

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Tanah

Tanah adalah lapisan atas bumi yang terdiri dari campuran bahan mineral, bahan organik, air, dan udara, yang berfungsi sebagai media tumbuh bagi tanaman serta habitat bagi berbagai organisme. Menurut (Arifin dkk., 2018), tanah adalah kumpulan benda alam di permukaan bumi yang tersusun dalam horizon-horizon, terdiri dari campuran bahan mineral, bahan organik, air, udara, dan merupakan media untuk tumbuhnya tanaman.

1) Jenis-Jenis Tanah

Tanah dapat dikategorikan ke dalam beberapa jenis berdasarkan tekstur, kandungan mineral, serta karakteristik fisik dan kimianya. Berikut adalah sepuluh jenis tanah yang digunakan dalam penelitian klasifikasi tanah menggunakan *Transfer Learning*:

1. Tanah Aluvial: Tanah yang terbentuk dari endapan sungai, memiliki tekstur halus dan sangat subur. Umumnya ditemukan di dataran banjir atau sekitar aliran sungai.
2. Tanah Hitam: Tanah berwarna hitam pekat, kaya akan mineral dan unsur hara, sangat cocok untuk tanaman seperti kapas dan jagung.
3. Tanah Vulkanik: Tanah yang berasal dari pecahan batuan vulkanik yang ringan dan berpori. Memiliki kemampuan drainase yang baik.
4. Tanah Liat: Tanah dengan tekstur sangat halus, mampu menahan air dalam jumlah besar, namun mudah tergenang jika tidak diolah dengan baik.
5. Tanah Laterit: Tanah berwarna merah yang kaya akan kandungan besi dan aluminium. Umum ditemukan di wilayah tropis, namun tingkat kesuburannya rendah.
6. Tanah Lempung: Tanah campuran antara pasir, lanau, dan tanah liat. Memiliki tekstur seimbang dan sangat baik untuk pertanian karena drainase dan kesuburannya tinggi.

7. Tanah Gambut: Tanah yang terbentuk dari sisa tumbuhan yang membusuk, memiliki kandungan bahan organik tinggi, bersifat asam, dan mampu menyimpan air dalam jumlah besar.
8. Tanah Merah: Tanah yang memiliki warna merah khas akibat kandungan oksida besi. Kandungan haranya rendah dan biasanya ditemukan di wilayah kering dan berbatu.
9. Tanah Pasir: Tanah bertekstur kasar, memiliki pori besar sehingga air mudah meresap. Kandungan haranya rendah sehingga memerlukan pemupukan tambahan.
10. Tanah Kuning: Tanah yang berwarna kuning karena kandungan mineral seperti limonit. Umumnya terdapat di daerah lembab dan terbentuk dari pelapukan batuan lanjut.



Gambar 2. 1 Tanah Aluvial (Sumber: konsultanpemetaan.com)

2) Klasifikasi Tanah Menggunakan *Deep Learning*

Deep learning adalah metode kecerdasan buatan yang mampu mengidentifikasi pola kompleks dalam data, termasuk dalam klasifikasi tanah. Model *deep learning*, seperti *Convolutional Neural Networks (CNN)*, sering digunakan dalam analisis citra tanah untuk mengidentifikasi jenis tanah berdasarkan tekstur dan warna (Lecun dkk., 2015). Dalam penelitian ini, model *CNN* dilatih menggunakan *dataset* citra tanah yang mencakup tiga belas jenis tanah, dengan preprocessing yang meliputi augmentasi data, normalisasi, dan ekstraksi fitur. Menurut penelitian terbaru, penggunaan *deep learning* dalam klasifikasi tanah telah menunjukkan akurasi yang tinggi dibandingkan metode konvensional seperti klasifikasi berbasis aturan

dan pengolahan citra tradisional. Dengan memanfaatkan *dataset* yang besar dan teknik *fine-tuning* model, akurasi klasifikasi dapat ditingkatkan secara signifikan.

2.1.2 *Deep Learning*

Deep Learning (DL) adalah salah satu cabang dari *Machine Learning* (ML) yang terinspirasi oleh pola pemrosesan informasi pada otak manusia. *Deep learning* menggunakan jaringan saraf tiruan (*Artificial Neural Networks*, *ANNs*) dengan banyak lapisan untuk mengekstrak pola yang kompleks dari data dalam jumlah besar. *Deep learning* memungkinkan pembelajaran representasi data secara otomatis melalui proses transformasi yang dilakukan oleh berbagai algoritma. Hal ini berbeda dengan pendekatan tradisional dalam *machine learning* yang sering kali memerlukan rekayasa fitur manual untuk meningkatkan performa model.

Deep learning mampu menangani berbagai jenis data seperti gambar, teks, dan suara. Salah satu keunggulan utama dari *deep learning* ini adalah kemampuannya dalam mengekstraksi fitur secara otomatis melalui proses pembelajaran secara berlapis. Dalam arsitektur ini, lapisan awal biasanya mempelajari fitur dasar, seperti tepi atau pola pada gambar, sedangkan lapisan yang lebih dalam menangkap representasi fitur yang lebih abstrak dan kompleks. Keunggulan ini membuat DL sangat efektif dalam berbagai aplikasi seperti *image classification*, *speech recognition*, dan *natural language processing* (NLP).

Deep learning memiliki tiga kategori utama, yaitu *Supervised Learning*, *Semi-supervised Learning*, dan *Unsupervised learning*.

a) *Supervised learning*

Supervised learning adalah teknik pembelajaran yang bergantung pada data berlabel, di mana pasangan *input* dan *output* digunakan untuk melatih model. Model ini mempelajari hubungan antara *input* dan *output* melalui proses optimasi fungsi *loss* untuk meminimalkan kesalahan prediksi. Pendekatan ini sering digunakan dengan algoritma seperti *Convolutional Neural Networks* (CNNs), *Recurrent Neural Networks* (RNNs), dan *Deep Neural Networks* (DNNs).

b) *Semi-supervised learning*

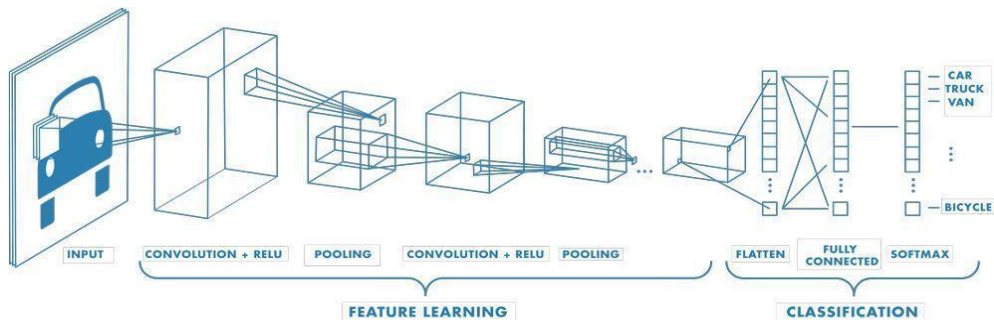
Semi-supervised learning adalah pendekatan yang menggunakan kombinasi data berlabel dan tidak berlabel untuk melatih model. Teknik ini bertujuan untuk mengurangi kebutuhan akan data berlabel yang besar dengan memanfaatkan data tidak berlabel untuk meningkatkan akurasi model. Contoh populer termasuk pengklasifikasi dokumen teks di mana data berlabel sulit diperoleh. Algoritma seperti *Generative Adversarial Networks (GANs)* dan beberapa bentuk RNN sering digunakan dalam *semi-supervised learning*.

c) *Unsupervised learning*

Unsupervised learning adalah proses pembelajaran tanpa data berlabel. Model dirancang untuk mengenali struktur atau hubungan tersembunyi dalam data *input*. Teknik seperti *generative networks*, *clustering*, dan *dimensionality reduction* sering digunakan dalam kategori ini. *Restricted Boltzmann Machines (RBMs)*, *Autoencoders*, dan *GANs* adalah beberapa algoritma populer dalam *unsupervised learning*.

2.1.3 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang dirancang untuk mengolah data berbentuk dua dimensi. *CNN* sering digunakan untuk mempelajari dan mendeteksi fitur dalam gambar (MatWorks, 2024). Berbeda dengan *neural network* pada umumnya yang menggunakan data array satu dimensi sebagai *input*, *CNN* menggunakan data dua dimensi sebagai *input*. *CNN* terdiri banyak neuron yang memiliki *weight*, *bias*, dan *activation function*. Gambar 2.3 menunjukkan tahapan algoritma *CNN* dalam melakukan proses *input* gambar.



Gambar 2. 2 Tahapan proses *CNN* (MatWorks, 2024)

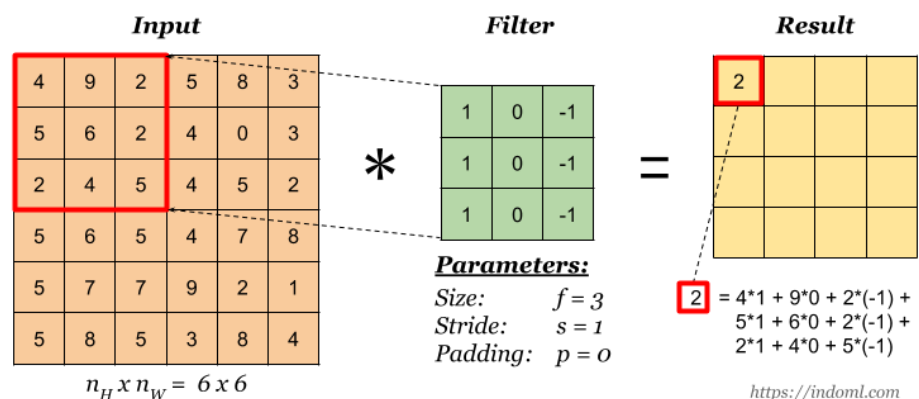
Pada gambar 2.2 kita bisa melihat pada tahapan proses *CNN* ini, terdapat dua bagian utama yaitu, *feature learning* dan *classification*.

1. *Feature Learning*

Feature Learning merupakan bagian dari *CNN* yang bertujuan untuk mengekstraksi fitur-fitur penting dari data *input* (misalnya gambar). Bagian ini terdiri dari beberapa lapisan yang berperan dalam mengenali pola-pola lokal dalam gambar secara bertahap, dari pola sederhana hingga yang lebih kompleks. Berikut adalah penjelasan dari setiap *layer* yang terdapat pada *feature learning* yaitu (Alzubaidi dkk., 2021):

1) *Convolutional Layer*

Convolutional layer merupakan komponen utama dalam arsitektur *CNN*, yang terdiri dari sekumpulan filter konvolusi (*kernel*). Gambar *input*, yang berupa matriks N-dimensi, dikonvolusikan dengan *kernel* untuk menghasilkan peta fitur keluaran. Operasi Konvolusi dimulai dengan gambar masukan yang dapat berupa gambar *grayscale* (saluran tunggal) atau *RGB* (tiga saluran). Misalnya, pada gambar 6x6 dengan *kernel* 3x3, *kernel* ini digeser melintasi gambar secara horizontal dan vertikal. Pada setiap posisi, dilakukan perkalian titik antara *kernel* dan bagian gambar yang sesuai, dan hasilnya dijumlahkan untuk menghasilkan nilai skalar yang menjadi entri peta fitur. Proses ini diulang sampai tidak ada pergeseran lebih lanjut yang memungkinkan.



Gambar 2. 3 Operasi ‘dot’ antara *input* dengan filter (Priyono, 2018)

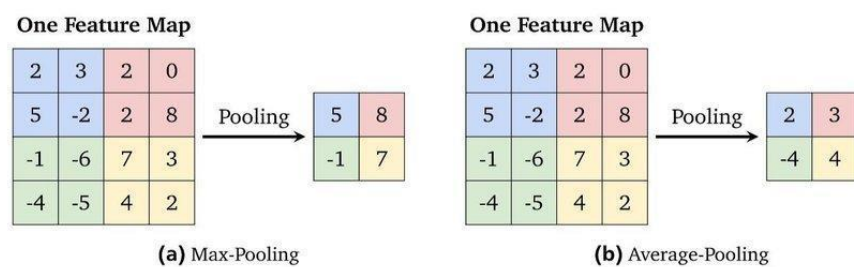
2) *Rectified Linear Unit (ReLU) activation*

ReLU (*Rectified Linear Unit*) adalah fungsi aktivasi yang banyak digunakan dalam jaringan saraf tiruan (*neural network*), terutama dalam konteks *deep learning*. Fungsi ini bertujuan untuk memperkenalkan sifat non-linear pada model, yang penting agar jaringan saraf bisa memecahkan masalah yang lebih kompleks, seperti pengenalan gambar atau pemrosesan bahasa alami. *Output* dari fungsi aktivasi dinyatakan sebagai 0 (nol) jika *input*-nya negatif. Namun jika *input*-nya positif, maka *output*-nya sama dengan nilai *input* aktivasi itu sendiri. Fungsi ReLU dapat dituliskan sebagai:

$$f(x) = \max(0, x) \quad 2 - 1$$

3) *Pooling Layer*

Pooling Layer adalah salah satu komponen penting dalam arsitektur *Convolutional Neural Network (CNN)* yang digunakan untuk mengurangi dimensi dari peta fitur (*feature map*) dan mengurangi jumlah parameter yang perlu dipelajari, sambil mempertahankan informasi penting. Proses ini dikenal dengan istilah *down sampling*. terdapat dua jenis pooling yang umum digunakan, yaitu *max pooling* dan *average pooling*.



Gambar 2. 4 Perbedaan *Max-pooling* dan *Average-Pooling* (Edbert, 2021)

2. *Classification*

Classification berfungsi untuk mengklasifikasikan setiap neuron yang telah diekstraksi selama proses *feature learning*. Bagian ini terdiri dari beberapa lapisan yang saling terhubung satu sama lain. Berikut adalah penjelasan mengenai fungsi-fungsi yang ada pada bagian klasifikasi.

1) *Flattening*

Flattening adalah proses mengubah data berdimensi lebih tinggi (misalnya, matriks) menjadi vektor satu dimensi. Ini dilakukan untuk mempersiapkan data agar bisa diproses lebih lanjut oleh lapisan-lapisan selanjutnya dalam jaringan syaraf (*neural network*). Setelah fitur diekstraksi melalui lapisan konvolusi dan pooling, data biasanya berada dalam bentuk matriks atau tensor dan *flattening* akan meratakan matriks tersebut menjadi satu baris vektor untuk diproses di lapisan berikutnya.

2) *Fully Connected Layer*

Fully connected layer adalah bagian dari *feedforward neural networks* yang mencakup *hidden layer*, *activation function*, *output layer*, dan *loss function*. Lapisan ini sering diterapkan dalam *multi-layer perceptron (MLP)* dengan tujuan untuk mentransformasi dimensi data agar dapat diklasifikasikan secara linear. *Input* pada *fully connected layer* berasal dari data yang dihasilkan melalui *feature learning*. Data ini, yang sudah berbentuk vektor setelah proses *flattening*, kemudian diproses lebih lanjut dalam lapisan ini dan setiap neuron dihubungkan ke neuron-neuron di lapisan berikutnya, sehingga memungkinkan informasi untuk dipertimbangkan secara menyeluruh dan menghasilkan *output* yang lebih akurat.

3) SoftMax

SoftMax adalah fungsi aktivasi yang sering digunakan pada lapisan terakhir dari *neural network* untuk klasifikasi multikelas. Fungsi ini mengubah *output* dari jaringan menjadi probabilitas, yang dapat digunakan untuk menentukan kelas yang paling mungkin. Fungsi SoftMax mengonversi nilai-nilai logit yang tidak terbatas menjadi nilai probabilitas yang berada di antara 0 dan 1, dan jumlah total probabilitas untuk semua kelas selalu sama dengan 1.

2.1.4 VGGNet

VGGNet (Visual Geometry Group Network) merupakan salah satu arsitektur *deep learning* yang dikembangkan berdasarkan konsep *Convolutional Neural Network (CNN)*. Salah satu variannya, yaitu VGG16, memiliki arsitektur yang

sederhana dengan menggunakan *kernel* konvolusi berukuran 3×3 secara berulang untuk meningkatkan kedalaman jaringan. Selain itu, VGG16 juga menggunakan *max-pooling layer* untuk mengurangi ukuran dimensi *feature map* sehingga proses komputasi menjadi lebih efisien. Pada bagian akhir jaringan terdapat dua *fully connected layer* yang masing-masing terdiri dari 4.096 neuron (Mateen et al., 2018).

Pada proses pelatihan, *convolutional layer* berfungsi untuk mengekstraksi fitur dari citra masukan, sedangkan *max-pooling layer* digunakan untuk mengurangi dimensi hasil ekstraksi fitur tanpa menghilangkan informasi penting. Lapisan konvolusi pertama menggunakan 64 buah *kernel* berukuran 3×3 untuk mengenali pola-pola dasar pada gambar. Selanjutnya, hasil ekstraksi fitur diproses pada *fully connected layer* untuk membentuk vektor fitur yang akan digunakan dalam proses klasifikasi.

2.1.5 ResNet

ResNet (Residual Network) merupakan salah satu arsitektur deep *convolutional neural network* (CNN) yang dikembangkan untuk mengatasi penurunan performa pada jaringan yang memiliki banyak lapisan. Konsep utama ResNet adalah penggunaan *shortcut connection* atau *residual connection*, yaitu koneksi yang memungkinkan data melewati beberapa lapisan konvolusi secara langsung. Dengan adanya koneksi tersebut, proses pelatihan jaringan menjadi lebih stabil dan mampu menghasilkan performa yang lebih baik pada model yang memiliki kedalaman tinggi (Rezende et al., 2017).

ResNet memiliki blok utama yang disebut *Bottleneck Block*. Blok ini dirancang dengan dua aturan utama, yaitu setiap lapisan menggunakan jumlah filter yang sama apabila ukuran *feature map* tidak berubah. Sebaliknya, jika ukuran *feature map* diperkecil menjadi setengahnya, maka jumlah filter pada lapisan berikutnya akan ditingkatkan menjadi dua kali lipat agar kemampuan ekstraksi fitur tetap terjaga.

Pada proses pelatihan, citra masukan diubah menjadi format *RGB* dengan ukuran 224×224 piksel agar sesuai dengan ukuran *input* model. Setiap hasil konvolusi kemudian diproses menggunakan *Batch Normalization* sebelum diterapkan fungsi aktivasi ReLU. *Batch Normalization* bertujuan untuk

menstabilkan distribusi data pada setiap lapisan sehingga proses pelatihan menjadi lebih cepat dan stabil.

ResNet menggunakan *identity shortcut* ketika dimensi *input* dan *output* sama. Namun, apabila dimensi berubah, digunakan *projection shortcut* melalui *pointwise convolution* (1×1 convolution) untuk menyesuaikan ukuran dimensi. Pada bagian akhir arsitektur terdapat *fully connected layer* yang diikuti fungsi aktivasi Softmax untuk menghasilkan probabilitas kelas. Salah satu varian yang banyak digunakan, yaitu ResNet-50, memiliki 50 lapisan berbobot dengan sekitar 25,5 juta parameter yang dapat dilatih.

2.1.6 *EfficientNet*

EfficientNet merupakan keluarga arsitektur *Convolutional Neural Network* (CNN) yang terdiri dari beberapa varian, yaitu EfficientNet-B0 hingga EfficientNet-B7. Model EfficientNet-B0 menjadi arsitektur dasar yang kemudian dikembangkan menjadi varian lainnya dengan ukuran dan kompleksitas yang berbeda. Pengembangan tersebut dilakukan untuk memperoleh keseimbangan antara akurasi model dan kebutuhan komputasi (Andre, Rachmad & Purbaningtyas, 2021).

Salah satu keunggulan EfficientNet adalah mampu menghasilkan akurasi yang tinggi dengan jumlah parameter dan kebutuhan komputasi (*Floating Point Operations Per Second* atau *FLOPS*) yang lebih rendah dibandingkan beberapa arsitektur CNN lainnya. Hal ini membuat EfficientNet lebih efisien dalam proses pelatihan maupun pengujian model.

EfficientNet menerapkan metode *compound scaling*, yaitu teknik penskalaan yang dilakukan secara seimbang pada tiga dimensi utama jaringan, yaitu *depth* (kedalaman jaringan), *width* (lebar jaringan), dan *resolution* (ukuran citra masukan). Dimensi *width* berkaitan dengan jumlah channel pada setiap lapisan, *depth* menunjukkan jumlah lapisan yang digunakan dalam arsitektur CNN, sedangkan *resolution* mengacu pada ukuran citra yang digunakan sebagai masukan model. Dengan melakukan penskalaan secara seimbang pada ketiga dimensi tersebut, EfficientNet mampu meningkatkan performa model tanpa meningkatkan kebutuhan komputasi secara berlebihan

2.1.7 Python

Python merupakan bahasa pemrograman tingkat tinggi yang dirancang untuk kemudahan penggunaan dan keterbacaan kode. Diciptakan oleh Guido van Rossum pada akhir 1980-an. Salah satu dari keunggulan python adalah sintaknya yang sederhana dan mirip dengan bahasa inggris, sehingga membuatnya mudah dipahami oleh pemula. Python memiliki komunitas besar yang secara aktif mengembangkan pustaka-pustaka tambahan yang memudahkan berbagai macam pengembangan aplikasi. Dengan lebih dari 125.000 pustaka yang tersedia, Python dapat digunakan untuk berbagai keperluan, seperti analisis data, *machine learning*, dan sebagainya. Library seperti Pandas, NumPy, dan Matplotlib menjadikan Python sebagai pilihan utama di bidang data science dan *machine learning* karena memungkinkan pemrogram untuk melakukan analisis data, visualisasi, serta membangun model algoritma *machine learning* dengan lebih efisien (Dicoding Intern, 2023).

2.1.8 TensorFlow

TensorFlow adalah sebuah *framework open-source* yang digunakan untuk membangun dan melatih model *machine learning* dan *deep learning*. Dikembangkan oleh Google Brain pada tahun 2015, TensorFlow memungkinkan *developer* untuk membuat, melatih, dan menerapkan model-model pembelajaran mesin yang kompleks dalam berbagai aplikasi. Selain itu, TensorFlow menyediakan berbagai alat dan library tambahan yang memungkinkan pengguna untuk melakukan berbagai tugas, mulai dari pelatihan model hingga penyebaran aplikasi *machine learning* ke berbagai perangkat, baik di *cloud*, perangkat *mobile*, maupun di perangkat keras khusus seperti *TPU (Tensor Processing Unit)* untuk komputasi yang lebih cepat. TensorFlow terus berkembang menjadi salah satu *framework* paling populer dalam dunia *machine learning*, memberikan solusi untuk penelitian, pengembangan produk, dan aplikasi komersial di berbagai sektor industri (TensorFlow, 2024).

2.1.9 Confusion Matrix

Confusion matrix merupakan sebuah metode yang digunakan untuk menganalisis kinerja sebuah model klasifikasi. Elemen-elemen pada diagonal utama menunjukkan prediksi yang benar, sementara hasil yang salah

diklasifikasikan muncul di luar diagonal utama. Oleh karena itu, klasifikasi yang ideal akan menghasilkan *confusion matrix* dengan nilai-nilai yang hanya pada diagonal utama, dan semua elemen lainnya bernilai nol. *Confusion matrix* memberikan informasi tentang nilai aktual dan nilai prediksi setelah proses klasifikasi. Efektivitas dalam sistem ditentukan berdasarkan nilai-nilai yang terkandung dalam matriks. Untuk pengklasifikasi dengan dua kelas, confusion marix dapat digambarkan dalam tabel berikut (Arias-Duart dkk., 2023).

Tabel 2. 1 *Confusion Matrix*

Matriks		Prediksi Kelas	
		<i>Positive</i>	<i>Negative</i>
Kelas Sebenarnya	<i>Positive</i>	<i>True Positive</i>	<i>False Negative</i>
	<i>Negative</i>	<i>False Positive</i>	<i>True Negative</i>

Confusion Matrix dapat dijelaskan sebagai berikut:

- 1) *True Positive* (TP): Merupakan jumlah hasil atau prediksi yang benar ketika kelas yang sebenarnya adalah positif.
- 2) *False Positive* (FP): Merupakan jumlah hasil atau prediksi yang salah ketika kelas yang sebenarnya adalah positif.
- 3) *True Negative* (TN): Merupakan jumlah hasil atau prediksi yang benar ketika kelas yang sebenarnya adalah negatif.
- 4) *False Negative* (FN): Merupakan jumlah hasil atau prediksi yang salah ketika kelas yang sebenarnya adalah negatif.

Berdasarkan hasil dari *confusion matrix* yang diperoleh, dilakukan tahap evaluasi untuk mengukur efektivitas model klasifikasi:

- 1) *Accuracy*: Dihitung untuk mengevaluasi kinerja sistem dengan mempertimbangkan jumlah total prediksi yang benar yang dibuat oleh pengklasifikasi. Rumus untuk menghitung akurasi adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad 2 - 2$$

- 2) *Recall*: Dihitung berdasarkan proporsi *input* positif yang teridentifikasi dengan benar. Ini mengukur tingkat TP dan dihitung dengan rumus berikut:

$$Recall = \frac{TP}{TP + FN} \quad 2 - 3$$

- 3) *Precision*: Mengukur jumlah kasus positif yang diprediksi dengan benar oleh pengklasifikasi. Rumus untuk menghitung *precision* adalah sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \quad 2 - 4$$

- 4) *F1-Score*: Merupakan ukuran akurasi yang menggabungkan *precision* dan *recall*. Nilai *F1-Score* berada di antara 0 dan 1, dengan nilai tertinggi di 1 dan terendah di 0. *F1-Score* dihitung dengan rumus berikut.

$$F1\ Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad 2 - 5$$

2.2 Penelitian Terdahulu

Penelitian yang telah dilakukan sebelumnya berkaitan dengan judul proyek akhir ini, antara lain :

Penelitian oleh Gyasi & Purushotham (2023) mengembangkan model *Transfer Learning* khusus bernama Soil-MobiNet untuk klasifikasi morfologi tanah berbasis citra digital. Penelitian ini menggunakan *dataset* VITSoil yang terdiri dari 4.864 citra *RGB* tanah, mencakup sembilan kelas utama. Untuk memperluas data latih, dilakukan augmentasi citra hingga menghasilkan 37.869 citra. Soil-MobiNet dirancang dengan struktur *Transfer Learning* yang efisien, sehingga mampu mencapai akurasi tinggi dalam klasifikasi jenis tanah. Penelitian ini membuktikan bahwa *Transfer Learning* dapat digunakan secara efektif untuk klasifikasi citra tanah dengan data yang beragam. Namun, penelitian ini masih menghadapi keterbatasan pada variasi *dataset* yang hanya diambil dari satu lokasi (VIT Bhopal, India), sehingga potensi generalisasi model terhadap kondisi tanah di wilayah lain belum diuji secara menyeluruh.

Penelitian oleh Natsir (2022) fokus pada klasifikasi tekstur tanah sawah, seperti pasir, lempung, dan lempung berpasir, menggunakan beberapa arsitektur *Transfer Learning*, antara lain DenseNet201, ResNet152, VGG16, Xception, dan MobileNetV2. Hasil pengujian menunjukkan bahwa DenseNet201 memberikan akurasi tertinggi, yaitu 95%, setelah 100 *epoch*. Penelitian ini mengonfirmasi keandalan *Transfer Learning* dalam klasifikasi tekstur tanah berbasis citra. Namun, penelitian ini hanya mengandalkan *dataset* dengan pencahayaan dan kondisi tertentu, sehingga performa model pada citra tanah dengan karakteristik lingkungan yang berbeda masih perlu dieksplorasi.

Penelitian yang dilakukan oleh Langewar & Gurav (2022) membandingkan beberapa arsitektur *Transfer Learning*, seperti ResNet152V2, VGG-16, VGG-19, Inception-ResNetV2, Xception, dan DenseNet201, untuk klasifikasi citra tanah dari empat jenis: alluvial, black, clay, dan red. Hasilnya, *Transfer Learning* dasar berhasil mencapai akurasi pelatihan sebesar 99,86% dan akurasi validasi 97,68%, menunjukkan bahwa *Transfer Learning* mampu melakukan klasifikasi citra tanah

dengan sangat baik. Akan tetapi, penelitian ini menggunakan *dataset* terbatas pada empat kelas utama, sehingga aplikasi model pada kondisi lapangan yang lebih kompleks dengan banyak kelas tanah memerlukan validasi lanjutan.

Terakhir, penelitian oleh Kiran (2023) mengembangkan Light-SoilNet, yaitu *Transfer Learning* ringan untuk klasifikasi citra tanah pertanian yang diambil menggunakan kamera smartphone. Model ini dirancang untuk menjaga efisiensi komputasi sambil tetap menghasilkan akurasi tinggi. Penelitian ini menunjukkan bahwa model *Transfer Learning* ringan dapat menjadi solusi praktis bagi aplikasi klasifikasi citra tanah di perangkat dengan sumber daya terbatas. Meski demikian, penelitian ini hanya fokus pada citra tanah pertanian, sehingga belum diuji pada tipe tanah non-pertanian atau kondisi lingkungan yang ekstrem.

Tabel 2. 2 Penelitian Terdahulu

No .	Nama Peneliti	Judul Penelitian	Algoritma	Hasil
1	(Gyasi & Purushotham, 2023)	<i>Soil-MobiNet: A Convolutional Neural Network Model Base Soil Classification to Determine Soil Morphology and Its Geospatial Location</i>	<i>Transfer Learning</i> (Soil-MobiNet)	Model Soil-MobiNet berhasil mengklasifikasikan 9 kelas citra tanah dari <i>dataset</i> VITSoil dengan akurasi tinggi.
2	(Natsir et al., 2022)	<i>Texture Classification of Paddy Soil using Convolutional Neural Network</i>	<i>Transfer Learning</i> (DenseNet201, ResNet152, dll.)	DenseNet201 menunjukkan akurasi 95% pada tekstur tanah.
3	(Lanjewar & Gurav, 2022)	<i>Convolutional Neural Networks based classifications of soil images</i>	<i>Transfer Learning</i> (ResNet, VGG, Xception, dll.)	<i>Transfer Learning</i> dasar mencapai akurasi pelatihan 99,86% dan akurasi validasi 97,68%, namun hanya

No .	Nama Peneliti	Judul Penelitian	Algoritma	Hasil
				mencakup empat kelas tanah.
4	(Kiran et al., 2023)	<i>Smart soil image classification system using lightweight convolutional neural network</i>	<i>Transfer Learning</i> ringan (Light-SoilNet)	Model <i>Transfer Learning</i> ringan menghasilkan akurasi tinggi, tetapi belum diuji pada tipe tanah non-pertanian atau kondisi ekstrem.