

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada bagian ini akan menjelaskan tentang tinjauan pustaka yang akan dipakai dalam pembuatan Model Klasifikasi Varietas Tanaman Padi Menggunakan Metode CNN, berisi tentang penelitian terdahulu:

Penelitian terdahulu pertama dengan judul Klasifikasi Penyakit Mata Menggunakan *Convolutional neural network* dengan Arsitektur VGG-19. Berdasarkan hasil pengujian yang telah dilakukan dalam penelitian ini, dapat disimpulkan bahwa metode *Convolutional neural network* (CNN) dengan arsitektur VGG-19 efektif diterapkan untuk mengklasifikasi penyakit mata. Model ini berhasil mencapai nilai akurasi sebesar 65,29%, dengan *precision* sebesar 66,53%, *recall* sebesar 65,29%, dan *f1-score* sebesar 65,40%. Penggunaan *optimizer adagrad* pada arsitektur VGG-19 terbukti lebih baik dalam meningkatkan akurasi dibandingkan dengan *optimizer SGD*, yang hanya mencapai akurasi tertinggi sebesar 61,98%. Skenario kedua dalam penelitian ini menunjukkan hasil terbaik, di mana kombinasi VGG-19, *clipLimit=1.0*, *optimizer Adagrad*, dan 150 iterasi menghasilkan akurasi maksimum sebesar 65,29% (Marcella dkk. 2022).

Penelitian yang dilakukan oleh Ibrahim dkk. (2022) Klasifikasi Tingkat kematangan Pucuk Daun Teh menggunakan Metode CNN. Penelitian ini merancang sistem klasifikasi tingkat kematangan pucuk daun teh varietas *Assamica Klon GMB 7* dengan metode *Convolutional neural network* (CNN) menggunakan arsitektur VGGNet19 dan ResNet50. Hasil penelitian menunjukkan bahwa arsitektur VGGNet19 memberikan performa terbaik dengan akurasi mencapai 97,5%, *loss* 0,1208, dan *precision* 97%. Keberhasilan ini dicapai melalui penggunaan data augmentasi dengan parameter optimal, yaitu *optimizer RMSprop*, *learning rate* 0.01, *batch size* 32, dan *epoch* 100.

Selanjutnya yang dilakukan oleh Area, (2023) dengan judul Analisis Arsitektur Deep Learning Klasifikasi Jenis Jamur. Berdasarkan hasil analisis yang dilakukan dalam penelitian ini, dapat disimpulkan bahwa penggunaan arsitektur VGG-19 dalam klasifikasi jenis jamur menunjukkan hasil yang cukup baik. Dari 18

skenario model yang dilatih, terdapat 4 model dengan akurasi terbaik, di mana model skenario 15 menggunakan *hyperparameter* jumlah *epoch* 50, *batch size* 64, dan *optimizer* SGD mencapai akurasi 77,3%. Sementara itu, model skenario 16 dengan *hyperparameter* serupa tetapi menggunakan *optimizer* Adadelta mencatat akurasi 77,7%. Model skenario 17 dan 18 juga menunjukkan performa yang baik dengan akurasi masing-masing sebesar 77,3% dan 77,4%. Setelah pengujian menggunakan data testing dan evaluasi melalui *confusion matrix* serta *classification report*, model skenario 15 dinyatakan sebagai yang paling optimal dengan skor akurasi sebesar 70%.

Penelitian oleh Hernandoko dkk. (2024) dengan judul Penerapan Sistem Kontrol Kualitas Dengan Menggunakan Model CNN *Transfer Learning* VGG 19 Pada Inspeksi Kain Di Industri Tekstil. Penelitian ini menghasilkan sebuah model klasifikasi *defect* kain menggunakan CNN *transfer learning* VGG19 untuk otomatisasi sistem kontrol kualitas di industri tekstil. Model ini dirancang untuk mengklasifikasikan 6 jenis kondisi kain, yaitu cacat *double pick*, cacat jarang, cacat netting, cacat tepi masuk, cacat lainnya, dan produk baik. Dalam proses pengembangan model, dataset dibagi menjadi tiga bagian: 80% untuk *training set*, 10% untuk *validation set*, dan 10% untuk *testing set*. Proses *training* dan *validation* menunjukkan bahwa model mampu mengenali jenis *defect* kain grey dengan baik, tanpa menunjukkan tanda-tanda *overfitting* atau *underfitting*. Hal ini dapat disimpulkan bahwa model VGG19 yang dikembangkan memiliki generalisasi yang baik dan dapat diaplikasikan pada data baru.

Penelitian yang dilakukan oleh Firmansyah dkk. (2026) berjudul Perbandingan Strategi *Fine-Tuned* dan *Pre-Trained EfficientNet-B0* untuk Klasifikasi Penyakit Daun Padi. Penelitian ini membahas penerapan arsitektur *EfficientNet-B0* untuk mengklasifikasikan enam jenis penyakit daun padi dengan membandingkan strategi *pre-trained* dan *fine-tuned*. Permasalahan yang diangkat adalah tingginya tingkat kesalahan identifikasi penyakit daun padi secara manual yang dapat memengaruhi produktivitas hasil panen. Penelitian menggunakan 4.770 citra yang dibagi menjadi 3.333 data pelatihan, 958 data validasi, dan 479 data pengujian. Hasil penelitian menunjukkan bahwa strategi *fine-tuned* *EfficientNet-B0* menghasilkan performa yang lebih baik dibandingkan strategi *pre-trained*, sehingga

dapat meningkatkan kemampuan model dalam mengenali karakteristik penyakit daun padi secara lebih akurat.

Penelitian yang dilakukan oleh Paemo dkk. (2026) berjudul Klasifikasi Penyakit Daun Padi Berbasis Transfer *Learning Convolutional Neural Network* Menggunakan *MobileNetV2* dan *EfficientNetB0*. Permasalahan yang diangkat adalah proses identifikasi penyakit daun padi yang masih dilakukan secara visual sehingga memerlukan waktu, bergantung pada pengalaman petani, dan berpotensi menimbulkan kesalahan identifikasi. Dataset yang digunakan berjumlah 2.756 citra daun padi yang terdiri dari enam kategori, dengan pembagian 1.683 data pelatihan, 690 data validasi, dan 383 data pengujian. Hasil penelitian menunjukkan bahwa arsitektur *EfficientNetB0* memberikan performa terbaik dengan nilai akurasi sebesar 99,74%, precision 99,75%, recall 99,74%, dan F1-score 99,74%, sehingga dinilai mampu melakukan klasifikasi citra daun padi secara akurat dan efektif.

Penelitian yang dilakukan oleh Sapanca dkk. (2026) berjudul Klasifikasi Penyakit Daun Tomat Berbasis Transfer *Learning EfficientNetB0*. Permasalahan yang diangkat adalah proses identifikasi penyakit yang masih dilakukan secara manual sehingga membutuhkan waktu dan sangat bergantung pada pengalaman petani. Hasil evaluasi menunjukkan bahwa model dengan batch size 16 dan epoch 50 menghasilkan performa terbaik dengan *test accuracy* 92.55%, *precision* 92.82%, *recall* 92.55%, dan *F1-score* 92.39%. Performa per-kelas mencapai *Early Blight* (*F1-score* 92.46%), *Healthy* (*F1-score* 96.26% dengan *recall* sempurna 100%), dan *Late Blight* (*F1-score* 88.30%). Analisis komparatif membuktikan batch size 16 konsisten unggul dibanding *batch size* 32 dengan selisih akurasi 4.21%, dan peningkatan *epoch* dari 25 ke 50 memberikan kenaikan 2.10% terutama pada deteksi *Early Blight*. Model yang dikembangkan memenuhi standar aplikasi praktis dengan akurasi tinggi dan waktu training efisien, berpotensi membantu petani melakukan deteksi dini penyakit untuk meningkatkan produktivitas tanaman tomat.

Penelitian ini, para peneliti membandingkan performa model CNN menggunakan data asli dan data yang telah melalui *preprocessing Histogram Equalization* (HE). Peneliti menguji berbagai konfigurasi, termasuk rasio pembagian data *training- testing* yang berbeda dan ukuran *batch* yang bervariasi. Hasil penelitian menunjukkan bahwa model yang menggunakan data asli tanpa

preprocessing HE memberikan performa yang lebih baik, dengan akurasi mencapai 97%, presisi 100%, *recall* 97%, dan skor F1 98%.

Penelitian yang dilakukan sekarang adalah klasifikasi Varietas Tanaman Padi menggunakan metode *Convolutional neural network* (CNN) dengan menggunakan arsitektur *EfficientNetB0* untuk digunakan mengenali dan mengklasifikasi sebuah objek pada sebuah citra digital. Hasil dari penelitian ini adalah model Klasifikasi Varietas Tanaman Padi yang diharapkan dapat mencapai model akurasi yang tepat, efisiensi dan akurat.

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti	Metode	Temuan
1.	(Marcella dkk. 2022)	<i>Convolutional neural network</i> (CNN) dan Arsitektur VGG-19	Penelitian ini mengkaji klasifikasi penyakit mata menggunakan metode <i>Convolutional neural network</i> (CNN) dengan arsitektur VGG-19. Penulis menggunakan dataset yang terdiri dari 601 citra fundus mata, mencakup empat kelas penyakit: normal, katarak, glaukoma, dan penyakit retina. Proses <i>preprocessing</i> melibatkan metode CLAHE (<i>Contrast Limited Adaptive Histogram Equalization</i>) untuk meningkatkan kontras citra. Delapan skenario pengujian dilakukan dengan variasi optimizer (SGD (<i>Stochastic Gradient Descent</i>) dan Adagrad) dan

			nilai clipLimit. Hasil terbaik diperoleh menggunakan optimizer Adagrad dengan clipLimit 1.0, menghasilkan akurasi 65,29%, presisi 66,53%, recall 65,29%, dan f1-score 65,40%. Analisis per kelas menunjukkan kinerja terbaik pada klasifikasi katarak dengan akurasi 92%. Penelitian ini menyimpulkan bahwa metode CNN dengan arsitektur VGG-19 dapat diterapkan untuk klasifikasi penyakit mata, dengan optimizer Adagrad mengungguli SGD dalam meningkatkan akurasi
2.	(Ibrahim dkk. 2022)	<i>Convolutional neural network</i> (CNN) menggunakan Arsitektur VGGNET19 dan <i>ResNet50</i>	Menghasilkan sistem klasifikasi tingkat kematangan pucuk daun teh varietas Assamica Klon GMB 7 menggunakan metode <i>Convolutional neural network</i> (CNN) dengan arsitektur VGGNet19 dan <i>ResNet50</i>. Sistem ini mampu mengklasifikasikan pucuk daun teh ke dalam empat kelas berdasarkan usia: 7

			hari, 14 hari, 21 hari, dan 28 hari. Performa terbaik dicapai oleh arsitektur VGGNet19 dengan akurasi 97.5%, loss 0.1208, dan presisi 97%. Hasil optimal ini diperoleh menggunakan data augmentasi dengan parameter <i>Optimizer</i> RMSprop, <i>learning rate</i> 0.01, <i>batch size</i> 32, dan <i>epoch</i> 1001 Penulis menemukan bahwa VGGNet19 mengungguli ResNet50 dalam tugas Klasifikasi.
3.	(Area, 2023)	<i>Transfer learning</i> dengan arsitektur vgg-19	Peneliti melakukan eksperimen dengan arsitektur VGG-19 dengan 18 skenario model yang berbeda, dengan memvariasikan parameter seperti jumlah <i>epoch</i>, <i>batch size</i>, <i>optimizer</i>, dan <i>learning rate</i>. Dari berbagai skenario tersebut, diperoleh 4 model dengan performa terbaik. Model skenario 15 yang menggunakan 50 <i>epoch</i>, <i>batch size</i> 64, dan <i>optimizer</i> SGD mencapai akurasi 77.3% pada proses training.

			Tiga model lainnya dengan konfigurasi serupa juga menunjukkan akurasi di atas 77%. Setelah dilakukan pengujian menggunakan data testing dan evaluasi dengan <i>confusion matrix</i> serta <i>classification report</i>, model skenario 15 terbukti memiliki performa paling optimal dengan skor akurasi sebesar 70%.
4.	(Hernandoko dkk. 2024)	Model <i>Convolutional Neural Netowrk</i> (CNN), <i>Transfer Learning</i> dan VGG-19	Menghasilkan sebuah model klasifikasi defect kain menggunakan CNN <i>transfer learning</i> VGG19 untuk otomatisasi sistem kontrol kualitas di industri tekstil. Model ini dirancang untuk mengklasifikasikan 6 jenis kondisi kain, yaitu cacat <i>double pick</i>, cacat jarang, cacat netting, cacat tepi masuk, cacat lainnya, dan produk baik. Hasil penelitian menunjukkan bahwa model VGG19 yang dikembangkan memiliki performa yang baik, dengan akurasi testing mencapai 88%. Pada tahap evaluasi, model mencapai nilai presisi rata-rata 0.89,

			recall 0.88, dan F1-score 0.88 untuk semua kategori.
5.	Firmansyah dkk. (2026)	Menggunakan metode Transfer Learning Convolutional Neural Network (CNN) dengan arsitektur EfficientNetB0 dan MobileNetV2	Arsitektur EfficientNetB0 memberikan hasil terbaik dengan akurasi 99,74%, precision 99,75%, recall 99,74%, dan F1-score 99,74%, sehingga dinilai efektif dalam klasifikasi citra daun padi.
6.	Paemo dkk. (2026)	Transfer Learning dengan arsitektur EfficientNetB0	Hasil penelitian menunjukkan bahwa strategi fine-tuned EfficientNet-B0 memperoleh performa terbaik dengan akurasi 99,79% dan macro F1-score 0,9981, sedangkan strategi pre-trained hanya memperoleh akurasi 19,62%. Temuan ini menunjukkan bahwa proses fine-tuning mampu meningkatkan kemampuan EfficientNet-B0 dalam mengenali karakteristik penyakit daun padi.

7.	Sapanca dkk. (2026)	arsitektur <i>EfficientNetB0</i>	Hasil penelitian menunjukkan bahwa akurasi model meningkat dari 93,38% menjadi 96,80% setelah dilakukan optimasi. Penelitian menyimpulkan bahwa <i>EfficientNetB0</i> memiliki kemampuan klasifikasi yang tinggi dengan jumlah parameter yang relatif lebih sedikit sehingga efisien digunakan pada klasifikasi citra dengan karakteristik visual yang mirip.
8.	(Khusna Billa, 2025)	<i>Convolutional Neural Netowrk</i> (CNN) dan VGG-19	

2.2 Landasan Teori

Landasan teori merupakan teori-teori yang berhubungan atau berfungsi sebagai pedoman atau acuan agar peneliti tidak menyimpang dari tujuan awal yang telah ditetapkan

2.2.1 Padi Lokal

Padi lokal (*landrace*) merupakan padi primitif atau kultivar yang sudah berkembang selama bertahun-tahun atau bahkan berabad-abad dan dipengaruhi oleh migrasi dan seleksi, baik secara alami maupun buatan (Wahdah, 2021). Padi lokal yang memiliki sifat-sifat spesifik umumnya memiliki potensi hasil rendah, umur dalam, mudah rebah, dan kurang respons terhadap pemupukan. Namun Padi

lokal juga mempunyai keunggulan tertentu seperti tahan terhadap hama dan penyakit, rasa nasi enak, beradaptasi di lingkungan spesifik seperti genangan air yang cukup dalam, tanah kering, kadar garam dan kandungan besi (Fe) tinggi (Chairuman & Jonharnas, 2012).

Padi varietas lokal sebagai bagian sumber daya genetik merupakan sumber keragaman dan aset daerah yang perlu dilestarikan, dibudidayakan, dan dikembangkan. Mengingat dengan berkembangnya varietas unggul baru (VUB), plasma nutfah padi lokal semakin terdesak, sehingga banyak diantara padi lokal yang berpotensi sebagai sumber keragaman genetik menjadi punah (Wasito & Subagyono, 2012).

Di kampung Tapanuli Selatan, terdapat 4 jenis varietas tanaman padi yang sering ditanam oleh petani. Varietas-varietas ini dikenal karena kecocokannya terhadap kondisi lokal dan potensi hasil yang baik, serta memiliki karakteristik unik yang disukai oleh para petani di daerah tersebut. Berikut keempat varietas tanaman padi yang akan dilakukan Klasifikasi Varietas Tanaman Padi :

1. Kuku Balam

Padi lokal yang paling dikenal di Sumatera Utara adalah jenis Kuku Balam atau dikenal dengan beras KB (Kuku Balam). Padi jenis Kuku Balam ini banyak dijumpai di pusat-pusat perbelanjaan dan rumah makan, karena rasanya yang pulen sangat digemari konsumen (Chairuman & Jonharnas, 2012). Kuku Balam memiliki karakteristik unik, varietas jenis ini memiliki waktu penanaman antara 135 hingga 140 hari, dengan ciri fisik yang mencolok. Tinggi tanaman Kuku Balam mencapai sekitar 135 cm. Gabahnya memiliki bentuk ramping dan berwarna kuning cerah. Selain itu, warna kaki batangnya adalah hijau muda.



Gambar 2. 1 Tanaman Padi Kuku Balam
(Sumber: Foto Sendiri)

2. Sipera

Padi pera adalah padi dengan kadar amilosa pada pati lebih dari 20% pada berasnya, yaitu tidak lengket setelah dimasak. Padi ini ditanam di sawah kering dengan waktu penanaman sekitar 90–100 hari. Dengan masa panen yang relatif singkat, Sipera menjadi salah satu varietas yang cukup menarik bagi petani. Meski nasi yang dihasilkan dari padi ini tidak se-pulen jenis padi lain seperti Kuku Balam, Sipera tetap memiliki tempat di pasaran karena harganya yang cukup tinggi. Karakteristik utama padi Sipera adalah kemampuannya tumbuh optimal di lahan kering, sehingga sangat adaptif untuk daerah seperti Tapanuli Selatan. Waktu panen padi ini berkisar antara 90 hingga 100 hari. Padi Sipera ini cocok untuk hidangan tertentu, seperti nasi goreng atau makanan berkuah, yang lebih baik menggunakan nasi bertekstur pera.



Gambar 2. 2 Tanaman Padi Sipera
(Sumber: Foto Sendiri)

3. Sigundang

Padi Sigundang termasuk dalam kelompok tanaman yang relatif rendah, dengan tinggi tanaman sekitar 150 cm. Varietas ini dikenal memiliki hasil yang stabil dan tinggi, dengan rata-rata hasil mencapai 5,51 ton per hektar di beberapa lokasi. Keunggulan varietas Sigundang terletak pada ketahanannya terhadap serangan hama dan penyakit, serta memiliki rasa nasi yang sangat pulen (Chairuman & Jonharnas, 2012). Hal ini menjadikan Sigundang sebagai pilihan favorit di kalangan petani, dengan luas penanaman yang meningkat dari 2.261 hektar pada tahun 1996 menjadi 3.276 hektar pada tahun 2010 (Ginting dkk. 2011).



Gambar 2. 3 Tanaman Padi Sigundang

(Sumber: Foto Sendiri)

4. Sihasona

Padi Sihasona adalah jenis padi lokal yang dikenal di daerah Parau Sorat. Padi ini memiliki waktu penanaman sekitar 3,5 bulan (105 hari), menjadikannya termasuk varietas dengan masa panen menengah. Meskipun nama ini belum banyak dikenal secara luas atau terdokumentasi secara resmi seperti jenis padi lainnya, padi sihasona memiliki karakteristik tersendiri yang membedakannya. Padi ini ditanam di lahan kering, sehingga cocok untuk wilayah dengan sumber irigasi terbatas atau dengan pola tanam lahan tadah hujan. Karakteristik utama padi Sihasona adalah rasa nasinya, yang berada di antara nasi pulen dan pera. Dibandingkan dengan Kuku Balam, rasa nasi Sihasona tidak se-enak padi unggulan tersebut, namun lebih baik dibandingkan padi pera.

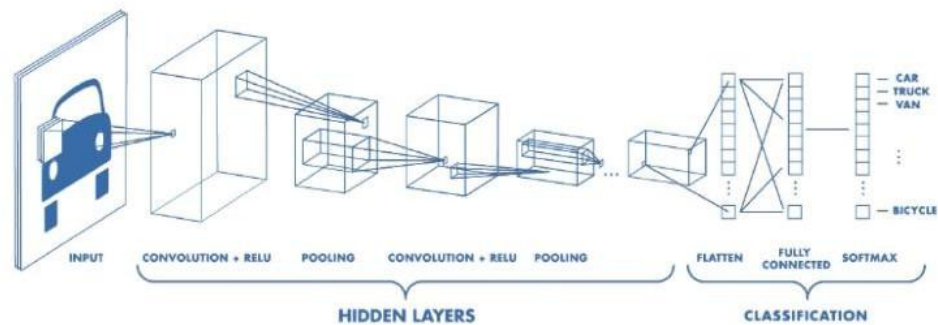


Gambar 2. 4 Tanaman Padi Sihasona
(Sumber: Foto Sendiri)

2.2.2 *Convolutional Neural Network (CNN)*

Convolutional neural network (CNN) merupakan bagian dari *Deep Neural Network*, yakni jenis jaringan saraf tiruan yang umumnya digunakan dalam pengenalan dan pemrosesan gambar (Trivusi, 2022). Pada umumnya CNN memiliki tiga lapisan utama. Masing-masing lapisan memiliki tugas yang berbeda. Pada lapisan pertama yang disebut sebagai *Convolutional Layer*, ini merupakan lapisan

terluar yang bertugas untuk mengidentifikasi gambar dan suara, seperti warna dan tepi. Kemudian ada lapisan kedua yaitu *Poolinglayer* berfungsi untuk mengurangi *overfitting* pada model. Pada lapisan ketiga/terakhir, terdapat *Fully Connected* merupakan lapisan paling kompleks dalam CNN. Lapisan ini bertugas untuk mengenali objek dan bentuk secara keseluruhan, sehingga dapat menentukan objek yang akan diklasifikasi berdasarkan fitur-fitur yang telah di ekstrak sebelumnya (Lestari, 2022).



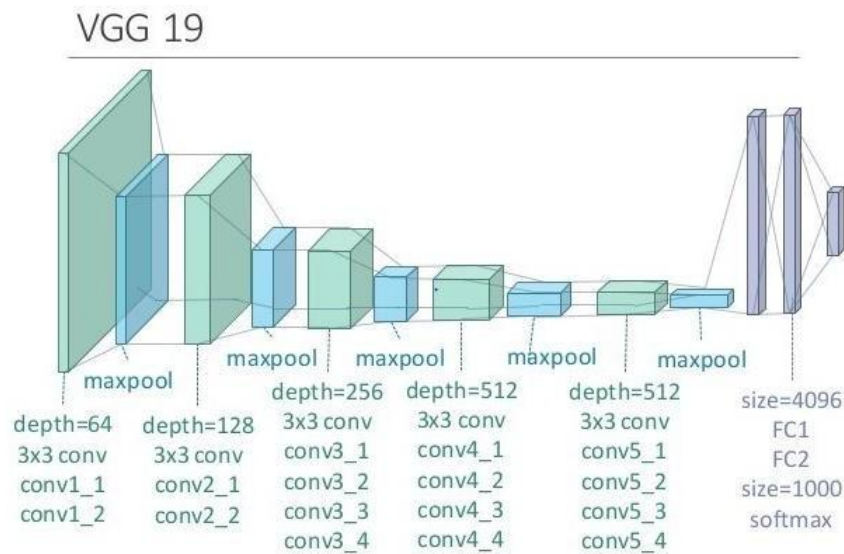
Gambar 2. 5 Algoritma CNN
(Sumber: Google.com)

2.2.3 Arsitektur Visual Geometry Group (VGG-19)

VGG-19 adalah jaringan *Convolutional* mendalam dengan 19 lapisan, yang terdiri dari 16 lapisan Convolutional layer dan 3 lapisan *fully connected*. Kedalaman ini memungkinkan model untuk mengekstraksi fitur yang lebih kompleks dan halus dari gambar, sehingga akan menghasilkan akurasi yang lebih tinggi dalam pengenalan objek pemrograman (Matlab, 2023). VGG-19 menggunakan filter konvolusi kecil berukuran 3x3 secara konsisten di seluruh lapisan konvolusinya (Setiawan, 2020). Penggunaan filter kecil ini membantu dalam mempertahankan informasi spasial dan meningkatkan kemampuan model untuk menangkap detail halus dalam gambar. Pada Gambar 2.2 merupakan struktur umum VGG-19 yang terdiri dari urutan convolution layers berikut penjelasan gambarnya :

1. *Block 1 : 2x Convolutional Layer 3x3 (64) + ReLU, Max Pooling*
2. *Block 2 : 2x Convolutional Layer 3x3 (128) + ReLU, Max Pooling*
3. *Block 3 : 4x Convolutional Layer 3x3 (256)+ ReLU, Max Pooling*
4. *Block 4 : 4x Convolutional Layer 3x3 (512) + ReLU, Max Pooling*
5. *Block 5 : 4x Convolutional Layer 3x3 (512) + ReLU, Max Pooling*

Fully Connected Layers1 (4096), Fully Connected Layers2 (4096), Fully Connected Layer (1000) dan softmax



Gambar 2. 6 Arsitektur VGG -19
(Sumber: Adi Pamungkas, 2023)

2.2.4 EfficientNetB0

EfficientNetB0 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan untuk meningkatkan akurasi klasifikasi citra dengan tetap mempertahankan efisiensi komputasi. Menurut Sapanca dkk. (2026) *EfficientNetB0* adalah varian dasar *EfficientNet* yang dirancang dengan *Neural Architecture Search* (NAS) dan *compound scaling*, sehingga menskalakan *depth*, *width*, dan *resolution* secara seimbang. Arsitektur ini memiliki 18 layer dalam 9 *stage*, menggabungkan *standard convolution* di awal, MB Conv sebagai blok utama, serta *global average pooling* dan *fully connected* layer untuk klasifikasi. *MBConv* menggunakan *inverted residual*, *depthwise convolution*, *squeeze and excitation*, dan aktivasi *Swish*, serta *shortcut connection* dan *stochastic depth* sebagai regularisasi. Model ini efisien dengan 5,3 juta parameter dan 0,39 miliar FLOPS, cocok untuk perangkat dengan sumber daya terbatas.



Gambar 2. 7 Arsitektur *EfficientNetB0*
(Sumber: Sapanca dkk. 2026)

2.2.5 Hugging Face Spaces

Hugging Face Spaces merupakan layanan berbasis *cloud* yang disediakan oleh *Hugging Face* untuk membangun, menjalankan, dan membagikan aplikasi berbasis *Machine Learning* maupun *Deep Learning* secara daring. Platform ini memungkinkan pengembang mengintegrasikan model yang telah dilatih ke dalam sebuah aplikasi *web* tanpa harus melakukan konfigurasi server secara manual. *Hugging Face Spaces* mendukung beberapa *framework* pengembangan, seperti *Gradio*, *Streamlit*, dan *Docker*, sehingga memudahkan proses implementasi berbagai model kecerdasan buatan. Selain itu, setiap perubahan kode yang dikirim ke repositori akan diproses secara otomatis melalui mekanisme *build* sehingga aplikasi yang telah *deploy* dapat diperbarui tanpa proses konfigurasi yang rumit.

Menurut Jones dkk. (2024), *Hugging Face* telah berkembang menjadi salah satu ekosistem terbesar dalam pengembangan dan distribusi model *deep learning*. Platform ini tidak hanya berfungsi sebagai repositori model, tetapi juga menyediakan sarana kolaborasi bagi pengembang untuk membagikan model, dataset, serta aplikasi berbasis kecerdasan buatan. Keberadaan *Hugging Face Spaces* mendukung proses implementasi model agar dapat diakses secara langsung melalui *web* sehingga mempermudah proses demonstrasi maupun pengujian model oleh pengguna.

Pada penelitian ini, *Hugging Face Spaces* digunakan sebagai media *deployment* model *EfficientNetB0* yang telah melalui proses pelatihan

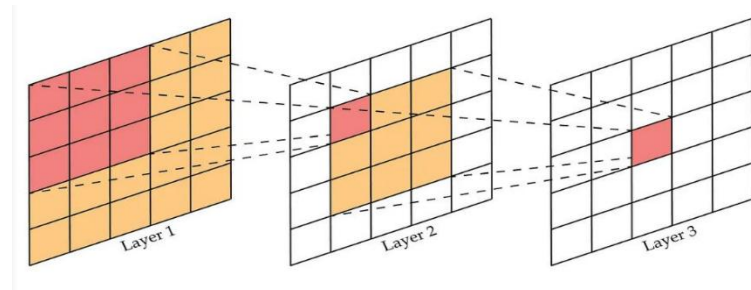
menggunakan *PyTorch*. Model yang telah disimpan dalam format *.pth* diintegrasikan ke dalam aplikasi berbasis Flask, kemudian dideploy ke *Hugging Face Spaces* sehingga dapat diakses melalui peramban web. Dengan demikian, pengguna cukup mengunggah citra tanaman padi melalui antarmuka aplikasi, sedangkan proses klasifikasi dilakukan secara otomatis menggunakan model yang telah tersedia pada server *Hugging Face Spaces*.

2.2.6 Feature Learning

Feature Learning adalah proses dalam *machine learning* dan *deep learning* di mana model secara otomatis mengekstrak fitur dari data mentah, seperti gambar, teks, atau suara, tanpa memerlukan rekayasa fitur manual. Proses ini terdiri dari beberapa layer yang saling bekerjasama untuk mengambil ciri dari sebuah gambar. Berikut ini adalah penjelasan dari setiap layer yang terdapat pada *feature learning*.

1. Convolutional Layer

Convolutional layers berfungsi untuk mengekstraksi fitur dari gambar input tanaman padi. Setiap lapisan konvolusi menerapkan filter (kernel) pada citra untuk menghasilkan *feature map* yang menangkap informasi penting seperti tepi, pola, dan tekstur.

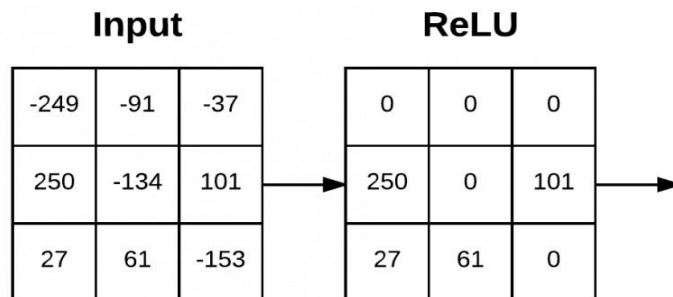


Gambar 2. 8 *Convolutional Layer*
(Sumber: Reza Kalantar, 2023)

2. Rectified Linear Unit (ReLU)

Setelah melakukan *Convolutional Layers* selanjutnya ada fungsi ReLU (*Rectified Linear Unit*) adalah aktivasi yang paling banyak digunakan dalam jaringan saraf, terutama dalam konteks *Deep Learning* dan *Convolutional neural networks* (CNN). ReLU diperkenalkan untuk mengatasi beberapa masalah yang dihadapi oleh fungsi aktivasi sebelumnya, seperti sigmoid dan tanh. Fungsi aktivasi

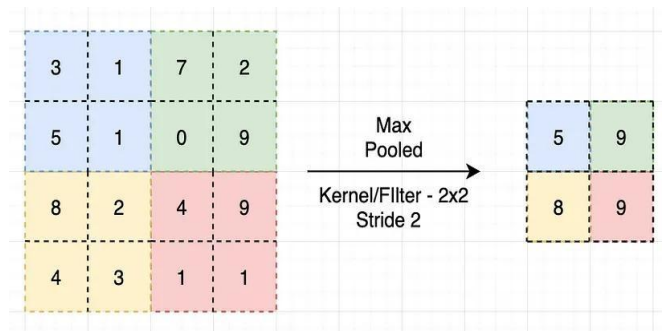
ReLU didefinisikan sebagai : $f(x) = \max(0, x)$. Artinya jika nilai input (x) positif, fungsi mengembalikan nilai input itu sendiri, sedangkan jika nilai input negatif atau nol, fungsi mengembalikan 0 (Martech Zone, 2024).



Gambar 2. 9 ReLU
(Sumber: Pyimagesearch, 2021)

3. Pooling Layer

Pooling layer berfungsi untuk mengurangi ukuran *spatial* dari *feature map*. Jenis *pooling* yang digunakan yaitu *max pooling*. Dengan mengambil nilai maksimum dari setiap jendela yang ditentukan. Misalnya, jika kita menggunakan jendela berukuran 2x2, *max pooling* akan melihat setiap blok 2x2 dari *feature map* dan memilih nilai maksimum dari blok tersebut untuk dimasukkan ke dalam *output*.



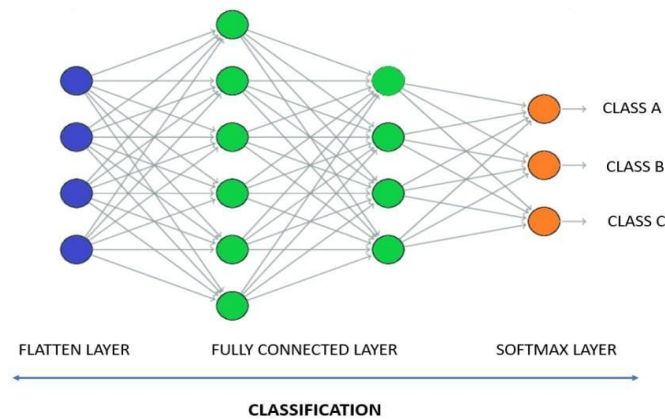
Gambar 2. 10 Max Pooling
(Sumber: Kartikeya Rana, 2020)

2.2.7 Classification

Proses pengelompokan data ke dalam kategori atau kelas yang telah ditentukan. Model *machine learning* dilatih menggunakan dataset yang memiliki label, dan setelah pelatihan, model dapat digunakan untuk mengklasifikasikan data baru yang belum diberi label.

1. Fletten

Flatten adalah proses yang digunakan dalam arsitektur jaringan saraf, khususnya dalam *Convolutional neural networks* (CNN), untuk mengubah output dari lapisan konvolusi atau *pooling* menjadi vektor satu dimensi. Proses ini penting karena input untuk *fully connected layer* harus dalam bentuk vektor, bukan matriks multidimensional. Sehingga, Sebelum masuk pada tahap *Fully Connected Layer*, *Feature Map* tersebut akan melalui proses “*flatten*” atau *reshape* (Nugroho dkk. 2020).



Gambar 2. 11 *Classification Layer*
(Sumber: Indian Tech Warrior, 2021)

2. *Fully connected layer*

Fully connected layer, juga dikenal sebagai *dense layer*, adalah lapisan pada jaringan saraf, yakni setiap neuron terhubung dengan setiap neuron dalam lapisan sebelumnya dan sesudahnya. Lapisan ini paling umum digunakan dalam jaringan saraf *multi-layer perceptron* (MLP) untuk mempelajari hubungan kompleks antara fitur-fitur input (dicoding, 2024).

3. *Softmax*

Fungsi ini mengubah nilai skor logit yang dihasilkan oleh lapisan terakhir (*fully connected layer*) menjadi probabilitas yang dapat diinterpretasikan, yaitu nilai antara 0 dan 1, di mana jumlah seluruh probabilitas adalah 1. Probabilitas ini digunakan untuk menentukan prediksi akhir pada tugas klasifikasi gambar tanaman padi.

2.2.8 *Software*

Software yang akan digunakan dalam penelitian untuk mengimplementasikan model klasifikasi citra tanaman padi menggunakan metode *Convolutional neural network* (CNN) dengan arsitektur VGG-19 adalah sebagai berikut:

1. *Python*

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk memudahkan pengembangan perangkat lunak dengan sintaksis yang sederhana dan mudah dipahami. *Python* pertama kali dikembangkan oleh Guido van Rossum pada tahun 1991. *Python* memiliki berbagai kelebihan yang membuatnya menjadi salah satu bahasa pemrograman paling populer di dunia, terutama dalam bidang pengembangan perangkat lunak, data science, dan machine learning (Dicoding, 2023). Selain itu, ekosistem pustaka yang kaya menjadi alasan lain mengapa *Python* sangat populer. Pustaka-pustaka seperti NumPy, Pandas, Matplotlib, TensorFlow, dan Keras memungkinkan pengembang untuk menangani berbagai tugas, mulai dari analisis data hingga pengembangan model deep learning, tanpa harus menulis kode dari awal. *Python* juga dikenal dengan komunitas yang aktif dan dukungan yang luas, sehingga pengembang dengan mudah menemukan solusi atas masalah yang dihadapi melalui forum atau dokumentasi.

2. *TensorFlow*

TensorFlow adalah salah satu *framework machine learning* yang sangat populer, yang dikembangkan oleh Google. *TensorFlow* Pertama kali dirilis pada tahun 2015, *framework* ini dengan cepat menjadi alat yang banyak digunakan untuk membangun, melatih, dan mengevaluasi model *machine learning* dan *deep learning* (fazry, 2024). *TensorFlow* dirancang untuk memberikan fleksibilitas dan skalabilitas tinggi, memungkinkan pengembang untuk menangani berbagai tugas *machine learning* dengan efisien, mulai dari pemrosesan data hingga implementasi model *deep learning* yang kompleks. *Framework* ini menjadi yang paling banyak digunakan dalam komunitas global, berkat dukungan terhadap berbagai aplikasi seperti pemrosesan bahasa alami (NLP), pengenalan gambar, dan sistem rekomendasi (Callaghan dkk. 2017) .

3. Keras

Keras adalah sebuah pustaka *open-source* yang ditulis dalam *Python* dan dirancang untuk memudahkan pengembangan dan pelatihan model *deep learning*. Keras berfungsi sebagai API tingkat tinggi yang dapat berjalan di atas beberapa *backend* seperti *TensorFlow*, CNTK (*Microsoft Cognitive Toolkit*) , atau Theano (Evelyn Clarke, 2024). Keras menawarkan kemudahan dalam membangun,

melatih, dan menyebarkan model jaringan saraf. Salah satu keunggulan utama dari Keras adalah kemampuannya untuk mengarahkan proses pembuatan model dengan menyediakan antarmuka yang pencitraan dan modular. Hal ini memungkinkan pengguna untuk dengan cepat mendefinisikan model arsitektur, menambahkan lapisan-lapisan yang diperlukan, serta mengkompilasi model dengan fungsi *loss* dan *optimizer* yang sesuai.

2.2.9 Confusion Matrix

Confusion Matrix merupakan alat evaluasi yang sangat penting dalam mengukur kinerja model klasifikasi dalam *machine learning*. Alat ini memungkinkan peneliti untuk membandingkan prediksi model dengan nilai sebenarnya, memberikan wawasan mendalam tentang akurasi dan jenis kesalahan yang dilakukan oleh model (Gemilang, 2016). *Confusion Matrix* terdiri dari empat komponen utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), yang memungkinkan perhitungan berbagai metrik evaluasi seperti akurasi, presisi, *recall*, dan F1- score.

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

Gambar 2. 12 *Confusion Matrix*
(Sumber: ml-sciensce.com)

1. *True Positive* (TP): Kasus dimana model memprediksi Positif dan aktualnya memang Positif
2. *False Positive* (FP): Kasus dimana model memprediksi Positif tetapi aktualnya Negatif
3. *False Negative* (FN): Kasus dimana model memprediksi Negatif tetapi aktualnya Positif
4. *True Negative* (TN): Kasus dimana model memprediksi Negatif dan aktualnya memang Negatif

2.2.10 Evaluation

Tahap analisis performa model dilakukan dengan mengevaluasi hasil *confusion matrix* yang telah diperoleh. Pengukuran kinerja model dapat dilihat melalui beberapa parameter penting yang menunjukkan seberapa baik model tersebut bekerja.

1. *Accuracy* (ketepatan) merupakan parameter yang mengukur seberapa tepat model dalam melakukan prediksi secara keseluruhan. Parameter ini menghitung perbandingan antara jumlah prediksi yang benar (baik positif maupun negatif) dengan total keseluruhan data. Semakin tinggi nilai *accuracy*, semakin baik model dalam melakukan klasifikasi. Nilai *accuracy* dihitung menggunakan persamaan.

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

2. *Precision* (presisi) fokus pada mengukur ketepatan model dalam memprediksi kasus positif. Parameter ini menghitung proporsi dari kasus yang diprediksi positif yang benar-benar merupakan kasus positif. *Precision* sangat penting ketika kita ingin menghindari false positive, yaitu situasi di mana model salah mengidentifikasi kasus negatif sebagai positif. Selanjutnya, tingkat *precision* (presisi) digunakan untuk melihat seberapa akurat model dalam mengidentifikasi kasus positif dari seluruh prediksi positif yang dilakukan. Nilai *Precision* dihitung menggunakan persamaan.

$$Precision = \frac{TP}{TP + FP}$$

3. Kemampuan model dalam mengenali data positif dapat diukur melalui nilai sensitivitas (*recall/sensitivity*), yang menunjukkan proporsi kasus positif yang berhasil diidentifikasi dengan benar. Nilai *recall* dihitung menggunakan persamaan.

$$Recall = \frac{TP}{TP + FN}$$

4. *F1-Score* memberikan nilai tunggal yang menyeimbangkan antara precision dan *recall*. Parameter ini sangat berguna ketika kita membutuhkan keseimbangan antara precision dan recall, terutama pada dataset yang tidak seimbang. *F1-Score* merupakan rata-rata harmonik yang memberikan bobot yang sama antara precision dan *recall*. Nilai *F1-Score* dapat dihitung menggunakan persamaan.

$$F1 - Score = 2 \left(\frac{Precision * recall}{Precision + recall} \right)$$