

BAB 2

LANDASAN TEORI

Dalam penelitian ini, berbagai teori dan penelitian terdahulu digunakan sebagai landasan dalam pengembangan sistem klasifikasi penyakit kulit berbasis *Deep Learning*. Kajian pustaka mencakup konsep penyakit kulit, metode *Convolutional Neural Network* (CNN), arsitektur *deep learning* yang digunakan, teknik penanganan ketidakseimbangan data, metode *Explainable Artificial Intelligence* (XAI), serta metrik evaluasi untuk mengukur performa model klasifikasi. Pemahaman terhadap teori-teori tersebut diperlukan untuk mendukung proses perancangan, implementasi, dan evaluasi model yang digunakan dalam penelitian.

2.1 Penyakit Kulit dan Citra Dermatoskopi

Bagian ini membahas konsep dasar mengenai penyakit kulit dan citra dermatoskopi sebagai landasan dalam penelitian. Pembahasan meliputi karakteristik penyakit kulit, penggunaan citra dermatoskopi dalam proses diagnosis, karakteristik lesi kulit, serta dataset HAM10000 yang digunakan sebagai sumber data pada penelitian ini.

2.1.1 Penyakit Kulit

Kulit merupakan organ terluar yang menutupi seluruh permukaan tubuh dan berfungsi sebagai pelindung terhadap berbagai pengaruh dari lingkungan. Selain berperan sebagai penghalang fisik, kulit juga membantu mengatur suhu tubuh, menjaga keseimbangan cairan, serta mendukung fungsi sensorik dan sistem imun [1, 2].

Secara anatomis, kulit tersusun atas tiga lapisan utama, yaitu epidermis, dermis, dan hipodermis. Epidermis berfungsi sebagai lapisan pelindung terluar, dermis mengandung jaringan ikat, pembuluh darah, dan struktur pendukung lainnya, sedangkan hipodermis berperan sebagai penyimpan energi dan pelindung terhadap benturan [1, 2].

Berbagai gangguan dapat terjadi pada kulit, mulai dari infeksi, peradangan, hingga kanker kulit. Salah satu jenis yang paling berbahaya adalah melanoma, yaitu kanker yang berasal dari sel melanosit dan memiliki kemampuan menyebar ke organ

lain apabila tidak terdeteksi sejak dini. Selain melanoma, terdapat pula kanker kulit non-melanoma seperti *basal cell carcinoma* (BCC) dan *squamous cell carcinoma* (SCC). Oleh karena itu, deteksi dini menjadi faktor penting dalam meningkatkan keberhasilan pengobatan dan menurunkan risiko komplikasi [10, 5, 6].

Dalam praktik klinis, diagnosis penyakit kulit umumnya dilakukan melalui pemeriksaan visual dan analisis citra dermatoskopi oleh dokter spesialis kulit. Meskipun metode ini banyak digunakan, hasil diagnosis masih dapat dipengaruhi oleh pengalaman dan subjektivitas pemeriksa sehingga berpotensi menimbulkan perbedaan interpretasi, terutama pada lesi yang memiliki karakteristik visual yang serupa [9].

Perkembangan teknologi kecerdasan buatan mendorong pemanfaatan sistem *Computer-Aided Diagnosis* (CAD) berbasis *machine learning* dan *deep learning* untuk membantu proses diagnosis penyakit kulit. Sistem tersebut mampu menganalisis citra dermatoskopi secara lebih objektif dan konsisten sehingga dapat mendukung tenaga medis dalam melakukan identifikasi dan klasifikasi penyakit kulit secara lebih akurat.

2.1.2 Dermatoskopi dan Citra Dermatoskopi

Dermatoskopi merupakan teknik pemeriksaan kulit non-invasif yang menggunakan alat dermatoskop untuk mengamati struktur kulit secara lebih detail. Teknik ini memungkinkan visualisasi pola pigmen, struktur vaskular, dan karakteristik lesi yang tidak dapat diamati dengan jelas menggunakan mata telanjang [11, 12].

Dalam dermatologi, citra dermatoskopi banyak digunakan untuk membantu diagnosis lesi kulit karena mampu memberikan informasi visual yang lebih kaya dibandingkan pemeriksaan visual biasa. Oleh karena itu, citra dermatoskopi menjadi sumber data utama dalam berbagai penelitian klasifikasi penyakit kulit berbasis kecerdasan buatan [10, 5].

Meskipun memberikan detail visual yang baik, interpretasi citra dermatoskopi secara manual masih dapat dipengaruhi oleh pengalaman dan subjektivitas dokter. Untuk mengatasi keterbatasan tersebut, berbagai metode *deep learning* dikembangkan guna membantu proses analisis dan meningkatkan konsistensi serta akurasi diagnosis penyakit kulit [13].

2.1.3 Karakteristik Lesi Kulit

Lesi kulit memiliki karakteristik visual yang kompleks yang dapat digunakan sebagai indikator dalam proses diagnosis. Karakteristik tersebut meliputi warna, bentuk, batas, tekstur, serta ukuran lesi.

- Warna (coklat, hitam, merah, putih)
- Bentuk (simetris atau asimetris)
- Batas (halus atau tidak teratur)
- Tekstur (halus, kasar, bersisik)
- Diameter dan evolusi lesi

Dalam praktik klinis, karakteristik tersebut sering dirangkum menggunakan aturan ABCDE (*Asymmetry, Border, Color, Diameter, Evolution*) yang digunakan sebagai panduan untuk mendeteksi melanoma secara dini [14].

Karakteristik visual ini juga menjadi dasar bagi model *deep learning* dalam mempelajari pola yang membedakan berbagai jenis lesi kulit pada citra dermatoskopi.

2.1.4 Dataset HAM10000

HAM10000 (*Human Against Machine with 10000 Training Images*) merupakan dataset citra dermatoskopi yang banyak digunakan dalam penelitian klasifikasi penyakit kulit berbasis *deep learning*. Dataset ini menyediakan ribuan citra lesi kulit yang telah diberi label diagnosis sehingga dapat digunakan untuk pelatihan dan evaluasi model klasifikasi [5].

Pada penelitian ini, dataset HAM10000 dikelompokkan ke dalam tiga kelas, yaitu *Benign*, *Intermediate*, dan *Malignant*. Ketiga kelas tersebut merepresentasikan berbagai jenis lesi kulit yang umum ditemukan dalam praktik dermatologi dan memiliki karakteristik visual yang beragam.

Salah satu tantangan pada dataset HAM10000 adalah ketidakseimbangan jumlah data antar kelas (*imbalanced dataset*). Untuk mengurangi pengaruh kondisi tersebut, penelitian ini menerapkan teknik *random undersampling* dan *data augmentation* sehingga distribusi data menjadi lebih seimbang dan model dapat mempelajari karakteristik setiap kelas dengan lebih baik.

Dataset HAM10000 kemudian digunakan sebagai sumber data utama dalam proses pelatihan dan evaluasi model. Pada penelitian ini, dataset tersebut digunakan untuk melatih model VGG16, ResNet18, dan ResNet34 dalam melakukan klasifikasi penyakit kulit berbasis citra dermatoskopi. Penggunaan dataset yang sama pada seluruh model memungkinkan proses perbandingan performa dilakukan secara lebih adil dan konsisten.

2.2 Deep Learning untuk Klasifikasi Citra Medis

Bagian ini menjelaskan konsep-konsep dasar *deep learning* yang mendukung penelitian. Pembahasan mencakup *machine learning*, *transfer learning*, *data augmentation*, dan penanganan ketidakseimbangan data yang menjadi dasar dalam pengembangan model klasifikasi penyakit kulit.

2.2.1 Machine Learning dan Deep Learning

Machine Learning (ML) merupakan cabang dari kecerdasan buatan yang memungkinkan sistem mempelajari pola dari data untuk melakukan prediksi atau pengambilan keputusan secara otomatis [15]. Teknologi ini telah banyak diterapkan dalam berbagai bidang, termasuk analisis data dan citra medis.

Salah satu perkembangan penting dalam ML adalah *deep learning*, yaitu pendekatan yang memanfaatkan jaringan saraf tiruan berlapis untuk mempelajari representasi data secara otomatis [3]. Berbeda dengan metode tradisional yang memerlukan ekstraksi fitur secara manual, *deep learning* mampu mengekstraksi fitur langsung dari data mentah sehingga lebih efektif dalam menangani data yang kompleks.

Dalam bidang medis, *deep learning* banyak digunakan pada sistem *Computer-Aided Diagnosis* (CAD) untuk membantu proses deteksi dan klasifikasi penyakit berdasarkan citra medis [4]. Kemampuannya dalam mengenali pola visual yang kompleks menjadikan metode ini banyak dimanfaatkan pada klasifikasi penyakit kulit berbasis citra dermatoskopi.

2.2.2 Transfer Learning

Transfer Learning merupakan pendekatan yang memanfaatkan model yang telah dilatih sebelumnya untuk membantu menyelesaikan tugas baru yang memiliki karakteristik serupa [16]. Pendekatan ini banyak digunakan pada klasifikasi

citra medis karena mampu meningkatkan efisiensi pelatihan dan performa model, terutama ketika jumlah data yang tersedia terbatas.

Secara umum, *Transfer Learning* dapat diterapkan melalui dua strategi, yaitu *feature extraction* dan *fine-tuning*. Pada *feature extraction*, model digunakan sebagai pengekstraksi fitur dan hanya lapisan klasifikasi yang dilatih ulang. Sementara itu, *fine-tuning* memungkinkan sebagian atau seluruh parameter model dilatih kembali agar lebih sesuai dengan karakteristik dataset target.

Dalam penelitian ini, model VGG16, ResNet18, dan ResNet34 menggunakan bobot awal (*pre-trained weights*) dari ImageNet untuk membantu proses pelatihan dan meningkatkan performa klasifikasi pada dataset HAM10000.

2.2.3 Data Augmentation

Data augmentation merupakan teknik untuk meningkatkan jumlah dan variasi data pelatihan melalui transformasi citra tanpa mengubah informasi esensial di dalamnya [17]. Teknik ini banyak digunakan dalam *deep learning*, khususnya pada citra medis, untuk mengatasi keterbatasan data serta meningkatkan kemampuan generalisasi model.

Pada citra dermatoskopi, variasi seperti pencahayaan, orientasi, posisi lesi, dan tekstur kulit dapat memengaruhi performa model. Oleh karena itu, *data augmentation* digunakan untuk menghasilkan variasi citra yang lebih beragam sehingga model dapat mempelajari karakteristik lesi dari berbagai kondisi dan menjadi lebih robust terhadap perubahan visual.

Beberapa teknik augmentasi yang umum digunakan meliputi:

- **Rotation**, yaitu teknik yang memutar citra pada sudut tertentu, misalnya $\pm 10^\circ$, $\pm 20^\circ$, atau $\pm 30^\circ$. Transformasi ini mensimulasikan kondisi ketika orientasi lesi kulit berbeda pada saat proses pengambilan gambar sehingga model dapat mengenali pola lesi meskipun posisi citra berubah.
- **Horizontal dan Vertical Flipping**, yaitu teknik yang membalik citra secara horizontal (kiri ke kanan) atau vertikal (atas ke bawah). Proses ini menghasilkan citra baru dengan orientasi yang berbeda tanpa mengubah karakteristik utama lesi maupun label kelas. Dengan demikian, model menjadi lebih mampu mengenali objek pada berbagai arah orientasi.
- **Zooming atau Scaling**, yaitu teknik yang memperbesar (*zoom in*) atau memperkecil (*zoom out*) citra untuk mensimulasikan perbedaan jarak antara

kamera dan objek. Variasi ini membantu model mempelajari karakteristik lesi pada berbagai ukuran sehingga lebih adaptif terhadap perubahan skala.

- **Brightness dan Contrast Adjustment**, yaitu teknik yang mengubah tingkat kecerahan (*brightness*) dan kontras (*contrast*) citra. Transformasi ini mensimulasikan kondisi pencahayaan yang berbeda pada saat proses akuisisi citra sehingga model menjadi lebih tahan terhadap variasi intensitas cahaya.
- **Translation**, yaitu teknik yang menggeser posisi citra ke arah kiri, kanan, atas, atau bawah tanpa mengubah isi utama gambar. Teknik ini membantu model mengenali lesi meskipun posisinya tidak selalu berada di tengah citra.

Melalui berbagai transformasi tersebut, *data augmentation* menghasilkan citra baru yang tetap merepresentasikan karakteristik lesi asli, tetapi memiliki variasi orientasi, ukuran, posisi, maupun pencahayaan. Penerapan teknik ini memperkaya data pelatihan, mengurangi risiko *overfitting*, serta membantu model mempelajari representasi fitur yang lebih umum sehingga kemampuan generalisasinya terhadap data baru menjadi lebih baik.

2.2.4 Penanganan Ketidakseimbangan Data

Dataset HAM10000 memiliki masalah ketidakseimbangan kelas (*imbalanced dataset*), di mana kelas seperti *melanocytic nevi* mendominasi dibandingkan kelas minoritas seperti *melanoma* dan *dermatofibroma* [5].

Ketidakseimbangan data dapat menyebabkan model menjadi bias terhadap kelas mayoritas sehingga performa klasifikasi pada kelas minoritas menurun. Padahal kelas minoritas sering kali memiliki tingkat kepentingan klinis yang lebih tinggi, seperti melanoma yang merupakan kanker kulit ganas.

Beberapa pendekatan yang umum digunakan untuk mengatasi permasalahan ketidakseimbangan data antara lain:

- *Data Augmentation*
- *Class Weighting*

Pada penelitian ini, ketidakseimbangan distribusi data ditangani dengan menerapkan *class weighting* pada proses pelatihan model. Melalui pendekatan ini, kelas yang memiliki jumlah sampel lebih sedikit diberikan kontribusi yang lebih besar dalam perhitungan fungsi kerugian (*loss function*). Dengan demikian,

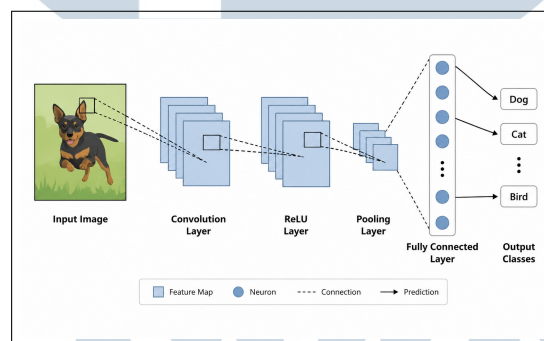
model tidak hanya berfokus pada kelas mayoritas, tetapi juga terdorong untuk mempelajari karakteristik kelas minoritas secara lebih optimal. Penerapan *class weighting* diharapkan dapat menghasilkan performa klasifikasi yang lebih seimbang pada seluruh kelas yang digunakan dalam penelitian.

2.3 Convolutional Neural Network (CNN)

Bagian ini membahas konsep dasar *Convolutional Neural Network* (CNN) sebagai metode utama yang digunakan dalam penelitian. Pembahasan meliputi arsitektur dasar CNN, fungsi *loss*, *optimizer*, serta teknik regularisasi yang digunakan selama proses pelatihan model.

2.3.1 Arsitektur CNN

Arsitektur dasar *Convolutional Neural Network* (CNN) yang digunakan sebagai dasar proses ekstraksi fitur ditunjukkan pada Gambar 2.1.



Gambar 2.1. Arsitektur CNN

Sumber: [18]

Berdasarkan Gambar 2.1, *Convolutional Neural Network* (CNN) merupakan salah satu arsitektur *deep learning* yang dirancang untuk mengolah data citra [19]. Berbeda dengan metode klasifikasi konvensional yang memerlukan proses ekstraksi fitur secara manual, CNN mampu mempelajari dan mengekstraksi fitur secara otomatis melalui proses pelatihan. Kemampuan tersebut menjadikan CNN banyak digunakan dalam berbagai tugas *computer vision*, termasuk klasifikasi penyakit kulit berbasis citra dermatoskopi.

Secara umum, CNN terdiri atas empat komponen utama, yaitu *convolution layer*, fungsi aktivasi, *pooling layer*, dan *fully connected layer*. Masing-masing

komponen memiliki peran dalam mengekstraksi fitur serta menghasilkan prediksi kelas.

Tahap pertama adalah *convolution layer*. Pada lapisan ini, sejumlah filter (*kernel*) digeser pada seluruh bagian citra untuk mengekstraksi karakteristik visual seperti tepi, tekstur, warna, dan bentuk lesi. Hasil proses konvolusi berupa *feature map* yang merepresentasikan fitur tertentu dari citra masukan.

Selanjutnya, *feature map* diproses menggunakan fungsi aktivasi ReLU (*Rectified Linear Unit*) yang didefinisikan sebagai berikut.

$$f(x) = \max(0, x) \quad (2.1)$$

Fungsi ReLU mengubah seluruh nilai negatif menjadi nol sehingga memperkenalkan sifat nonlinier pada jaringan. Penggunaan ReLU membantu jaringan mempelajari hubungan yang kompleks sekaligus mempercepat proses pelatihan.

Tahap berikutnya adalah *pooling layer*, yang bertujuan mengurangi dimensi *feature map* tanpa menghilangkan informasi penting. Salah satu metode yang paling umum digunakan adalah *max pooling*, yang dinyatakan sebagai berikut.

$$Y = \max(X) \quad (2.2)$$

Melalui proses *pooling*, ukuran data menjadi lebih kecil sehingga kebutuhan komputasi berkurang dan model menjadi lebih tahan terhadap perubahan posisi objek pada citra.

Proses konvolusi, aktivasi, dan *pooling* dilakukan secara berulang pada beberapa lapisan sehingga jaringan mampu mempelajari representasi fitur secara bertingkat (*hierarchical feature learning*). Lapisan awal umumnya mendeteksi fitur sederhana seperti garis dan tepi, sedangkan lapisan yang lebih dalam mampu mengenali pola tekstur, bentuk, serta karakteristik lesi kulit yang lebih kompleks.

Setelah proses ekstraksi fitur selesai, seluruh *feature map* diubah menjadi vektor satu dimensi melalui proses *flattening*. Vektor tersebut kemudian diproses oleh *fully connected layer* untuk melakukan klasifikasi.

Pada lapisan keluaran digunakan fungsi *Softmax* untuk menghasilkan probabilitas masing-masing kelas, yang dirumuskan sebagai berikut.

$$P_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (2.3)$$

Kelas dengan probabilitas terbesar dipilih sebagai hasil prediksi akhir. Kombinasi seluruh tahapan tersebut memungkinkan CNN mempelajari pola visual secara otomatis sehingga menghasilkan performa yang baik pada berbagai tugas klasifikasi citra medis.

2.3.2 Fungsi Loss dan Optimizer

Selama proses pelatihan, model perlu mengetahui tingkat kesalahan antara hasil prediksi dengan label sebenarnya. Kesalahan tersebut diukur menggunakan fungsi *loss*. Nilai *loss* digunakan sebagai indikator utama untuk mengevaluasi proses pembelajaran model. Semakin kecil nilai *loss*, semakin dekat hasil prediksi model terhadap label sebenarnya.

Pada penelitian ini digunakan fungsi *Cross Entropy Loss* karena permasalahan yang diselesaikan merupakan klasifikasi dua kelas, yaitu *Benign* dan *Malignant*. Fungsi ini menghitung perbedaan antara probabilitas prediksi yang dihasilkan model dengan label sebenarnya. Nilai *loss* akan semakin kecil apabila probabilitas prediksi semakin mendekati kelas yang benar, sedangkan prediksi yang salah akan menghasilkan nilai *loss* yang lebih besar.

Selama proses pelatihan, nilai *loss* dihitung pada setiap iterasi dan digunakan sebagai dasar untuk memperbarui parameter model. Penurunan nilai *loss* dari satu *epoch* ke *epoch* berikutnya menunjukkan bahwa model semakin mampu mempelajari pola yang terdapat pada data pelatihan. Namun demikian, nilai *loss* tidak dapat digunakan sebagai satu-satunya indikator performa karena model tetap berpotensi mengalami *overfitting*. Oleh karena itu, evaluasi model juga dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *Area Under Curve* (AUC).

Untuk meminimalkan nilai *loss*, digunakan algoritma optimasi (*optimizer*) yang bertugas memperbarui bobot jaringan berdasarkan gradien yang dihitung melalui proses *backpropagation*. Pada penelitian ini digunakan optimizer Adam (*Adaptive Moment Estimation*) karena mampu menyesuaikan *learning rate* secara adaptif pada setiap parameter sehingga proses pelatihan menjadi lebih cepat dan stabil.

Pada setiap iterasi pelatihan, proses dimulai dengan *forward propagation* untuk menghasilkan prediksi. Selanjutnya dihitung nilai *loss* berdasarkan perbedaan antara prediksi dan label sebenarnya. Setelah itu dilakukan *backpropagation* untuk memperoleh gradien setiap parameter terhadap fungsi

loss. Gradien tersebut digunakan oleh optimizer Adam untuk memperbarui bobot jaringan sehingga nilai *loss* pada iterasi berikutnya diharapkan semakin kecil. Proses tersebut diulang hingga mencapai jumlah *epoch* yang telah ditentukan atau memenuhi kriteria penghentian pelatihan.

2.3.3 Overfitting dan Regularisasi

Overfitting merupakan kondisi ketika model memiliki performa yang sangat baik pada data pelatihan, tetapi menunjukkan performa yang kurang baik pada data yang belum pernah dilihat sebelumnya. Kondisi ini terjadi karena model terlalu menyesuaikan diri terhadap data pelatihan sehingga kemampuan generalisasinya menurun.

Untuk mengurangi risiko *overfitting*, penelitian ini menerapkan beberapa teknik regularisasi, yaitu *dropout*, *batch normalization*, dan *early stopping*. *Dropout* bekerja dengan menonaktifkan sejumlah neuron secara acak selama proses pelatihan sehingga model tidak bergantung pada neuron tertentu. *Batch normalization* digunakan untuk menstabilkan distribusi data pada setiap mini-batch sehingga proses pelatihan menjadi lebih cepat dan stabil. Selain itu, *early stopping* diterapkan untuk menghentikan proses pelatihan ketika performa pada data validasi tidak lagi mengalami peningkatan, sehingga risiko *overfitting* dapat dikurangi.

Penerapan berbagai teknik regularisasi tersebut diharapkan mampu meningkatkan kemampuan generalisasi model sehingga menghasilkan performa yang lebih baik pada data pengujian.

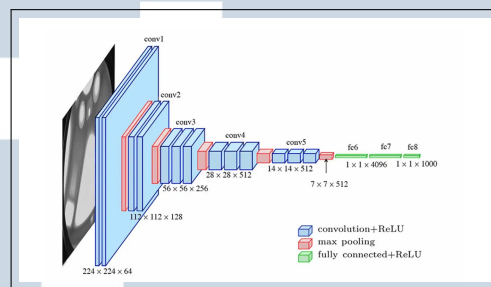
2.4 Arsitektur Model Deep Learning

Bagian ini menjelaskan implementasi model *deep learning* yang digunakan dalam penelitian. Tiga arsitektur, yaitu VGG16, ResNet18, dan ResNet34, diimplementasikan menggunakan pendekatan *transfer learning* dengan bobot awal dari ImageNet. Selain itu, bagian ini juga menjelaskan konfigurasi eksperimen yang digunakan selama proses pelatihan model.

2.4.1 VGG16

VGG16 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh Simonyan dan Zisserman dari Visual Geometry Group (VGG), University of Oxford [20]. Arsitektur ini dirancang untuk

meningkatkan kemampuan ekstraksi fitur melalui penggunaan lapisan konvolusi yang lebih dalam dengan filter berukuran kecil. VGG16 terdiri atas 13 lapisan konvolusi dan 3 lapisan *fully connected*, sehingga memiliki total 16 lapisan berparameter. Penggunaan filter konvolusi berukuran 3×3 secara berulang memungkinkan model mempelajari fitur citra secara bertahap, mulai dari fitur sederhana hingga fitur yang lebih kompleks. Arsitektur VGG16 yang digunakan sebagai dasar penelitian ini ditunjukkan pada Gambar 2.2.



Gambar 2.2. Arsitektur VGG16

Sumber: Adaptasi dari [20]

Cara kerja VGG16 diawali dengan proses konvolusi menggunakan filter berukuran 3×3 yang diterapkan secara berulang pada citra masukan. Hasil konvolusi kemudian diproses menggunakan fungsi aktivasi ReLU untuk memperkenalkan sifat nonlinier pada jaringan. Setelah beberapa lapisan konvolusi, dilakukan proses *max pooling* untuk mengurangi dimensi *feature map* sekaligus mempertahankan informasi yang paling penting. Proses tersebut diulang beberapa kali sehingga model mampu mempelajari representasi fitur secara bertingkat dari fitur sederhana hingga pola visual yang lebih kompleks.

Pada bagian akhir arsitektur, seluruh *feature map* diubah menjadi vektor satu dimensi melalui proses *flattening*. Vektor tersebut kemudian diproses oleh tiga lapisan *fully connected* untuk menghasilkan representasi fitur tingkat tinggi yang digunakan dalam proses klasifikasi. Lapisan keluaran menggunakan fungsi *Softmax* untuk menghasilkan probabilitas setiap kelas, kemudian kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi. Struktur yang sederhana dan konsisten menjadikan VGG16 mudah diimplementasikan serta banyak digunakan sebagai model dasar pada berbagai penelitian klasifikasi citra.

Keunggulan utama VGG16 terletak pada kemampuannya menghasilkan representasi fitur yang baik melalui penggunaan filter konvolusi berukuran kecil secara berulang. Arsitektur ini juga banyak dimanfaatkan dalam pendekatan

transfer learning karena telah tersedia bobot awal (*pre-trained weights*) yang dilatih menggunakan dataset ImageNet. Dalam penelitian ini, VGG16 digunakan sebagai salah satu model pembandingan untuk mengevaluasi performa klasifikasi penyakit kulit berbasis citra dermatoskopi.

2.4.2 ResNet18 dan ResNet34

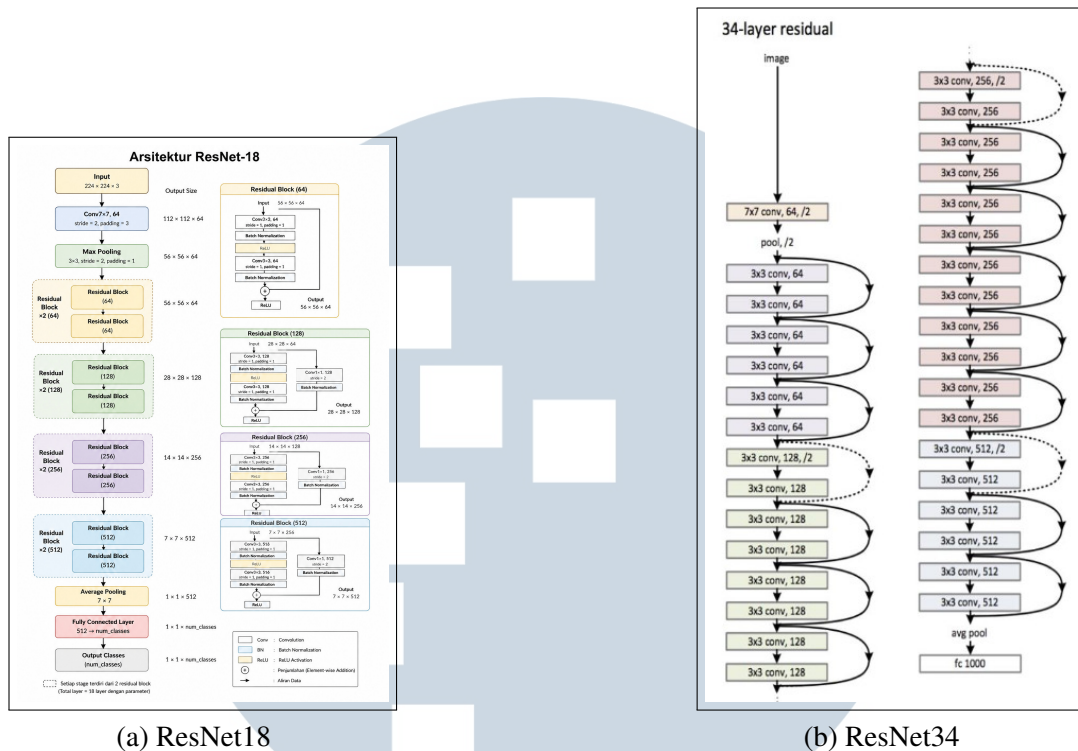
ResNet (*Residual Network*) merupakan arsitektur *deep learning* yang diperkenalkan oleh He dkk. untuk mengatasi permasalahan penurunan performa pada jaringan yang semakin dalam (*degradation problem*) [21]. Berbeda dengan CNN konvensional, ResNet memperkenalkan mekanisme *skip connection* atau *shortcut connection* yang memungkinkan informasi dari lapisan sebelumnya diteruskan secara langsung ke lapisan berikutnya. Mekanisme ini membantu mengurangi permasalahan *vanishing gradient* sehingga proses pelatihan jaringan yang lebih dalam menjadi lebih stabil dan efektif.

Hubungan antara masukan dan keluaran pada *residual block* dinyatakan pada Persamaan (2.4).

$$y = F(x) + x \quad (2.4)$$

Persamaan (2.4) menunjukkan bahwa keluaran suatu *residual block* merupakan hasil penjumlahan antara fungsi residual $F(x)$ dan masukan awal x . Dengan pendekatan ini, jaringan tidak perlu mempelajari pemetaan secara langsung, tetapi cukup mempelajari fungsi residual yang lebih sederhana. Akibatnya, proses optimasi menjadi lebih mudah dan model mampu mempertahankan performa meskipun jumlah lapisan bertambah. Arsitektur ResNet18 dan ResNet34 yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2.3.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.3. Perbandingan arsitektur ResNet18 dan ResNet34
 Sumber: Visualisasi penulis berdasarkan [21] dan [22]

Berdasarkan Gambar 2.3, ResNet18 memiliki jumlah *residual block* yang lebih sedikit dibandingkan ResNet34 sehingga kompleksitas modelnya lebih rendah. Sebaliknya, ResNet34 memiliki kedalaman jaringan yang lebih besar sehingga mampu mempelajari representasi fitur yang lebih kompleks. Kedua arsitektur sama-sama memanfaatkan mekanisme *skip connection* untuk menjaga aliran gradien selama proses pelatihan dan meningkatkan stabilitas pembelajaran.

ResNet18 dan ResNet34 memiliki prinsip kerja yang sama, yaitu melakukan ekstraksi fitur melalui rangkaian lapisan konvolusi yang dihubungkan menggunakan *residual block*. Setiap *residual block* terdiri atas beberapa lapisan konvolusi, fungsi aktivasi ReLU, dan *skip connection* yang menjaga aliran informasi selama proses pelatihan. Setelah seluruh proses ekstraksi fitur selesai, hasilnya diproses menggunakan *global average pooling* dan *fully connected layer* untuk menghasilkan prediksi kelas.

Perbedaan utama kedua arsitektur terletak pada jumlah lapisan yang digunakan. ResNet18 memiliki 18 lapisan sehingga lebih ringan, membutuhkan waktu pelatihan yang lebih singkat, dan memerlukan sumber daya komputasi yang lebih rendah. Sebaliknya, ResNet34 terdiri atas 34 lapisan sehingga mampu

mempelajari representasi fitur yang lebih kompleks, meskipun membutuhkan waktu pelatihan dan kapasitas komputasi yang lebih besar. Dalam penelitian ini, kedua model digunakan untuk mengevaluasi pengaruh kedalaman jaringan terhadap performa klasifikasi penyakit kulit menggunakan pendekatan *transfer learning*.

2.5 Explainable Artificial Intelligence (XAI)

Bagian ini menjelaskan konsep *Explainable Artificial Intelligence* (XAI) yang digunakan untuk meningkatkan interpretabilitas hasil prediksi model. Pembahasan meliputi konsep dasar XAI serta metode Grad-CAM, Grad-CAM++, dan LIME yang digunakan untuk memberikan penjelasan visual terhadap keputusan model.

2.5.1 Konsep Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) merupakan pendekatan yang dikembangkan untuk membantu menjelaskan proses pengambilan keputusan pada model kecerdasan buatan yang kompleks [23]. Pendekatan ini menjadi penting karena model *deep learning* sering kali mampu menghasilkan prediksi dengan tingkat akurasi yang tinggi, namun alasan di balik prediksi tersebut tidak mudah dipahami oleh pengguna.

Dalam bidang kesehatan, kemampuan untuk memahami dasar pengambilan keputusan model memiliki peran yang penting karena hasil prediksi dapat digunakan sebagai informasi pendukung dalam proses diagnosis. Dengan adanya XAI, pengguna dapat memperoleh gambaran mengenai fitur atau bagian data yang paling berpengaruh terhadap hasil prediksi yang dihasilkan model. Informasi tersebut membantu meningkatkan kepercayaan terhadap model sekaligus mendukung proses evaluasi terhadap hasil klasifikasi yang diberikan.

Selain meningkatkan transparansi, XAI juga berperan dalam proses validasi model, identifikasi kesalahan prediksi, dan evaluasi kualitas fitur yang dipelajari oleh model *deep learning*. Oleh karena itu, penggunaan XAI menjadi salah satu aspek penting dalam pengembangan sistem diagnosis berbasis kecerdasan buatan.

Pada penelitian ini digunakan tiga metode XAI, yaitu Grad-CAM, Grad-CAM++, dan LIME untuk menjelaskan hasil prediksi model klasifikasi penyakit kulit.

2.5.2 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) merupakan metode *Explainable Artificial Intelligence* (XAI) yang digunakan untuk memvisualisasikan area citra yang paling berpengaruh terhadap prediksi model *deep learning* [24]. Metode ini menghasilkan *heatmap* yang menunjukkan bagian citra yang menjadi fokus perhatian model saat melakukan klasifikasi.

Grad-CAM memanfaatkan informasi gradien dari kelas target terhadap peta fitur pada lapisan konvolusi terakhir untuk menghasilkan peta aktivasi. Area dengan kontribusi yang lebih besar terhadap hasil prediksi akan ditampilkan dengan intensitas warna yang lebih tinggi pada *heatmap*.

Persamaan Grad-CAM dapat dituliskan sebagai:

$$L^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right) \quad (2.5)$$

dengan L^c sebagai peta aktivasi untuk kelas c , α_k^c sebagai bobot gradien, dan A^k sebagai peta fitur pada lapisan konvolusi terakhir.

Dalam penelitian ini, Grad-CAM digunakan untuk membantu menginterpretasikan hasil klasifikasi penyakit kulit dengan menunjukkan area lesi yang menjadi dasar pengambilan keputusan model.

2.5.3 Grad-CAM++

Grad-CAM++ merupakan pengembangan dari Grad-CAM yang diperkenalkan untuk meningkatkan akurasi lokalisasi objek pada citra [25]. Berbeda dengan Grad-CAM yang menggunakan rata-rata gradien sebagai bobot, Grad-CAM++ menerapkan pembobotan gradien yang lebih kompleks sehingga mampu menangani beberapa area penting dalam satu citra secara lebih baik.

Keunggulan utama Grad-CAM++ adalah kemampuannya menghasilkan visualisasi yang lebih detail dan lebih akurat dibandingkan Grad-CAM, terutama pada citra yang memiliki lebih dari satu objek atau lebih dari satu area yang berkontribusi terhadap prediksi model.

Dalam penelitian klasifikasi penyakit kulit, Grad-CAM++ digunakan untuk memberikan interpretasi visual yang lebih mendalam terhadap area lesi yang memengaruhi keputusan model.

2.5.4 LIME

Local Interpretable Model-agnostic Explanations (LIME) merupakan metode interpretasi yang bersifat *model-agnostic*, sehingga dapat digunakan pada berbagai jenis model machine learning maupun deep learning [9].

LIME bekerja dengan membangun model sederhana di sekitar suatu prediksi untuk menjelaskan perilaku model yang kompleks pada area lokal tertentu. Pada citra, LIME membagi gambar menjadi beberapa segmen (*superpixel*) dan menganalisis kontribusi masing-masing segmen terhadap hasil prediksi.

Keunggulan LIME terletak pada fleksibilitasnya karena tidak bergantung pada arsitektur model tertentu. Hasil visualisasi LIME menunjukkan bagian citra yang memberikan kontribusi positif maupun negatif terhadap keputusan model.

Dalam penelitian ini, LIME digunakan sebagai metode interpretasi tambahan untuk membandingkan hasil visualisasi yang diperoleh dari Grad-CAM dan Grad-CAM++, sehingga diperoleh pemahaman yang lebih komprehensif mengenai proses pengambilan keputusan model.

2.6 Metode Evaluasi

Evaluasi model dilakukan menggunakan data pengujian untuk mengetahui kemampuan model dalam mengklasifikasikan citra lesi kulit secara tepat. Pada penelitian ini, performa model dianalisis menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, ROC, dan AUC. Penggunaan beberapa metrik tersebut bertujuan untuk memberikan penilaian yang lebih menyeluruh terhadap hasil klasifikasi yang dihasilkan oleh model.

2.6.1 Confusion Matrix

Confusion Matrix digunakan untuk mengevaluasi hasil klasifikasi dengan membandingkan prediksi model terhadap label sebenarnya. Matriks ini memberikan informasi mengenai jumlah prediksi yang benar maupun salah pada setiap kelas sehingga performa model dapat dianalisis secara lebih detail.

Komponen utama dalam *Confusion Matrix* meliputi:

- *True Positive* (TP), yaitu data positif yang berhasil diklasifikasikan sebagai positif.

- *True Negative* (TN), yaitu data negatif yang berhasil diklasifikasikan sebagai negatif.
- *False Positive* (FP), yaitu data negatif yang keliru diklasifikasikan sebagai positif.
- *False Negative* (FN), yaitu data positif yang keliru diklasifikasikan sebagai negatif.

Keempat komponen tersebut digunakan sebagai dasar dalam perhitungan berbagai metrik evaluasi untuk mengukur kualitas hasil klasifikasi yang dihasilkan model.

2.6.2 Accuracy

Accuracy digunakan untuk menggambarkan tingkat keberhasilan model dalam melakukan klasifikasi pada seluruh data pengujian. Metrik ini menunjukkan seberapa besar proporsi data yang berhasil diklasifikasikan dengan benar dibandingkan dengan jumlah keseluruhan data yang dievaluasi. Semakin tinggi nilai *accuracy*, semakin baik kemampuan model dalam menghasilkan prediksi yang sesuai dengan label sebenarnya.

Nilai *accuracy* dihitung menggunakan persamaan berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

Keterangan:

- *TP* (*True Positive*) : jumlah data positif yang berhasil diprediksi sebagai positif.
- *TN* (*True Negative*) : jumlah data negatif yang berhasil diprediksi sebagai negatif.
- *FP* (*False Positive*) : jumlah data negatif yang keliru diprediksi sebagai positif.
- *FN* (*False Negative*) : jumlah data positif yang keliru diprediksi sebagai negatif.

Meskipun *accuracy* dapat memberikan gambaran umum mengenai performa model, metrik ini belum tentu mencerminkan kemampuan klasifikasi secara menyeluruh pada dataset yang memiliki distribusi kelas tidak seimbang. Oleh karena itu, diperlukan metrik evaluasi lain seperti *precision*, *recall*, dan *F1-score* untuk memperoleh analisis yang lebih komprehensif.

2.6.3 Precision

Precision digunakan untuk mengevaluasi keandalan prediksi positif yang dihasilkan model. Metrik ini menunjukkan proporsi data yang diprediksi sebagai positif dan benar-benar termasuk ke dalam kelas positif. Nilai *precision* yang tinggi mengindikasikan bahwa model mampu mengurangi jumlah prediksi positif yang salah (*false positive*), sehingga hasil klasifikasi yang diberikan menjadi lebih akurat.

Secara matematis, nilai *precision* dapat dihitung menggunakan persamaan berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

dengan *TP* menyatakan jumlah data positif yang berhasil diprediksi dengan benar dan *FP* menyatakan jumlah data negatif yang keliru diprediksi sebagai positif.

Dalam klasifikasi penyakit kulit, nilai *precision* yang tinggi menunjukkan bahwa sebagian besar lesi yang diprediksi sebagai kelas tertentu memang benar berasal dari kelas tersebut. Oleh karena itu, metrik ini penting untuk mengukur tingkat kepercayaan terhadap hasil prediksi yang dihasilkan model.

2.6.4 Recall

Recall digunakan untuk mengukur kemampuan model dalam menemukan data yang benar-benar termasuk ke dalam kelas positif. Metrik ini menunjukkan seberapa banyak sampel positif yang berhasil dikenali dengan benar oleh model dibandingkan dengan seluruh sampel positif yang tersedia pada data pengujian. Nilai *recall* yang tinggi mengindikasikan bahwa model mampu mengurangi jumlah kasus positif yang terlewatkan (*false negative*).

Secara matematis, nilai *recall* dapat dihitung menggunakan persamaan berikut:

$$Recall = \frac{TP}{TP + FN} \quad (2.8)$$

dengan *TP* menyatakan jumlah data positif yang berhasil diprediksi dengan benar dan *FN* menyatakan jumlah data positif yang keliru diprediksi sebagai negatif.

Dalam klasifikasi penyakit kulit, *recall* memiliki peran yang penting karena menunjukkan kemampuan model dalam mendeteksi kasus yang termasuk ke dalam kelas target. Nilai *recall* yang tinggi membantu mengurangi risiko terlewatnya lesi yang berpotensi berbahaya sehingga model dapat memberikan hasil klasifikasi yang lebih andal.

2.6.5 F1-Score

F1-score digunakan untuk mengevaluasi keseimbangan antara kemampuan model dalam menghasilkan prediksi yang tepat dan kemampuannya dalam menemukan seluruh data yang termasuk ke dalam kelas positif. Metrik ini memberikan gambaran performa yang lebih menyeluruh karena mempertimbangkan nilai *precision* dan *recall* secara bersamaan.

Nilai *F1-score* dihitung menggunakan rata-rata harmonis dari *precision* dan *recall* sebagai berikut:

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.9)$$

Semakin tinggi nilai *F1-score*, semakin baik keseimbangan antara ketepatan prediksi dan kemampuan model dalam mendeteksi kelas target. Oleh karena itu, metrik ini sering digunakan pada permasalahan klasifikasi yang memiliki distribusi data tidak seimbang karena mampu memberikan penilaian yang lebih representatif dibandingkan hanya menggunakan *accuracy*.

Dalam penelitian klasifikasi penyakit kulit, *F1-score* digunakan untuk mengevaluasi seberapa baik model mampu mengenali lesi pada setiap kelas tanpa mengabaikan aspek ketepatan maupun kelengkapan deteksi yang dihasilkan.

2.6.6 ROC Curve dan AUC

Receiver Operating Characteristic (ROC) merupakan kurva yang menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai nilai *threshold*. Kurva ini digunakan untuk mengevaluasi kemampuan model dalam membedakan antar kelas.

Sementara itu, *Area Under Curve* (AUC) merupakan nilai luas area di bawah kurva ROC yang digunakan untuk mengukur performa keseluruhan model. Nilai AUC berada pada rentang 0 hingga 1. Semakin mendekati nilai 1, semakin baik kemampuan model dalam membedakan antar kelas penyakit kulit.

2.7 Visualisasi Fitur

Visualisasi fitur merupakan teknik yang digunakan untuk memahami representasi data yang dipelajari oleh model *deep learning*. Melalui visualisasi fitur, distribusi dan pola pemisahan antar kelas pada ruang fitur dapat diamati sehingga kualitas representasi yang dihasilkan model dapat dievaluasi [26, 4].

Karena fitur yang dihasilkan model *deep learning* umumnya memiliki dimensi yang sangat tinggi, diperlukan metode reduksi dimensi agar data dapat divisualisasikan dalam ruang dua dimensi atau tiga dimensi. Pada penelitian ini digunakan tiga metode reduksi dimensi, yaitu *t-distributed Stochastic Neighbor Embedding* (t-SNE), *Uniform Manifold Approximation and Projection* (UMAP), dan *Principal Component Analysis* (PCA).

2.7.1 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE merupakan metode reduksi dimensi non-linear yang dikembangkan oleh Van der Maaten dan Hinton [27]. Metode ini memproyeksikan data berdimensi tinggi ke ruang dua dimensi atau tiga dimensi dengan mempertahankan hubungan kedekatan antar sampel.

t-SNE bekerja dengan mengubah jarak antar data pada ruang berdimensi tinggi menjadi distribusi probabilitas, kemudian mencari representasi berdimensi rendah yang memiliki distribusi serupa. Dengan pendekatan tersebut, sampel yang memiliki kemiripan tinggi akan ditempatkan berdekatan pada hasil visualisasi.

Dalam penelitian *deep learning*, t-SNE banyak digunakan untuk mengevaluasi kualitas representasi fitur yang dipelajari model. Sampel yang berasal dari kelas yang sama diharapkan membentuk kelompok (*cluster*) yang terpisah dengan jelas dari kelas lainnya.

2.7.2 Uniform Manifold Approximation and Projection (UMAP)

Uniform Manifold Approximation and Projection (UMAP) merupakan metode reduksi dimensi non-linear yang dikembangkan oleh McInnes dkk. [28].

Metode ini dirancang untuk memproyeksikan data berdimensi tinggi ke ruang berdimensi rendah dengan tetap mempertahankan struktur lokal maupun global data. Dibandingkan dengan beberapa metode reduksi dimensi lainnya, UMAP mampu menghasilkan representasi data yang lebih informatif sehingga banyak digunakan untuk visualisasi dan analisis data berdimensi tinggi.

Cara kerja UMAP diawali dengan membangun graf tetangga terdekat (*k-nearest neighbor graph*) pada ruang berdimensi tinggi untuk merepresentasikan hubungan antar data. Selanjutnya, UMAP membentuk representasi manifold yang menggambarkan struktur data berdasarkan kedekatan antar sampel. Proses optimasi kemudian dilakukan untuk memproyeksikan data ke ruang berdimensi rendah dengan tetap mempertahankan hubungan kedekatan tersebut. Dengan pendekatan ini, sampel yang memiliki karakteristik serupa akan berada pada posisi yang berdekatan, sedangkan sampel yang berbeda cenderung terpisah dengan lebih jelas.

UMAP memiliki beberapa keunggulan, antara lain waktu komputasi yang relatif lebih cepat, kemampuan mempertahankan struktur lokal maupun global data secara lebih konsisten, serta skalabilitas yang baik pada dataset berukuran besar. Selain itu, UMAP mampu menghasilkan visualisasi yang lebih stabil sehingga sering digunakan untuk mengevaluasi representasi fitur yang dihasilkan oleh model *deep learning*. Oleh karena itu, metode ini banyak diterapkan pada berbagai penelitian klasifikasi citra dan analisis data medis.

Pada penelitian ini, UMAP digunakan untuk memvisualisasikan distribusi fitur hasil ekstraksi model VGG16, ResNet18, dan ResNet34. Visualisasi tersebut bertujuan untuk mengamati pola pemisahan antar kelas penyakit kulit berdasarkan representasi fitur yang dipelajari oleh masing-masing model. Hasil visualisasi UMAP kemudian dibandingkan dengan PCA dan t-SNE untuk mengevaluasi kualitas representasi fitur yang dihasilkan oleh ketiga arsitektur *deep learning* yang digunakan.

2.7.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) merupakan metode reduksi dimensi linear yang bertujuan mengubah data berdimensi tinggi menjadi sejumlah komponen utama (*principal components*) yang mampu mempertahankan sebagian besar variasi data [29].

PCA bekerja dengan mencari kombinasi linear dari fitur-fitur asli sehingga diperoleh sumbu baru yang memiliki variansi terbesar. Komponen utama pertama

menjelaskan variasi data terbesar, sedangkan komponen berikutnya menjelaskan variasi yang tersisa secara bertahap.

Metode ini memiliki keunggulan berupa proses komputasi yang relatif sederhana, cepat, dan mudah diinterpretasikan. Oleh karena itu, PCA sering digunakan sebagai metode dasar dalam analisis data berdimensi tinggi dan sebagai pembanding terhadap metode reduksi dimensi non-linear seperti t-SNE dan UMAP.

Pada penelitian ini, PCA digunakan untuk memvisualisasikan representasi fitur hasil ekstraksi model serta membandingkan kemampuan pemisahan kelas yang dihasilkan dengan metode t-SNE dan UMAP.

2.8 Penelitian Terkait

Penelitian terdahulu yang menjadi dasar dalam pengembangan penelitian ini dirangkum pada Tabel 2.1. Tabel tersebut menunjukkan beberapa penelitian yang menggunakan arsitektur CNN, dataset HAM10000, serta ResNet untuk klasifikasi penyakit kulit beserta kontribusi utamanya terhadap bidang penelitian ini.

Tabel 2.1. Penelitian terkait klasifikasi penyakit kulit

Peneliti	Metode	Hasil Penelitian
Esteva et al. [10]	CNN	Model CNN mampu mencapai performa setara dengan dokter dermatologi dalam klasifikasi kanker kulit berbasis citra dermatoskopi.
Tschandl et al. [5]	Dataset HAM10000	Mengembangkan dataset HAM10000 sebagai benchmark utama dalam penelitian klasifikasi lesi kulit berbasis <i>deep learning</i> .
He et al. [21]	ResNet	Arsitektur ResNet meningkatkan performa klasifikasi melalui <i>residual learning</i> dan <i>skip connection</i> untuk mengatasi <i>vanishing gradient</i> .

Berdasarkan Tabel 2.1, sebagian besar penelitian telah memanfaatkan CNN, dataset HAM10000, dan arsitektur ResNet untuk meningkatkan akurasi klasifikasi penyakit kulit. Namun, penelitian-penelitian tersebut umumnya masih berfokus pada peningkatan performa klasifikasi tanpa melakukan analisis representasi fitur maupun interpretabilitas model secara komprehensif.

Oleh karena itu, penelitian ini membandingkan beberapa arsitektur CNN berbasis transfer learning, yaitu VGG16, ResNet18, dan ResNet34, menggunakan dataset HAM10000. Selain mengevaluasi performa klasifikasi, penelitian ini juga menganalisis representasi fitur serta interpretabilitas model melalui visualisasi untuk memberikan pemahaman yang lebih mendalam terhadap proses pengambilan keputusan model.

