

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berdasarkan penelitian sebelumnya, terdapat beberapa penelitian terdahulu yang telah menjadi rujukan dan tumpuan dalam mengimplementasikan proses penelitian, Berikut beberapa penelitian terdahulu, dapat dilihat pada tabel 2.1 dibawah ini:

Tabel 2. 1 Penelitian Terdahulu

No.	Penulis	Judul	Metode	Hasil
1.	M. Bijoy, N. Hasan, M. Biswas <i>et al.</i> [13]	Towards Sustainable Agriculture: A Novel Approach for Rice Leaf Disease Detection Using dCNN and Enhanced Dataset	Lightweight deep Conolucional Neural Network (Dcnn), dan 21 arsitektur benchmark seperti AlexNet, MobileNet, ResNet50, DenseNet121, ResNeXt50, ShuffleNet, ConvNext, EfficientNet, GogoleNet, SwinTransformer, VisionTransforme r, dan MaxVit	Hasil penelitian dengan penggambaran dua dataset yang ada dan tambahan 95 gambar beranotasi manual yang dikumpulkan dari sumber internet, menghasilkan akurasi pengujian pada <i>MobileNetV2</i> : 99,712 dan dCNN 99,808. serta pada penelitian ini, pengembangan sistem API terbuka untuk anotasi otomatis kejadian baru.
2.	M. M. Hasan <i>et al.</i> [16]	Enhancing Rice Crop Management: Disease Classification Using Convolutional	MobileNet, Inception-V3, AlexNet, dan new model CNN small size	Hasil penelitian dengan 120 data penyakit daun padi dengan 3 kelas penyakit. Menghasilkan akurasi pengujian

		Neural Networks and Mobile Application Integration		<p>MobileNet: 94,79% (10 epoch) dan CNN: 90,24% (150 epoch)</p> <p>Dengan masing-masing waktu pemrosesan tercepat diraih oleh New Model CNN selama 103 ms, sedangkan <i>MobileNetV2</i> berhasil meraih posisi kedua dengan 145 ms lebih cepat dibandingkan ke dua model <i>transfer learning</i> lainnya.</p> <p>Serta pada penelitian ini, model yang telah dibangun berhasil di implementasikan ke dalam aplikasi Android.</p>
3.	M. Dutta <i>et al.</i> [15]	Rice Leaf Disease Classification —A Comparative Approach Using Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-	CNN, Cascading Autoencoder dengan Attention Residual U-Net (CAAR-U-Net), dan <i>MobileNetV2</i>	<p>Hasil penelitian dengan 5932 data penyakit daun padi dengan 4 kelas penyakit yaitu Bacterial Blight 1584 Citra, Blast 1440 Citra, Brown Spot 1600 Citra, dan Tungro 1308 Citra.</p> <p>Menghasilkan akurasi pengujian, CNN: 98%</p>

		Net (CAAR-U-Net), and MobileNet-V2 Architecture		CCAR-U-Net: 95.41% dan <i>MobileNetV2</i> : 87.64%.
4.	I. F. Annur, J. Umami, M. N. Annafii, N. Trisnaningrum, and O. V. Putra. [19]	Klasifikasi Tingkat Keparahan Penyakit Leafblast Tanaman Padi Menggunakan <i>MobileNetV2</i>	CNN dengan arsitektur <i>MobileNetV2</i> yang menerapkan <i>transfer learning</i>	Hasil penelitian dengan 300 data dengan 1 jenis penyakit serta 3 kelas yaitu sehat-ringan, sedang, dan berat. Menghasilkan akurasi pengujian, rata-rata sebesar 78,33%
5.	Y. AUFAR and T. P. Kaloka. [18]	Robusta coffee leaf diseases detection based on <i>MobileNetV2</i> model	<i>MobileNetV2</i> DenseNet169 ResNet50 InceptionResNet V2	Hasil penelitian dengan menggunakan 1560 data penyakit daun kopi dengan 5 kelas penyakit serta 1 kelas sehat. Menghasilkan akurasi pengujian <i>MobileNetV2</i> : 99.93 % DenseNet169: 99.74% ResNet50: 99.41% InceptionResNetV2: 99.09% Model <i>MobileNetV2</i> memiliki waktu eksekusi tercepat dan ukuran model terkecil, menjadikannya model paling efisien untuk implementasi di

				perangkat mobile/web.
--	--	--	--	-----------------------

2.2 Daun Padi

Padi merupakan tanaman yang sangat signifikan pada pembudidayaan di peradaban sekarang, temuan dari pengelolaan padi disebut beras[22]. Padi diduga berasal dari India atau Indocina dan masuk ke Indonesia melalui nenek moyang yang bermigrasi dari daratan Asia pada 1500 SM[22]. Struktur tanaman padi yakni memiliki batang yang bulat dan berongga, daun yang pipih, dan malai di bagian atas tanaman[23]. Organ vegetative tanaman padi terdiri dari: akar, batang, daun, dan organ reproduksi dan malai yang terdiri dari bulir-bulir padi[23].

Daun padi tumbuh secara bergantian pada batangnya, dengan satu daun per ruas, daun terakhir yang membungkus malai disebut daun malai atau daun bendera[23]. Daun adalah mesin pertumbuhan tanaman padi, karena daun menangkap radiasi matahari dan menghasilkan karbohidrat, tanaman padi bisa bernafas dan mengeluarkan keringat melalui daunnya[23]. Arsitektur daun padi bisa tegak, miring, atau terkulai hal itu tergantung pada varietasnya dan daun padi merupakan faktor krusial dalam kemampuan menangkap radiasi matahari.

2.3 Penyakit Daun Padi

2.3.1 Bacterial Leaf Blight

Bacterial leaf blight adalah penyakit yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae*. Penyakit ini menyebabkan layu pada bibit dan kekuningan serta pengeringan daun. Penyakit ini dapat terjadi di lingkungan tropis maupun subtropic, terutama di daerah dataran rendah yang diirigasi dan bergantung pada hujan. Penyakit ini sering terjadi saat angin kencang dan hujan deras yang terus-menerus, yang memungkinkan bakteri penyebab penyakit menyebar dengan mudah melalui tetesan lender pada luka tanaman yang terinfeksi[24].

2.3.2 Brown Spot

Brown spot merupakan salah satu penyakit padi yang paling umum dan paling merusak tanaman padi. *Brown spot* adalah penyakit jamur yang menyerang

coleoptile, daun, pelepah daun, cabang malai, *glume*, dan *spikelet*. Kerusakan yang paling terlihat adalah bitnik-bintik besar yang banyak terdapat pada daun, yang dapat membunuh seluruh daun. Penyakit ini dapat berkembang di daerah dengan kelembapan relatif tinggi (86-100%) dan suhu antara 16 dan 36 derajat. Penyakit ini umum terjadi di tanah yang tidak tergenang air dan kekurangan nutrisi atau di tanah yang menumpuk zat beracun[24].

2.3.3 Leaf Blast

Leaf Blast disebabkan oleh jamur *Magnaporthe oryzae*. Penyakit ini dapat menyerang semua bagian tanaman padi di atas tanah seperti daun, leher batang, ruas leher, bagian malai, dan terkadang pelepah daun. Penyakit ini terjadi di daerah dengan kelembapan tanah rendah, hujan yang sering dan berkepanjangan, serta suhu siang hari yang sejuk[24].

2.3.4 Leaf Scald

Penyakit *leaf scald* adalah penyakit jamur yang disebabkan oleh *Microdochium oryzae*, yang menyebabkan daun tampak terbakar. Perkembangan penyakit biasanya terjadi pada akhir musim pada daun yang sudah matang dan dipicu oleh cuaca basah, pemupukan nitrogen yang tinggi, dan jarak tanam yang rapat. Penyakit ini berkembang lebih cepat pada daun yang terluka daripada yang tidak terluka[24].

2.3.5 Narrow Brown Leaf Spot

Penyakit *Narrow brown leaf* disebabkan oleh jamur *Sphaerulina oryzina* yang dapat menginfeksi daun, pelepah, dan malai. Penyakit ini menyebabkan kematian dini daun dan pelepah daun, pematangan dini biji padi dan dalam kasus yang parah, rebah tanaman. Penyakit ini biasanya terjadi di tanah yang kekurangan kalium, dan di daerah dengan suhu 25-28 derajat. Penyakit ini muncul pada tahap pertumbuhan akhir tanaman padi, mulai dari tahap pembungaan[24].

2.3.6 Rice Hispa

Dicladispa armigera, yang dikenal sebagai hama padi (*rice hispa*), merupakan salah satu penyakit yang menyerang tanaman padi, khususnya selama fase pertumbuhan anakanan. Pada tahap ini, populasi penyakit cenderung meningkat. Penyakit ini merusak daun dengan cara memakan lapisan atas kulit ari daun[25].

2.3.7 Sheath Blight

Penyakit *Sheath Blight* adalah penyakit jamur yang disebabkan oleh *Rhizoctonia solani*. daun yang terinfeksi akan layu, mengering, dan mati lebih cepat, sementara tunas muda juga dapat hancur. Penyakit *sheath blight* terjadi di daerah dengan suhu tinggi (28-32 derajat) tingkat pupuk nitrogen yang tinggi, dan kelembapan relatif kanopi tanaman antara 85-100%. Tanaman lebih rentan terhadap penyakit sheath blight selama musim hujan[24].

2.4 Pre-processing

Pre-processing merupakan tahapan awal dalam pengembangan suatu model, seperti pada penelitian-penelitian sebelumnya tentang klasifikasi penyakit daun padi menggunakan *deep learning* atau *Convolutional Neural Network* (CNN) dan variasinya seperti *MobileNet*[10][14][16][19]. Tujuan utama dari tahapan ini adalah untuk meningkatkan kualitas citra, menyeragamkan data, dan menyiapkan citra agar dapat diproses dengan mudah dan efektif oleh model[10][14]. Proses *pre-processing* ini juga membantu dalam memperjelas fitur-fitur penting dalam gambar yang akan dianalisis oleh model[10].

Tahap *pre-processing* yang pertama dimulai dengan pembersihan dataset citra, seperti mengubah ukuran (*Resize*) yang bertujuan untuk menyeragamkan ukuran semua citra menjadi dimensi yang konsisten, seperti pada penelitian[10] yang mengatur default ukuran gambar dalam bentuk *tuple* dengan nilai 256x256 piksel dan ukuran 224x224 piksel pada penelitian[14][19]. Lalu terdapat penghapusan latar belakang dan area tidak perlu (*segmentasi*), teknik ini digunakan untuk menghapus latar belakang atau area daun yang sehat yang tidak menunjukkan gejala penyakit, sehingga model

dapat fokus pada bagian yang terinfeksi[16]. Pada penelitian ini beberapa metode segmentasi yang akan digunakan yaitu *HSV Thresholding* dan *GrabCut*.

Penelitian [16] secara spesifik menyebutkan penggunaan teknik *HSV thresholding* untuk teknik penghapusan latar belakang dan area yang tidak diperlukan (segmentasi) ini, penelitian [16] menggunakan model transformasi warna dalam pemrosesan gambar (*image processing*) berdasarkan persamaan (*equation*) $g(x, y) = T[f(x, y)]$, yang dimana, $f(x, y)$ merupakan citra warna masukan (*input color image*), lalu $g(x, y)$ adalah citra warna keluaran (*output color image*). Setelah itu penggunaan teknik segmentasi lainnya seperti pada penelitian [20] yang menggunakan metode *GrabCut* untuk memisahkan objek daun padi, *GrabCut* bekerja dengan prinsip *iterative graph cut*, di mana citra direpresentasikan dalam bentuk *graf* yang terdiri dari *node* (piksel) dan *edge* (hubungan antar piksel), algoritma ini memanfaatkan model *Gaussian Mixture Model* (GMM) untuk mempelajari distribusi warna antar latar depan (*foreground*) dan latar belakang (*background*) [26].

2.5 Transfer learning

Transfer learning merupakan metode dalam pembelajaran mesin yang memanfaatkan model yang sudah dilatih sebelumnya untuk menyelesaikan permasalahan baru yang serupa[27]. Teknik ini tidak memulai pelatihan dari nol, melainkan memanfaatkan pengetahuan yang telah diperoleh dari tugas sebelumnya[27]. Dalam bidang pengenalan citra, *transfer learning* sangat populer karena jaringan saraf tiruan secara bertahap mempelajari fitur seperti tepi, bentuk, hingga pola yang lebih kompleks[27]. Dengan menggunakan *transfer learning*, lapisan awal dari model prelatih yang mampu mendeteksi tepi dan bentuk dapat dipertahankan, sementara lapisan akhir disesuaikan untuk mendeteksi fitur khusus dari dataset baru[27].

Tujuan utama penggunaan *transfer learning* guna mengatasi masalah kekurangan data pelatihan (*insufficient training data*)[28] dan mengurangi waktu pelatihan secara signifikan[27][28]. Dalam pengimplementasiannya, terdapat berbagai cara untuk mengintegrasikan *transfer learning* ke dalam model *deep learning*. Salah satu

pendekatan umum adalah dengan membekukan (*freeze*) bobot pada Sebagian besar lapisan dari model *pre-trained* dan hanya melatih lapisan-lapisan baru yang ditambahkan untuk tugas spesifik. Menurut penelitian [28] yang menggambarkan pendekatan hibrida dimana pada awal pelatihan, hanya lapisan-lapisan yang baru ditambahkan yang dilatih berdasarkan dataset baru, dan setelah jumlah iterasi, Misalnya, setelah iterasi ke-20 seperti yang dilakukan dalam penelitian klasifikasi buah, lapisan yang sudah ada dibekukan sebelumnya, kemudian tidak dibekukan untuk penyesuaian bobot ringan[28].

Hasil penelitian [28] yang menerapkan *transfer learning* sebagai penambahan lapisan khusus pada model TL-*MobileNetV2* yang merupakan modifikasi *MobileNetV2*, yang dimana model ini dilatih pada dataset berisi 26.149 citra dari 40 jenis buah, yang menghasilkan akurasi sebesar 99%, yang dimana hasil akurasi ini membuktikan lebih tinggi 3% dibandingkan arsitektur *MobileNetV2* tanpa modifikasi dan *transfer learning*, peneliti menyimpulkan bahwa *transfer learning* mempunyai peran besar dalam mencapai hasil yang lebih baik pada penelitiannya[28].

2.6 *MobileNetV2*

MobileNet merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang telah dikembangkan secara spesifik untuk klasifikasi citra[10]. Salah satu karakteristik utama *MobileNet* dan variannya, seperti *MobileNetV2* dan *MobileNetV3-small*, adalah desainnya yang ringan (*lightweight*) dan efisien, yang menjadikannya kompatibel dengan tujuan implementasi pada perangkat dengan sumber daya komputasi terbatas, seperti perangkat seluler atau *edge devices*[10]. Secara teknis, efisiensi *MobileNet* dicapai melalui penggunaan Depthwise Separable Covolution[19][18].

Operasi ini memecah konvolusi standar menjadi dua langkah yang lebih sederhana dan terpisah yaitu *depthwise convolution* yang menerapkan filter tunggal pada setiap *channel input*, dan *pointwise convolution* (konvolusi 1x1) yang menggabungkan output dari *depthwise convolution* tersebut sehingga jumlah operasinya menjadi.

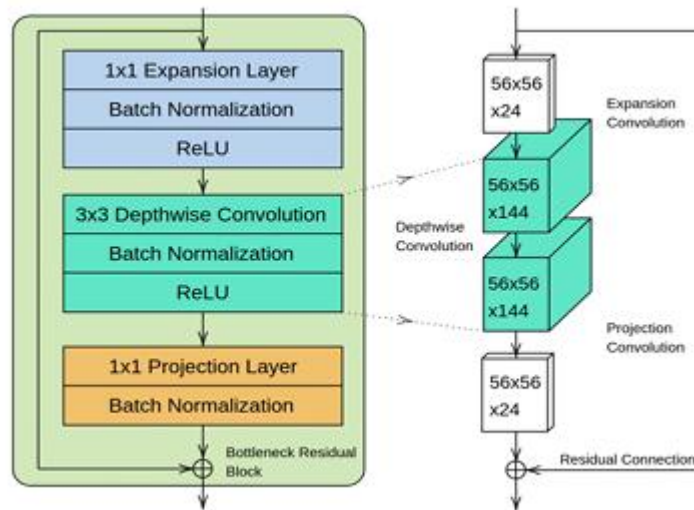
$$D_K \times D_K \times M \times D_F \times D_F \times M \times N \times D_F \times D_F$$

Yang dimana D_K adalah ukuran kernel, M dan N masing-masing merupakan jumlah saluran input dan output, serta D_F adalah ukuran fitur spasial[29]. Teknik ini secara signifikan mengurangi jumlah parameter dan komputasi yang dibutuhkan oleh model dibandingkan dengan konvolusi standar[19]. *MobileNetV2* semakin meningkatkan efisiensi modelnya dengan memperkenalkan konsep *Inverted Residual* dan *Linear Bottlenecks*[19][18]. Yang dimana *Inverted Residual* dan *Linear Bottlenecks* terdiri dari tiga tahap yaitu:

1. Ekspansi mengaplikasikan konvolusi 1×1 untuk memperbesar dimensi saluran ($C \rightarrow t \cdot C$)
2. *Depth wise convolution* 3×3
3. Proyeksi Kembali ke dimensi semula $t \cdot C \rightarrow C'$ mengaplikasikan konvolusi 1×1 dengan aktivasi linear[29].

Alih-alih mengurangi dimensi channel di awal blok residual seperti pada arsitektur ResNet konvensional, *MobileNetV2* memperluas dimensi channel, mengimplementasikan *depthwise convolution*, lalu menguranginya Kembali dengan *pointwise convolution linear* tanpa fungsi aktivasi non-linear seperti ReLU[19]. Penggunaan *linear bottleneck* di akhir blok ini bertujuan untuk menjaga informasi yang signifikan di dimensi rendah[19]. Selain itu, *MobileNetV2* juga mengaplikasikan *Shortcut Connections* antar *Bottleneck*, mirip dengan *ResNet*, yang membantu pelatihan jaringan yang lebih dalam dan meningkatkan akurasi[19]. Kombinasi fitur-fitur ini membuat *MobileNetV2* sangat efektif dalam mengklasifikasikan objek[19].

Secara struktural, arsitektur *MobileNetV2* terdiri dari rangkaian blok Misalnya, 17 blok residual bottleneck dalam satu implementasi[19], di mana setiap blok biasanya mencakup *Expansion Layer*, *Depthwise Covolution*, dan *Projection Layer*[19]. Sebagian besar lapisan dalam blok ini dilengkapi dengan Conv2D, *BatchNormalization*, dan aktivasi ReLU, kecuali *Projection Layer* tidak memanfaatkan ReLU[19]. *BatchNormalization* sendiri diterapkan untuk memaksimalkan proses pelatihan dan meminimalkan kemungkinan terjadinya *overfitting*[19]. Berikut visualisasi arsitektur *MobileNetV2* pada gambar 2.1 dibawah ini:



Gambar 2. 1 Arsitektur *MobileNetV2* [19]

Dalam konteks deteksi dan klasifikasi penyakit tanaman berdasarkan citra daun, *MobileNetV2* dan variannya telah mengindikasikan kinerja yang efektif, bahkan tinggi, seperti temuan dari penelitian-penelitian sebelumnya. Pada penelitian[10] perbandingan klasifikasi penyakit daun padi mengindikasikan bahwa MobileNet mencapai akurasi 94,79 persen dalam 10 epoch, melampaui CNN konvensional yang mencapai akurasi 90,24 persen setelah 150 epoch, hal itu mengungkapkan keunggulan MobileNet dalam efisiensi waktu pelatihan dan kinerja. Penelitian lain yang membandingkan MobileNetV3-Small dengan ResNet50 dan VGG16 untuk klasifikasi lima penyakit daun padi, dimana MobileNetV3-Small memberikan akurasi pengujian yang optimal sebesar 79 persen[14]. Berdasarkan penelitian-penelitian sebelumnya, arsitektur *MobileNetV2* dengan kombinasi fitur *Depthwise Seperable Convolution*, *Inverted Residual*, *Linear Bottlenecks*, dan *Shortcut Connections* menawarkan keseimbangan yang memadai antara akurasi klasifikasi dan efisiensi komputasi[14], menjadikannya pilihan yang efektif, terutama untuk pengaplikasian deteksi dan klasifikasi citra pada platform sumber daya terbatas[18].

2.7 Fine-Tuning

Setelah tahap pelatihan awal (*head training*) selesai, dilakukan proses *fine-tuning* untuk meningkatkan performa model. *Fine-tuning* merupakan tahapan dalam *transfer*

learning di mana sebagian lapisan dari *pre-trained* model yang sebelumnya dibekukan (*frozen*) kemudian dibuka (*unfreeze*) agar ikut dilatih pada dataset penelitian. Dengan cara ini, bobot yang sudah dimiliki model dari dataset besar (*ImageNet*) dapat disesuaikan lebih spesifik terhadap karakteristik citra daun padi[30].

2.8 Explainability dengan Grad-CAM

Seiring dengan meningkatnya kompleksitas model deep learning, muncul kebutuhan untuk memahami proses pengambilan keputusan yang dilakukan oleh model tersebut. Salah satu pendekatan yang banyak digunakan untuk menjelaskan hasil prediksi adalah *Gradient-weighted Class Activation Mapping (Grad-CAM)* [31] yang mampu menghasilkan heatmap untuk menunjukkan area citra yang paling berkontribusi terhadap keputusan klasifikasi. Dalam penerapan pada klasifikasi penyakit daun padi, *Grad-CAM* berperan dalam memverifikasi bahwa model benar-benar memusatkan perhatian pada bagian daun yang mengalami bercak atau kerusakan, bukan pada latar belakang gambar. Dengan demikian, metode ini dapat meningkatkan tingkat transparansi serta kepercayaan pengguna terhadap sistem klasifikasi yang dihasilkan.

2.9 Perancangan Aplikasi

Pemanfaatan teknologi kecerdasan buatan, khususnya deep learning, dalam mendeteksi penyakit pada tanaman telah berkembang pesat, dengan fokus pada penyebaran model-model ini ke dalam aplikasi yang digunakan secara praktis oleh masyarakat, seperti petani[32]. Petani berpotensi memanfaatkan model yang telah dibuat untuk mengklasifikasi jenis penyakit pada daun padi di lapangan melalui aplikasi web atau aplikasi berbasis server yang telah dikembangkan[33]. Beberapa penelitian telah berhasil mengimplementasikan model yang dilatih ke dalam aplikasi seluler[32][16]. Seperti pada penelitian [16] yang membuat model untuk klasifikasi penyakit daun padi menggunakan CNN diintegrasikan ke dalam aplikasi seluler dengan mengkonversi model *TensorFlow* keras menjadi buffer datar terkompresi dengan *TensorFlow Lite Converter*, yang kemudian ditempatkan dalam aplikasi untuk klasifikasi penyakit. yang dimana aplikasi seluler ini memungkinkan identifikasi penyakit pada tanaman padi dan memberikan solusi yang relevan.

Dalam konteks perancangan aplikasi berbasis web, salah satu penelitian yang telah dilakukan sebelumnya [33] membahas pengembangan aplikasi web dan seluler untuk klasifikasi penyakit daun singkong (*cassava*). Penelitian [33] tersebut menggunakan model CNN yang dilatih menggunakan teknik *transfer learning*. Model yang telah dikembangkan ini disebar (deployed) sebagai aplikasi web[33]. Untuk perancangan aplikasi web tersebut, digunakan kerangka kerja *Python* yang disebut *Streamlit*[33]. Menurut [33], *Streamlit* adalah pustaka *Python* yang memungkinkan pengguna dan pengembang untuk membuat aplikasi web yang menarik dan mudah digunakan[33]. *Streamlit* dirancang untuk membuat dan berbagi aplikasi web yang dipersonalisasi untuk analisis data dan machine learning secara sederhana[33].

