

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh (Kassem et al., 2020) dengan judul “Skin Lesions Classification Into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning”. Penelitian ini menggunakan Metode Convolutional Neural Network (CNN) dengan arsitektur GoogleNet untuk mengklasifikasikan berbagai jenis lesi kulit. Dataset yang digunakan untuk analisis dalam penelitian ini adalah ISIC 2019. Pada penelitian ini akurasi maksimum yang dicapai adalah 94.92%.

Penelitian yang dilakukan oleh (Nigar et al., 2022) dengan judul “A Deep Learning Approach Based on Explainable Artificial Intelligence for Skin Lesion Classification”. Pada penelitian ini menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur ResNet-18. Penelitian ini bertujuan untuk membuat sebuah sistem klasifikasi lesi kulit berbasis explainable artificial intelligence (XAI) untuk meningkatkan akurasi klasifikasi lesi kulit. Model XAI yang diusulkan divalidasi menggunakan dataset International Skin Imaging Collaboration ISIC 2019. Hasil penelitian ini menunjukkan bahwa model yang digunakan berhasil mencapai akurasi 94.47%.

Penelitian yang dilakukan oleh (Orlando & Al Rivan, 2023) dengan judul “KLASIFIKASI JENIS KANKER KULIT MANUSIA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK”. Pada penelitian ini menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur VGG-16. Penelitian ini bertujuan untuk mengklasifikasikan jenis kanker kulit dengan memanfaatkan deep learning. Data yang digunakan berasal dari International Skin Imaging Collaboration ISIC 2019, yang berisi delapan jenis kanker kulit. Hasil penelitian ini menunjukkan bahwa model yang digunakan berhasil mencapai akurasi 68.11%.

Penelitian yang dilakukan oleh (Nugroho et al., 2023) dengan judul “Boosting the Performance of Pretrained CNN Architectures on Dermoscopic Pigmented Skin Lesion Classification”. Pada penelitian ini menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur Inception-v3. Penelitian ini bertujuan untuk meningkatkan

performa klasifikasi lesi kulit berpigmen menggunakan CNN pralatih. Hasil penelitian menunjukkan bahwa metode yang diusulkan berhasil meningkatkan performa klasifikasi dari model CNN pralatih, dengan akurasi 96.38%.

Penelitian yang dilakukan oleh (Pandey et al., 2024) dengan judul “Skin cancer classification using non-local means denoising and sparse dictionary learning based CNN”. Pada penelitian ini menggunakan metode non-local means (NLM) untuk menghilangkan noise (denoising) dan metode Convolutional Neural Network (CNN) untuk mengklasifikasikan citra. Dataset yang digunakan berisi citra resolusi tinggi dari ISIC 2019. Hasil penelitian ini menunjukkan bahwa model yang digunakan berhasil mencapai akurasi 81.23%.

Penelitian yang dilakukan pada proyek akhir ini, ialah membuat sebuah aplikasi android yang dapat mendeteksi penyakit kulit dengan deep learning. Algoritma yang akan digunakan adalah Vision Transformers (ViT). Keberhasilan model dalam penelitian ini akan diukur berdasarkan target akurasi minimal 80%, dengan harapan dapat meningkat lebih setelah dilakukan proses optimasi. Pencapaian target ini akan didukung oleh beberapa teknik, seperti augmentasi data dan fine-tuning model, guna meningkatkan kemampuan model dalam melakukan prediksi secara lebih akurat dan dapat dijelaskan. Evaluasi performa model akan dilakukan menggunakan metrik accuracy, precision, recall, dan F1-score untuk memastikan model memiliki kinerja yang optimal sebelum diterapkan dalam aplikasi Android.

Tabel 2.1 Tabel Penelitian Terdahulu

Peneliti	Metode	Accuracy	Precision	F1 score
(Kassem et al., 2020)	Metode CNN GoogleNet	94.92%	80.36%	80.07%
(Nigar et al., 2022)	Metode CNN ResNet	94.97%	93.57%	94.45%
(Orlando & Al Rivan, 2023)	Metode CNN VGG	68.11%	35.12%	-

Peneliti	Metode	Accuracy	Precision	F1 score
(Nugroho et al., 2024)	Metode CNN Inception-v3	96.38%	96.40%	96.29%
(Pandey et al., 2024)	Metode CNN & NLM	81.23%	-	-
Proyek Akhir (Sekarang)	Metode Vision Transformer (ViT)	-	-	-

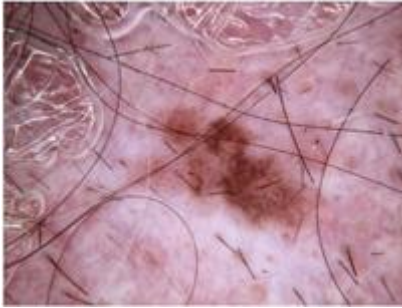
2.2 Landasan Teori

2.2.1 Kanker Kulit

Kanker kulit adalah pertumbuhan jaringan kulit yang tidak terkendali, yang dapat memengaruhi sebagian atau seluruh lapisan kulit. Penyakit ini ditandai oleh struktur sel kulit yang tidak teratur, dengan variasi tingkat kelainan pada kromatin, nukleus, dan sitoplasma. Kanker kulit bersifat ekspansif dan infiltratif, sehingga berpotensi merusak jaringan di sekitarnya. Selain itu, kanker ini juga dapat menyebar melalui jaringan pembuluh darah atau pembuluh getah bening, menyebabkan dampak yang lebih luas (Yohannes & Al Rivan, 2022). Walaupun penyakit kanker kulit ada beragam, pada proyek ini hanya 7 jenis kanker kulit yang digunakan, antara lain yaitu:

1) Actinic Keratoses (AKIEC)

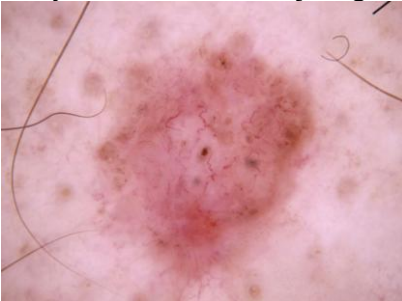
Actinic keratoses adalah lesi kulit prakanker yang disebabkan oleh paparan sinar matahari kronis. Lesi ini sering muncul sebagai bercak kasar dan bersisik pada area kulit yang sering terkena sinar matahari, seperti wajah, tangan, atau lengan.



Gambar 2.1 Penyakit kulit Actinic Keratoses (AKIEC)

2) Basal Cell Carcinoma (BCC)

Karsinoma sel basal adalah jenis kanker kulit paling umum tetapi memiliki tingkat invasif rendah. BCC berkembang di lapisan basal epidermis dan biasanya tampak sebagai benjolan kecil berwarna merah muda, luka yang tidak sembuh, atau bercak bersisik. Jenis ini jarang menyebar (metastasis) tetapi dapat menyebabkan kerusakan jaringan lokal jika tidak diobati.



Gambar 2. 2 Penyakit kulit Basal Cell Carcinoma (BCC)

3) Benign Keratoses (BKL)

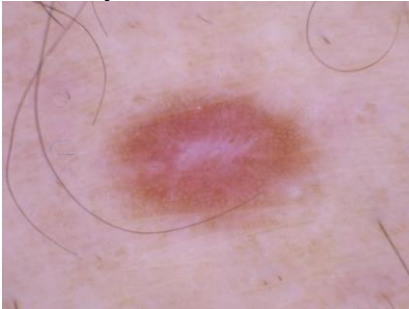
Keratosis jinak mengacu pada lesi kulit yang tidak bersifat kanker, seperti keratosis seboroik. Lesi ini sering berbentuk bintik cokelat hingga hitam, dengan tekstur kasar atau seperti lilin.



Gambar 2. 3 Penyakit kulit Benign Keratoses (BKL)

4) Dermatofibroma (DF)

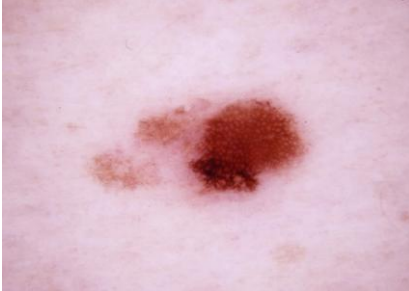
Dermatofibroma adalah tumor jinak pada kulit yang biasanya kecil, keras, dan sering berwarna cokelat atau merah muda. Lesi ini sering terjadi pada tungkai bawah dan sering tidak memerlukan pengobatan kecuali jika menyebabkan ketidaknyamanan.



Gambar 2. 4 Penyakit Dermatofibroma (DF)

5) Melanoma (MEL)

Melanoma adalah bentuk kanker kulit paling berbahaya yang berasal dari melanosit, sel yang menghasilkan pigmen kulit. Melanoma biasanya ditandai oleh perubahan pada tahi lalat (nevus) seperti bentuk yang tidak simetris, tepi yang tidak teratur, warna yang tidak merata, diameter lebih dari 6 mm, atau perubahan yang terus-menerus (prinsip ABCDE).



Gambar 2. 5 Penyakit Melanoma (MEL)

6) Melanocytic Nevi (NV)

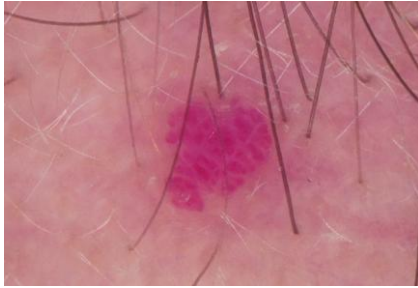
Nevus melanocytic, atau tahi lalat, adalah pertumbuhan kulit yang berasal dari melanosit. Sebagian besar nevi bersifat jinak, tetapi beberapa dapat berkembang menjadi melanoma.



Gambar 2. 6 Penyakit Melanocytic Nevi (NV)

7) Vascular Lesions (VASC)

Lesi vaskular adalah kelainan pada pembuluh darah kulit, seperti hemangioma atau spider angioma. Meskipun sebagian besar lesi vaskular tidak bersifat kanker, ada beberapa yang perlu diawasi karena potensi komplikasi atau penampilan yang menyerupai jenis kanker tertentu.



Gambar 2. 7 Penyakit Vascular Lesions (VASC)

8) Squamous Cell Carcinoma (SCC)

Squamous Cell Carcinoma adalah jenis kanker kulit yang berasal dari sel skuamosa, yaitu sel tipis dan pipih yang terdapat di lapisan tengah hingga luar kulit (epidermis). SCC merupakan kanker kulit kedua paling umum setelah Basal Cell Carcinoma (BCC). SCC bisa tumbuh di bagian kulit yang sering terpapar sinar matahari, seperti wajah, telinga, leher, tangan, dan lengan, tetapi juga bisa muncul di area lain, termasuk mulut, bibir, alat kelamin, atau luka kronis.



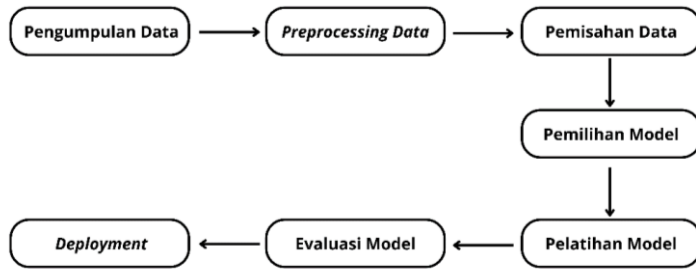
Gambar 2. 8 Penyakit Squamous Cell Carcinoma (SCC)

2.2.2 Deep Learning

Deep Learning merupakan subbidang khusus dari machine learning yang berfokus pada pendekatan baru dalam analisis data, dengan menekankan pembelajaran secara bertahap melalui berbagai lapisan (layers). Metode ini dirancang untuk mempelajari abstraksi tingkat tinggi dari data dengan memanfaatkan struktur arsitektur yang kompleks. Selain

itu, deep learning bertujuan untuk meningkatkan akurasi hasil dan mengoptimalkan waktu pemrosesan dalam berbagai aplikasi komputasi (Halim, 2024). Ada beberapa metode dan arsitektur yang menggunakan layers dalam prosesnya yang dapat digunakan untuk kebutuhan dan tujuan tertentu, yaitu:

- 1) *Convolutional Neural Network (CNN)*
Convolutional Neural Network (CNN) adalah salah satu metode dalam *deep learning* yang digunakan untuk mengklasifikasikan gambar atau video. Selain itu, CNN juga berperan dalam mendeteksi objek yang terdapat dalam suatu gambar serta mengidentifikasi area tertentu di dalamnya (Qotrunnada & Utomo, 2022).
- 2) *Recurrent Neural Network (RNN)*
Recurrent Neural Network (RNN) adalah salah satu metode *deep learning* yang sering digunakan dalam penelitian analisis sentimen. RNN dapat memproses informasi kontekstual dari urutan kata dalam teks secara berurutan dan menyimpan informasi tersebut untuk mengenali pola dalam data (R.A.Pramunendar, D.P.Prabowo, 2022).
- 3) *Long Short Term Memory (LSTM)*
Long Short Term Memory (LSTM) adalah salah satu pengembangan dari *Recurrent Neural Network (RNN)*. Jaringan neural LSTM sangat cocok digunakan untuk klasifikasi dan prediksi berdasarkan data time series, karena bisa ada durasi yang tidak terduga di antara peristiwa penting dalam urutan waktu (Khumaidi et al., 2020).
- 4) *Vision Transformer (ViT)*
Vision Transformer merupakan salah satu model *deep learning* untuk klasifikasi gambar yang memanfaatkan arsitektur *Transformer* dengan membagi gambar menjadi beberapa bagian. Penjelasan lebih lanjut dapat dilihat pada subbab 2.2.3.



Gambar 2. 9 Tahapan Deep Learning

Deep learning sendiri juga memiliki beberapa tahapan yang penting, yaitu:

1. Pengumpulan Data

Tahap pertama dalam deep learning adalah mengumpulkan data, yang menjadi faktor penting karena kualitas data sangat berpengaruh terhadap kinerja model. Data yang dikumpulkan harus sesuai dengan permasalahan yang ingin diselesaikan, baik dalam bentuk gambar, teks, maupun angka. Selain itu, diperlukan jumlah data yang cukup dan representatif agar model bisa belajar secara optimal. Jika jumlah data terbatas, teknik seperti data *augmentation* bisa diterapkan untuk menambah variasi data tanpa mengubah makna aslinya.

2. *Preprocessing Data*

Tahap *preprocessing data* bertujuan untuk menyiapkan data agar dapat digunakan secara optimal oleh model *deep learning*. Proses ini meliputi pembersihan data dari noise serta duplikasi yang dapat memengaruhi hasil model. Selain itu, normalisasi atau standarisasi dilakukan untuk menyelaraskan skala data, sehingga model lebih mudah mengenali pola yang ada. Data juga perlu dikonversi ke format yang sesuai dengan arsitektur model, misalnya menyesuaikan ukuran gambar atau merepresentasikan teks dalam bentuk vektor numerik.

3. Pemisahan Data

Pemisahan data dilakukan untuk membagi data menjadi dua bagian, yaitu data training yang digunakan untuk melatih model dan data testing yang digunakan untuk menguji performa model setelah dilatih. Data training digunakan untuk mengajarkan

model pola-pola yang ada, sementara data testing berfungsi untuk mengevaluasi seberapa baik model dapat memprediksi atau menggeneralisasi data baru yang belum pernah dilihat sebelumnya.

4. Pemilihan Model

Tahap pemilihan model dalam *deep learning* melibatkan penentuan arsitektur yang paling sesuai dengan jenis data dan permasalahan yang ingin diselesaikan. Misalnya, *Convolutional Neural Network* (CNN) sering digunakan untuk pemrosesan gambar, sementara *Recurrent Neural Network* (RNN) atau *Long Short-Term Memory* (LSTM) lebih cocok untuk data berbasis urutan seperti teks dan time series.

5. Pelatihan Model

Tahap pelatihan model adalah proses di mana model deep learning belajar dari data yang telah diproses sebelumnya. Di tahap ini, data training digunakan untuk melatih model melalui dua proses utama, dimulai dengan *forward propagation* yaitu saat data diproses melalui jaringan, kemudian ada *backpropagation* yang berfungsi untuk memperbaiki bobot jaringan berdasarkan kesalahan yang terjadi. Proses ini diulang dalam beberapa *epoch* untuk mengurangi *error* atau *loss function*. Selama pelatihan, berbagai parameter seperti *learning rate*, *batch size*, dan *optimizer* juga disesuaikan untuk membuat model belajar dengan lebih efektif. Memantau *loss function* dan akurasi di setiap *epoch* sangat penting agar model tidak mengalami *overfitting* atau *underfitting*.

6. Evaluasi Model

Tahap evaluasi model dilakukan setelah proses pelatihan untuk mengukur seberapa baik model yang sudah dilatih dalam memprediksi data yang belum pernah dilihat sebelumnya. Di tahap ini, data uji digunakan untuk menguji kinerja model, dengan menggunakan metrik evaluasi seperti *akurasi*, *precision*, *recall*, *F1-score*, atau *loss function*. Jika hasil evaluasi menunjukkan model kurang optimal, maka perlu dilakukan penyesuaian pada arsitektur model atau data yang digunakan.

7. Validasi dan *Hyperparameter Tunning*

Pada tahap ini, performa model dievaluasi menggunakan dataset validasi untuk memastikan model tidak hanya menghafal data

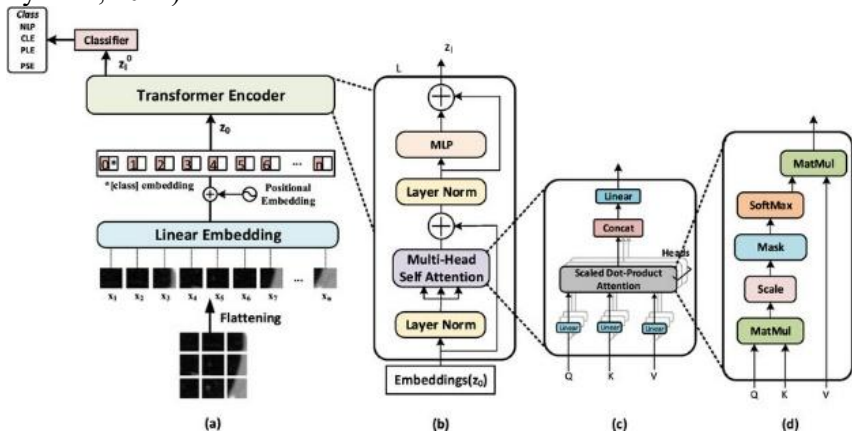
training, tetapi juga dapat mengenali pola dari data baru. Selama proses ini, dilakukan penyesuaian *hyperparameter* untuk menemukan kombinasi terbaik yang dapat meningkatkan kinerja model. *Hyperparameter* sendiri adalah parameter yang tidak dipelajari langsung oleh model, seperti jumlah *neuron*, *learning rate*, *batch size*, atau jumlah lapisan dalam jaringan. Proses *tuning* ini bisa dilakukan secara otomatis menggunakan metode seperti *grid search* atau *Bayesian optimization*. Selain itu, untuk memastikan evaluasi model lebih akurat, dataset dapat dibagi menjadi beberapa subset menggunakan teknik *Cross-Validation* guna menemukan konfigurasi model yang paling optimal.

8. *Deployment*

Tahap *Deployment*, model yang telah melalui proses pelatihan dan evaluasi akan dipersiapkan untuk digunakan secara langsung oleh pengguna. Proses ini mencakup konversi model ke dalam format *TensorFlow Lite* agar dapat diimplementasikan pada *android*. Selain itu, tahap *deployment* juga meliputi integrasi model dengan aplikasi dan pengujian akhir guna memastikan model berfungsi sesuai dengan yang diharapkan. Selama tahap ini, pemantauan performa model secara berkala menjadi aspek penting yang perlu diperhatikan. Jika terjadi penurunan kinerja, maka langkah perbaikan dapat segera dilakukan guna menjaga kualitas prediksi yang dihasilkan oleh model.

2.2.3 Vision Transformer (ViT)

Vision Transformer (ViT) merupakan perkembangan terbaru dalam bidang deep learning yang mengadaptasi arsitektur *Transformer*, yang awalnya dikembangkan untuk pemrosesan bahasa alami. ViT menerapkan konsep *patch embeddings*, *multi-head attention* dan *multi-layer perceptron* untuk meningkatkan pemahaman serta analisis data visual dengan lebih mendalam dan efisien. Pendekatan ini memungkinkan model untuk memproses informasi visual dengan cara yang lebih optimal dibandingkan metode konvensional (Febriyanto & Syofian, 2024).



Gambar 2. 10 Arsitektur Vision Transformer

1) Patch Embeddings

Syarat data masukan untuk arsitektur *vision transformer* adalah citra yang dapat dibagi menjadi patch berukuran $n \times n$. Pada gambar *Gambar 2. 10 Arsitektur Vision Transformer* *Patch embeddings* merupakan proses di mana citra dibagi menjadi beberapa *patch*, yang kemudian diubah menjadi vektor satu dimensi yang berisi nilai piksel dari citra tersebut. Persamaan 2.1 akan menunjukkan rumus yang digunakan untuk mengonversi vektor dua dimensi menjadi vektor satu dimensi dalam proses *patch embeddings* (Figo et al., 2023).

$$X \in \mathbb{R}^{H \times W \times C} \rightarrow X_p \in \mathbb{R}^{N \times (P^2 \cdot C)} \quad (2.1)$$

Keterangan:

- V : Citra masukkan
- X_p : Citra masukkan setelah transformasi
- H : Tinggi citra masukkan
- W : Lebar citra masukkan
- C : Ukuran patch
- N : $\frac{HW}{P^2}$

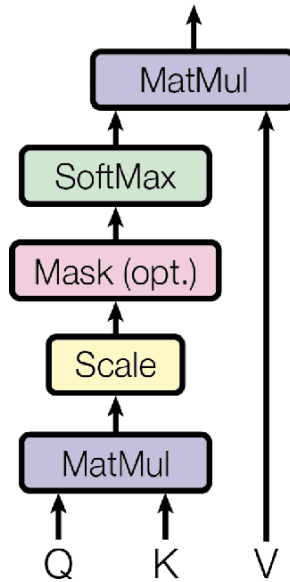
2) Layer Normalization

Proses pelatihan dalam arsitektur *deep learning* umumnya memerlukan sumber daya komputasi yang besar, sehingga dibutuhkan metode yang dapat mengurangi penggunaan daya dan waktu yang diperlukan selama pelatihan. Salah satu cara yang dapat dilakukan adalah dengan menggunakan *Layer Normalization*. Teknik ini mengurangi beban komputasi dan waktu pelatihan dengan menormalkan nilai input menggunakan rata-rata dan standar deviasi. Semakin dekat nilai input dengan rata-rata dan standar deviasi, maka nilai tersebut akan mendekati nol. Persamaan 2.2 akan menjelaskan rumus yang digunakan untuk menghitung *Layer Normalization* (Figo et al., 2023).

$$X^i_{i,k} = \frac{x_{i,k} - \mu^i}{\sqrt{\sigma_i^2 + \epsilon}} \quad (2.2)$$

3) Multi-Head Attention

Setelah melalui *Layer Normalization*, *Layer* berikutnya adalah *Multi-Head Attention*. Pada lapisan ini, proses *attention* dilakukan untuk mengekstrak informasi jangka panjang yang relevan dari vektor masukan. *Multi-Head Attention* bekerja dengan cara menduplikasi vektor masukan menjadi tiga bagian, yaitu *query*, *key* dan *value*. Selanjutnya *query*, *key* dan *value* ini akan diproses menggunakan metode *scaled dot product*. Proses ini dilakukan sesuai dengan jumlah *head* yang ditentukan, kemudian hasilnya akan digabungkan dan dikompresi menjadi ukuran vektor yang telah ditetapkan. Gambar 2. 10 akan menggambarkan secara visual proses yang terjadi selama penerapan *scaled-dot product* (Figo et al., 2023).



Gambar 2. 11 Proses scaled-dot product

4) *Multi Layer Perceptron*

Multi-Layer Perceptron (MLP) adalah model jaringan saraf tiruan yang terinspirasi dari sistem saraf pada tubuh manusia. MLP terdiri dari tiga bagian utama, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Sama seperti sistem saraf manusia, setiap *neuron* dalam MLP memiliki bobot, dan setiap *neuron*, kecuali pada lapisan masukan, dilengkapi dengan fungsi aktivasi non-linear (Figo et al., 2023).

2.2.4 Python

Python adalah salah satu bahasa pemrograman yang sering digunakan oleh perusahaan besar maupun programmer untuk mengembangkan berbagai jenis aplikasi, seperti desktop, web, hingga mobile. *Python* pertama kali diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan hingga sekarang masih dikembangkan oleh *Python Software Foundation*. *Python* telah menjadi salah satu bahasa pemrograman yang digunakan secara luas dalam berbagai sektor, baik industri maupun pendidikan. Hal ini disebabkan oleh bahasanya yang sederhana dan mudah dipahami, ringkas, memiliki sintaks yang intuitif, serta dilengkapi dengan pustaka yang luas (Muhammad Romzi & Kurniawan, 2020).

PyTorch adalah library *Python* yang memudahkan pengembangan proyek deep learning. Library ini mengutamakan fleksibilitas dan memungkinkan model deep learning diekspresikan dalam bahasa *Python*. Karena kemudahan akses dan penggunaan, *PyTorch* cepat diterima oleh komunitas riset. Sejak pertama kali dirilis, *PyTorch* telah menjadi salah satu alat utama dalam deep learning untuk berbagai aplikasi. Sama seperti *Python* dalam dunia pemrograman, *PyTorch* juga menjadi pengantar yang baik untuk memulai deep learning (Febriyanto & Syofian, 2024).

2.2.5 Confusion Matrix

Confusion matrix merupakan sebuah tabel yang digunakan untuk menghitung nilai akurasi atau kecocokan dari data baru yang masuk ke dalam aplikasi dengan data yang sudah dilatih sebelumnya (Normawati & Prayogi, 2021). Sebagai contoh *confusion matrix* untuk klasifikasi citra dapat dilihat pada Tabel 2. 2 *Bentuk Confusion Matrix*.

Tabel 2. 2 Bentuk Confusion Matrix

		Nilai Prekdisi	
		Positif	Negatif
Nilai Sebenarnya	Positif	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
	Negatif	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Penjelasan dari tabel diatas adalah:

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

Berdasarkan hasil prediksi yang diperoleh, perhitungan dapat dilakukan untuk menentukan nilai akurasi, presisi dan recall. Berikut adalah rumus yang digunakan untuk menghitung nilai *akurasi*, *presisi* dan *f1-score*.

$$Akurasi = \frac{Total\ TP}{Total\ Data} \times 100\% \quad (2.3)$$

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (2.4)$$

$$F1 - Score = \frac{2TP}{2TP+FP+FN} \times 100\% \quad (2.5)$$

- 1) Akurasi digunakan untuk mengukur proporsi jumlah prediksi yang benar terhadap keseluruhan data uji. Metrik ini menunjukkan seberapa baik model dalam melakukan klasifikasi secara umum.
- 2) Presisi digunakan untuk menilai ketepatan prediksi positif, yaitu perbandingan antara jumlah prediksi positif yang benar dengan seluruh prediksi positif yang dihasilkan oleh model.
- 3) F1-Score digunakan untuk menggabungkan presisi dan recall melalui rata-rata harmonik, sehingga memberikan keseimbangan antara keduanya. Metrik ini sangat berguna terutama pada dataset yang tidak seimbang (imbalanced), karena mampu mengevaluasi performa model secara lebih adil pada setiap kelas.