernilai 9, yang memberikan akurasi terbaik sebesar 88,89%.

Dalam permasalahan lain, (Saiwaeo et al., 2023) yang berfokus pada klasifikasi tipe kulit manusia menggunakan pendekatan pemrosesan citra dan deep learning. Dalam studi ini, mereka menerapkan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan tipe kulit menjadi tiga kategori: normal, kering, dan berminyak. Metode yang digunakan mencakup peningkatan kualitas citra melalui teknik Contrast Limited Adaptive Histogram Equalization (CLAHE) dan augmentasi data untuk memperbesar variasi dataset, yang menghasilkan total 1.316 gambar dari 329 gambar asli. Hasil penelitian menunjukkan bahwa arsitektur EfficientNet-V2 mencapai akurasi tertinggi sebesar 89.70% saat diuji pada data yang tidak terlihat menggunakan metode 10-fold cross-validation.

(Srinivasu et al., 2021) memaparkan hasil penelitian terkait klasifikasi penyakit kulit menggunakan LSTM dan MobileNetV2. Dengan menggunakan dataset untuk pelatihan dari kaggle dan pengujian menggunakan citra *real time*. Hasil pengujian mendapatkan nilai akurasi sebesar 85,34% dan mampu mengungguli beberapa metode konvensional seperti CNN, FTNN, dan HARIS. MobileNetV2 dipilih karena efisien secara komputasi dan cocok digunakan pada perangkat seperti android, sedangkan penggunaan LSTM membantu meningkatkan akurasi prediksi melalui pemanfaatan informasi dari data sebelumnya. Namun, performa model mengalami penurunan hingga di bawah 80% ketika diuji pada kondisi pencahayaan yang kurang baik.

Penelitian yang lain, oleh (Imanuel et al., 2025) menunjukkan bahwa kombinasi MTCNN + MobileNetV2 mampu mendeteksi citra wajah secara efektif. Penelitian ini memiliki fokus utama yaitu mendeteksi wajah untuk mengenali identitas. Dengan kombinasi dataset publik dan juga pribadi berjumlah 1800 untuk 30 label wajah yang berbeda, hasil dari pengujiannya berada di angka 93%, namun, pada penelitian ini tidak digunakan teknik augmentasi pada gambar seperti rotasi, zoom atau flipping.

Tabel 2.2 Perbandingan variabel penelitian

Kategori	Peneliti 1	Peneliti 2	Peneliti 3	Peneliti 4
Metode	GLCM & SVM	CNN + CLAHE+ EfficientNet-V2	LSTM + MobileNetV2	MTCNN + MobileNetV2
Pengujian	Citra kulit	Citra Kulit	Dataset Kaggle + citra real-time	Citra Wajah
Kelas	Kering, Berminyak	Normal, Kering, Berminyak	Beberapa Jenis Penyakit Kulit	30 label kelas karakteristik wajah
Akurasi	88,89%	89,70%	85,34%	93%
Kelebihan	Akurasi tinggi dengan kernel polynominal	Akurasi tinggi dengan memakai 10-fold cross-validation	Efisien untuk perangkat android	Akurasi tinggi dengan dataset terbatas
Kekurangan	Hanya 2 tipe kulit	Membutuhkan metode 10-fold cross-validation	Akurasi turun pada pencahayaan yang kurang maksimal	Datasset terbatas, dan tidak ada augmentasi data
Metode pengujian	Kernel SVM	10 Fold Cross Validation	Pengujian Real Time	Tidak dijelaskan

Berdasarkan kajian terhadap beberapa penelitian terdahulu pada Tabel 2.2, diketahui bahwa penelitian-penelitian tersebut telah membahas topik yang relevan dengan penelitian ini, namun memiliki keterbatasan dari sisi kelas yang terbatas, ketergantungan pada kondisi pencahayaan dan belum diimplementasikan dalam bentuk sistem berbasis android.

Penelitian ini memiliki perbedaan dengan penelitian sebelumnya karena mengkaji permasalahan klasifikasi tipe kulit dengan tiga kategori yaitu normal, kering dan berminyak. Selain itu, penelitian ini juga mengimplementasikan model dalam bentuk sistem berbasis android sehingga dapat digunakan dengan lebih efisien. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan sistem deteksi tipe kulit berbasis deep learning dalam menjawab perumusan masalah serta melengkapi hasil-hasil penelitian yang telah ada sebelumnya.