

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Penelitian terkait berisi dasar ilmiah perkembangan topik penelitian dalam skripsi (*state of the art*) dan memperlihatkan posisi penelitian yang akan dilakukan dalam perkembangan topik. Dasar ilmiah diambil dari penelitian-penelitian sebelumnya menggunakan metode yang sama dalam jurnal/paper, *proceeding*, maupun artikel ilmiah yang terbit pada periode 5 (lima) tahun terakhir. Penelitian terkait disajikan dalam bentuk tabel dan/atau *mind map*.

Tabel 1 Penelitian Terkait

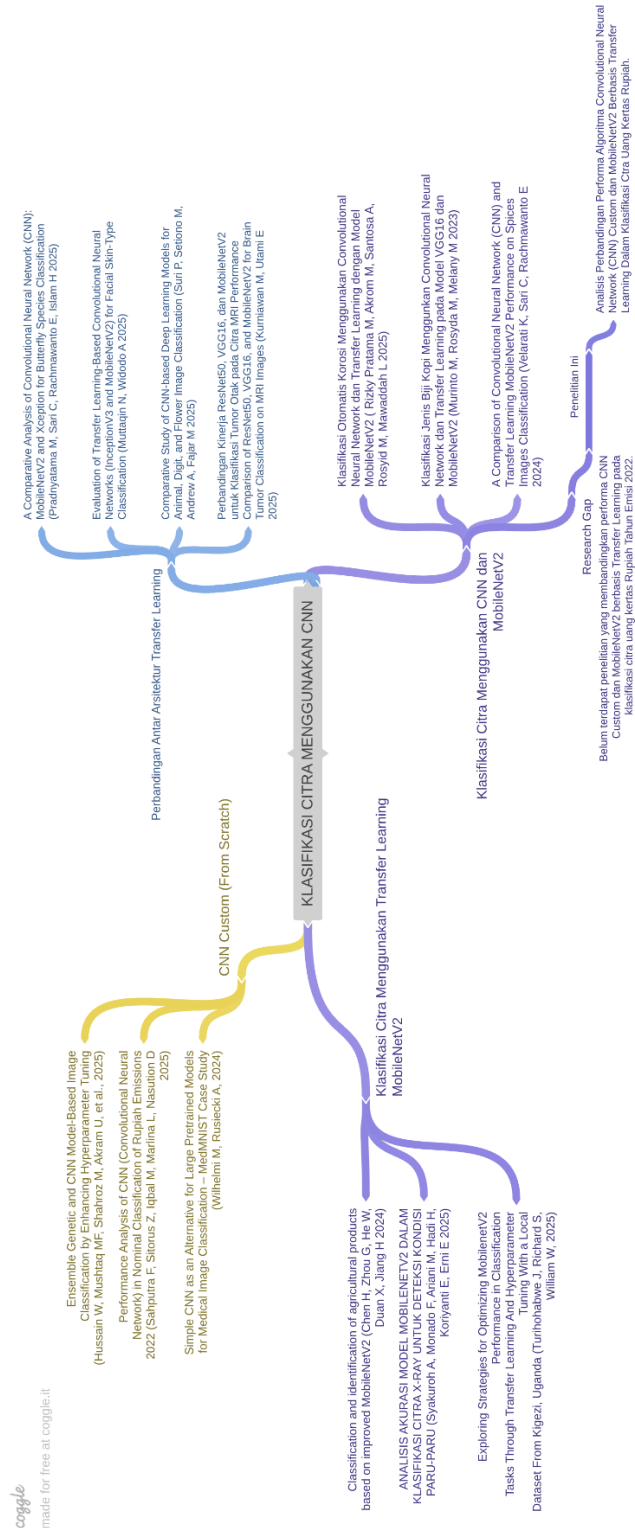
No.	Nama (Tahun)	Judul
(1)	(2)	(3)
1	Velarati K, Sari C, Rachmawanto E (2024)	Judul: <i>A Comparison of Convolutional Neural Network (CNN) and Transfer Learning MobileNetV2 Performance on Spices Images Classification</i> Metode: CNN Custom dan <i>Transfer Learning</i> MobileNetV2 Hasil: Pada eksperimen pertama, CNN tanpa <i>transfer learning</i> mencapai akurasi 93%. Eksperimen kedua menggunakan MobileNetV2, akurasi meningkat mencapai nilai 99% untuk semua kelas rempah-rempah.
2	Pradnyatama et al. (2025)	Judul: <i>A Comparative Analysis of Convolutional Neural Network (CNN): MobileNetV2 and Xception for Butterfly Species Classification</i> Metode: <i>Transfer Learning</i> MobileNetV2 dan Xception Hasil: Hasil eksperimen menunjukkan Xception mencapai akurasi uji 93,40% dibandingkan dengan 93,20% pada MobileNetV2. Tingkat akurasi tinggi ini dicapai melalui <i>transfer learning</i> yang efektif yang mempertahankan kemampuan ekstraksi fitur umum, teknik penyeimbangan kelas yang komprehensif, dan strategi laju pembelajaran yang disesuaikan dengan cermat untuk setiap arsitektur. Meskipun perbedaan kinerja minimal, MobileNetV2 menawarkan keunggulan efisiensi komputasi yang signifikan dengan 4,15 juta parameter dibandingkan dengan 25,27 juta parameter Xception, sementara

No.	Nama (Tahun)	Judul
(1)	(2)	(3)
		Xception memberikan klasifikasi yang sedikit lebih baik.
3	Muttaqin dan Widodo et al. (2025)	Judul: <i>Evaluation of Transfer Learning-Based Convolutional Neural Networks (InceptionV3 and MobileNetV2) for Facial Skin-Type Classification</i> Metode: Transfer Learning (InceptionV3 dan MobileNetV2) Hasil: Metode InceptionV3 mencapai akurasi tertinggi yaitu 90,12% dan skor F1 makro 89,47%, terutama unggul dalam mengidentifikasi kulit normal dan rentan jerawat. MobileNetV2 mencapai akurasi 81,15% dan berkinerja baik pada jenis kulit kering.
4	Suri et al. (2025)	Judul: <i>Comparative Study of CNN-based Deep Learning Models for Animal, Digit, and Flower Image Classification</i> Metode: Model jaringan saraf konvolusional (CNN) – MobileNetV2, DenseNet121, EfficientNetB0, dan InceptionV3 Hasil: Hasil menunjukkan bahwa DenseNet121 secara konsisten mencapai akurasi sebesar 98% pada gambar hewan dan 88% pada gambar bunga, sementara MobileNetV2 memberikan kinerja yang hampir sama (97% dan 82%) tetapi dengan waktu pemrosesan yang jauh lebih cepat, antara 11 dan 55 menit. EfficientNetB0, di sisi lain, berkinerja buruk pada dataset bunga yang lebih kompleks, hanya mencapai akurasi 5%. Temuan ini menunjukkan bahwa DenseNet121 ideal untuk proyek di mana akurasi menjadi prioritas utama, sedangkan MobileNetV2 lebih cocok untuk aplikasi <i>real-time</i> yang memerlukan respons cepat tanpa penurunan akurasi yang signifikan.
5	Chen et al. (2024)	Judul: <i>Classification and identification of agricultural products based on improved MobileNetV2</i> Metode: Improved MobileNetV2 (Res-Inception dan Efficient Multi-scale Attention) Hasil: Model Improved-MobileNetV2 menunjukkan peningkatan performa dibandingkan MobileNetV2

No.	Nama (Tahun)	Judul
(1)	(2)	(3)
		standar. MobileNetV2 memperoleh akurasi sebesar 97,26%, sedangkan Improved-MobileNetV2 mencapai akurasi hingga 99,96% pada dataset klasifikasi produk pertanian. Peningkatan tersebut diperoleh melalui modifikasi arsitektur dengan penambahan modul yang dirancang untuk meningkatkan kemampuan ekstraksi fitur multi-skala. Selain itu, model Improved-MobileNetV2 memiliki jumlah parameter sekitar 3.028.211, jumlah yang relatif lebih kecil dibandingkan beberapa arsitektur CNN lain.
6	Rizky Pratama et al. (2025)	Judul: Klasifikasi Otomatis Korosi Menggunakan <i>Convolutional Neural Network</i> dan <i>Transfer Learning</i> dengan Model MobileNetV2 Metode: CNN vs MobileNetV2 (Transfer Learning) Hasil: MobileNetV2 mencapai akurasi uji 95% dan AUC 0,9893, lebih tinggi dibanding CNN sederhana sebesar 82%. Hal ini menunjukkan efektivitas <i>transfer learning</i>.
7	Sahputra et al. (2025)	Judul: <i>Performance Analysis of CNN (Convolutional Neural Network) in Nominal Classification of Rupiah Emissions 2022</i> Metode: CNN+Adam, CNN+RMSprop, dan VGG16 (transfer learning) Hasil: Hasil menunjukkan bahwa CNN dengan optimizer Adam mencapai akurasi validasi 98,97%, sementara CNN dengan RMSprop mencapai 99,59%. Sementara itu, model VGG16 mencapai akurasi validasi sempurna sebesar 100%, dengan nilai presisi, <i>recall</i>, dan <i>F1-score</i> masing-masing 1.00. Hasil ini menunjukkan bahwa pendekatan <i>transfer learning</i> memberikan kinerja terbaik dibandingkan dengan model CNN konvensional.
8	Syakuroh et al. (2025)	Judul: Analisis Akurasi Model MobileNetV2 dalam Klasifikasi CITRA X-RAY untuk Deteksi Kondisi Paru-Paru Metode: <i>Transfer Learning</i> MobileNetV2 Hasil: Hasil pengujian pada data uji menunjukkan bahwa model yang diusulkan mampu mencapai nilai akurasi sebesar 99,42%, <i>precision</i> 98,87%, <i>recall</i>

No.	Nama (Tahun)	Judul
(1)	(2)	(3)
		98,88%, dan <i>F1-score</i> 98,86%. Seluruh hasil evaluasi ini melampaui standar medis minimal $\geq 90\%$ aplikasi klinis. Temuan penelitian ini menegaskan bahwa MobileNetV2 sangat potensial digunakan sebagai alat bantu diagnosis otomatis berbasis citra X-ray, guna meningkatkan efektivitas deteksi dini penyakit paru-paru pada lingkungan klinis.
9	Murinto M, Rosyda M, Melany M (2023)	Judul: Klasifikasi Jenis Biji Kopi Menggunakan <i>Convolutional Neural Network</i> dan <i>Transfer Learning</i> pada Model VGG16 dan MobileNetV2 Metode: CNN custom, <i>Transfer Learning</i> VGG16, dan <i>Transfer Learning</i> MobileNetV2 Hasil: Model CNN memperoleh akurasi 95%, VGG16 memperoleh akurasi 95%, dan MobileNetV2 memperoleh akurasi tertinggi sebesar 96%. Hasil menunjukkan bahwa pendekatan <i>transfer learning</i> memberikan peningkatan performa dibanding CNN biasa, meskipun peningkatan akurasi relatif kecil.
10	Kurniawan M, Utami E (2025)	Judul: Perbandingan Kinerja ResNet50, VGG16, dan MobileNetV2 untuk Klasifikasi Tumor Otak pada Citra MRI Metode: ResNet50, VGG16, dan MobileNetV2 Hasil: Hasil eksperimen menunjukkan bahwa VGG16 memberikan performa terbaik di antara ketiga model dengan akurasi 94.93%, presisi 94.68%, dan <i>specificity</i> 98.33%, serta waktu pelatihan terpendek, yaitu 47.15 menit. Sementara itu, MobileNetV2 menunjukkan kinerja yang solid dengan recall 94.08%, meskipun membutuhkan waktu pelatihan lebih lama (79.53 menit). ResNet50 mencatatkan akurasi terendah (91.67%), meskipun unggul dalam presisi (91.79%), namun memiliki kekurangan pada recall (91.25%) dan <i>specificity</i> (97.2%).

Gambar 1 merupakan *mind map* penelitian terkait sebagai referensi penulisan dalam bentuk gambar, yaitu sebagai berikut :



Gambar 1 *Mind Map* Penelitian

B. Landasan Teori

Landasan teori pada penelitian “Analisis Perbandingan Performa Algoritma *Convolutional Neural Network (CNN) Custom* dan *Transfer Learning* Berbasis MobileNetV2 dalam Klasifikasi Citra Uang Kertas Rupiah” yaitu:

1. Citra Digital

Citra digital merupakan representasi visual suatu objek atau fenomena yang dinyatakan dalam bentuk data numerik sehingga dapat diproses oleh komputer. Dalam pengolahan citra digital, sebuah gambar tidak hanya dipandang sebagai bentuk visual, melainkan sebagai sekumpulan nilai numerik yang tersusun secara sistematis dalam suatu struktur data. Citra digital tersusun atas elemen-elemen kecil yang disebut piksel (*picture element*), dengan setiap piksel memiliki nilai tertentu yang merepresentasikan tingkat kecerahan atau warna pada posisi tertentu dalam citra [17].

Secara matematis, citra digital dua dimensi direpresentasikan dalam bentuk matriks berukuran $M \times N$, dengan M menyatakan jumlah baris dan N menyatakan jumlah kolom. Setiap elemen matriks tersebut menunjukkan satu piksel dengan nilai intensitas tertentu. Nilai piksel pada posisi tertentu dapat dinyatakan sebagai $f(i, j)$, yaitu nilai intensitas pada koordinat spasial tertentu [18]. Representasi dalam bentuk matriks ini memungkinkan citra diproses menggunakan operasi matematika dan komputasi.

Berdasarkan jenis informasinya, citra digital dapat dibedakan menjadi citra *grayscale* dan citra berwarna. Pada citra *grayscale*, setiap piksel hanya memiliki satu nilai intensitas dalam rentang tertentu, umumnya berada pada rentang 0 hingga 255 pada sistem 8-bit, dengan 0 menunjukkan warna hitam dan 255 menunjukkan warna putih. Sementara itu, citra berwarna umumnya menggunakan model warna RGB (*Red, Green, Blue*), setiap piksel direpresentasikan oleh tiga nilai intensitas yang masing-masing menyatakan komponen merah, hijau, dan biru. Kombinasi ketiga kanal tersebut menghasilkan variasi warna yang luas pada citra digital [18].

Selain representasi piksel dan model warna, kualitas citra digital juga dipengaruhi oleh resolusi. Resolusi citra merujuk pada jumlah piksel yang menyusun suatu citra dan biasanya dinyatakan dalam bentuk $M \times N$. Citra dengan resolusi tinggi memiliki jumlah piksel yang lebih banyak sehingga mampu merepresentasikan detail visual secara lebih jelas, sedangkan resolusi rendah menghasilkan citra yang tampak lebih kasar [17].

Proses pembentukan citra digital dari citra analog melibatkan dua tahapan penting, yaitu *sampling* dan *quantization*. *Sampling* merupakan proses pengambilan titik-titik diskrit dari citra kontinu untuk membentuk *grid* piksel yang menentukan resolusi citra. *Quantization* adalah proses pembatasan nilai intensitas kontinu

menjadi sejumlah level diskrit tertentu, misalnya 256 tingkat pada citra 8-bit [18]. Kedua proses ini sangat menentukan kualitas citra digital yang dihasilkan. Dengan demikian, citra digital menjadi dasar dalam berbagai proses pengolahan dan analisis visual, karena seluruh teknik komputasi dalam bidang ini bekerja berdasarkan representasi piksel dalam bentuk numerik tersebut.

2. *Computer Vision*

Computer vision merupakan bidang ilmu yang mempelajari mengenai proses sistem komputer dapat memperoleh, memproses, dan memahami informasi dari data visual seperti citra maupun video. Tujuan utama bidang ini adalah mengembangkan algoritma yang memungkinkan mesin menafsirkan dunia visual secara otomatis, sehingga komputer tidak hanya membaca nilai piksel, tetapi juga mampu memahami struktur, pola, dan makna yang terkandung di dalam suatu gambar [19].

Secara umum, *computer vision* mencakup berbagai aspek, mulai dari proses pembentukan citra (*image formation*), pengolahan citra (*image processing*), hingga pengenalan dan interpretasi objek (*recognition*) [19]. Berbeda dengan pengolahan citra yang berfokus pada manipulasi atau peningkatan kualitas visual, *computer vision* menitikberatkan pada kemampuan sistem dalam melakukan analisis serta menghasilkan pemahaman terhadap isi visual tersebut. Permasalahan dalam *computer vision* dapat berupa pengenalan objek, deteksi objek, segmentasi citra, serta rekonstruksi tiga dimensi seperti *structure from motion* dan estimasi kedalaman (*depth estimation*) [19]. Berbagai permasalahan tersebut bertujuan untuk mengubah data visual menjadi informasi yang dapat digunakan dalam proses pengambilan keputusan oleh sistem komputasi.

Seiring perkembangan teknologi, pendekatan dalam *computer vision* semakin mengarah pada metode berbasis data (*data-driven approach*), dengan sistem belajar dari sejumlah besar data visual untuk meningkatkan akurasi dan kemampuan generalisasi model [19]. Perkembangan ini kemudian menjadi fondasi bagi munculnya berbagai pendekatan modern dalam analisis visual. Salah satu pendekatan yang berkembang pesat adalah deep learning, yang mampu mempelajari representasi fitur secara otomatis langsung dari data citra tanpa memerlukan proses ekstraksi fitur secara manual. Kemampuan tersebut menjadikan deep learning banyak diterapkan pada berbagai tugas computer vision, termasuk klasifikasi citra dengan karakteristik visual yang kompleks.

3. *Deep Learning*

Deep Learning (DL) merupakan bagian dari *machine learning* yang berfokus pada penggunaan jaringan saraf tiruan berlapis banyak (*multi-layer neural networks*) untuk mempelajari representasi data secara hierarkis. Dalam perkembangannya, *deep learning* muncul sebagai pendekatan yang mampu

mengatasi keterbatasan metode *machine learning* konvensional dalam menangani data mentah berskala besar dan kompleks [20].

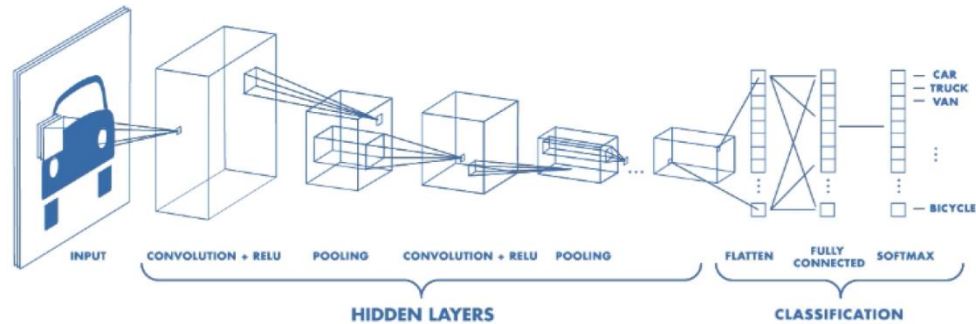
Berbeda dengan pendekatan *machine learning* tradisional yang umumnya memerlukan proses rekayasa fitur (*feature engineering*) secara manual, *deep learning* mengandalkan konsep *representation learning*. Konsep ini memungkinkan sistem secara otomatis mempelajari representasi fitur dari data mentah melalui serangkaian transformasi non-linear pada beberapa lapisan jaringan [20]. Dengan demikian, model tidak lagi sepenuhnya bergantung pada perancangan fitur oleh manusia, melainkan mampu membangun representasi internal yang lebih abstrak dan kompleks.

Istilah “*deep*” dalam *deep learning* merujuk pada kedalaman arsitektur jaringan, yaitu banyaknya lapisan pemrosesan (*processing layers*) yang digunakan dalam model. Setiap lapisan bertugas mengekstraksi pola yang semakin kompleks dari lapisan sebelumnya. Lapisan awal umumnya menangkap fitur sederhana, sedangkan lapisan yang lebih dalam mempelajari representasi tingkat tinggi yang lebih abstrak[20]. Struktur bertingkat ini memungkinkan model memahami hubungan *non-linear* dalam data secara lebih efektif.

Deep learning telah banyak digunakan dalam menangani data tidak terstruktur seperti citra, teks, dan suara. Berbagai arsitektur seperti *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), dan *Generative Adversarial Network* (GAN) dikembangkan untuk menyelesaikan permasalahan pada domain yang berbeda [20]. Perkembangannya didukung oleh ketersediaan data berskala besar serta peningkatan kemampuan komputasi modern.

Meskipun demikian, *deep learning* juga memiliki sejumlah tantangan, antara lain kebutuhan data pelatihan dalam jumlah besar, waktu pelatihan yang relatif lama, serta kompleksitas model yang tinggi. Selain itu, model *deep learning* berpotensi mengalami *overfitting* apabila tidak dilakukan pengaturan parameter dan strategi regularisasi yang tepat [20]. Dengan karakteristik tersebut, *deep learning* menjadi landasan penting dalam pengembangan berbagai metode analisis data modern.

4. Convolutional Neural Network (CNN)



Gambar 2 Algoritma *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan salah satu arsitektur *deep learning* yang dirancang khusus untuk mengolah data berbentuk *grid*, seperti citra dua dimensi. CNN termasuk dalam kategori *Artificial Neural Network* (ANN) dan banyak digunakan dalam bidang *computer vision* karena kemampuannya dalam mengekstraksi fitur visual secara otomatis tanpa memerlukan perancangan fitur manual [21], [22]. Proses ekstraksi fitur tersebut dilakukan secara bertingkat, mulai dari fitur sederhana seperti tepi dan tekstur hingga fitur yang lebih kompleks, sehingga model mampu mengenali pola visual secara lebih efektif. Kemampuan tersebut menjadikan CNN sebagai salah satu metode yang paling banyak digunakan dalam tugas klasifikasi citra, termasuk untuk mengenali objek yang memiliki karakteristik visual kompleks seperti uang kertas.

Berbeda dengan jaringan saraf konvensional yang menggunakan koneksi penuh (*fully connected network*), CNN memanfaatkan prinsip *local connectivity* dan *weight sharing*. Pada mekanisme ini, setiap filter (kernel) hanya bekerja pada area lokal citra dan menggunakan bobot yang sama saat digeser ke seluruh bagian citra. Pendekatan tersebut membuat CNN lebih efisien dalam jumlah parameter serta lebih efektif dalam menangkap pola spasial seperti tepi, tekstur, dan bentuk objek [21], [22].

Secara umum, arsitektur CNN terdiri atas beberapa komponen utama, yaitu *convolution layer*, *pooling layer*, *activation function*, dan *fully connected layer* [22], [23]. Kombinasi lapisan tersebut memungkinkan proses ekstraksi fitur dan klasifikasi dilakukan secara terintegrasi dalam satu model.

a. *Convolution Layer*

Convolution layer merupakan komponen utama dalam CNN. Pada lapisan ini, citra masukan dikonvolusikan dengan sejumlah kernel berukuran kecil, seperti 3×3 atau 5×5 , untuk menghasilkan *feature map* yang merepresentasikan pola tertentu dalam citra [22], [23]. Proses konvolusi dilakukan dengan menggeser kernel ke seluruh bagian citra, kemudian menghitung hasil perkalian

elemen yang bersesuaian dan menjumlahkannya. Parameter penting dalam *convolution layer* meliputi:

- 1) *Kernel size*, yaitu ukuran filter;
- 2) *Stride*, yaitu langkah pergeseran filter;
- 3) *Padding*, yaitu penambahan nilai pada tepi citra untuk mempertahankan dimensi keluaran.

Pengaturan parameter tersebut memengaruhi ukuran *feature map* dan kemampuan model dalam menangkap informasi visual [21], [22].

b. *Pooling Layer*

Pooling layer berfungsi untuk melakukan reduksi dimensi terhadap *feature map* yang dihasilkan oleh *convolution layer*. Tujuannya adalah mengurangi jumlah parameter, meningkatkan efisiensi komputasi, serta membuat model lebih tahan terhadap pergeseran kecil pada citra [22], [23]. Metode *pooling* yang umum digunakan antara lain:

- 1) *Max pooling*, yang mengambil nilai maksimum dalam suatu area;
- 2) *Average pooling*, yang mengambil nilai rata-rata dalam suatu area;

Penggunaan *pooling* membantu mempertahankan informasi penting sekaligus mengurangi kompleksitas data [22].

c. *Activation Function*

Activation Function digunakan untuk memperkenalkan sifat non-linear dalam jaringan. Tanpa fungsi aktivasi, jaringan hanya menghasilkan transformasi linear dan tidak mampu memodelkan hubungan kompleks dalam data. Beberapa fungsi aktivasi yang umum digunakan dalam CNN antara lain:

- 1) *Sigmoid*;
- 2) *Tanh*,
- 3) *ReLU (Rectified Linear Unit)*,
- 4) *Leaky ReLU*,

ReLU menjadi fungsi aktivasi yang paling banyak digunakan karena lebih sederhana secara komputasi dan membantu mengurangi permasalahan *vanishing gradient* [22], [23].

d. *Fully Connected Layer*

Fully Connected layer biasanya ditempatkan pada bagian akhir arsitektur CNN. Lapisan ini menerima hasil ekstraksi fitur yang telah diratakan (*flattened*)

dan melakukan proses klasifikasi berdasarkan fitur tersebut [23]. Pada permasalahan klasifikasi multi-kelas, lapisan akhir umumnya menggunakan fungsi aktivasi seperti *softmax* untuk menghasilkan probabilitas setiap kelas [22].

5. Arsitektur *Convolutional Neural Network Custom*

Arsitektur CNN *custom* merupakan model jaringan saraf konvolusional yang dirancang secara spesifik sesuai dengan kebutuhan permasalahan penelitian. Berbeda dengan model *pre-trained* seperti ResNet atau VGG yang arsitekturnya telah ditentukan sebelumnya dan dilatih pada dataset berskala besar misalnya ImageNet, CNN *custom* dibangun dari awal dengan menentukan sendiri jumlah layer, jumlah filter, ukuran kernel, serta strategi pelatihan yang sesuai dengan kompleksitas dataset yang digunakan.

Secara konseptual, struktur dasar CNN *Custom* tetap mengikuti struktur dasar CNN yang terdiri atas beberapa lapisan, yaitu *convolution layer*, *activation function*, *pooling layer*, dan *fully connected layer*. Namun, kedalaman jaringan dan jumlah parameter dirancang secara adaptif sesuai kebutuhan. Menurut Taye [21] arsitektur *deep learning* dapat dikembangkan secara fleksibel untuk menyesuaikan kompleksitas model dengan karakteristik data yang digunakan. Prinsip ini menjadi dasar dalam perancangan CNN *Custom*, dengan struktur jaringan tidak harus mengikuti arsitektur besar yang telah ditentukan sebelumnya.

Penggunaan CNN *custom* umumnya dipertimbangkan ketika dataset bersifat spesifik atau jumlah data relatif terbatas, sehingga diperlukan model dengan kompleksitas yang lebih terkendali. Selain itu, model *custom* sering kali lebih efisien dari sisi jumlah parameter dan kebutuhan komputasi. Penelitian yang dilakukan oleh Alam *et al.* [24] menunjukkan bahwa CNN *Custom* yang dirancang dengan konfigurasi lapisan yang tepat mampu menghasilkan performa klasifikasi yang kompetitif dibandingkan model *pre-trained*, dengan waktu pelatihan dan ukuran model yang lebih efisien. Dengan demikian, arsitektur CNN *custom* merupakan pendekatan yang relevan dalam pengembangan sistem klasifikasi citra, terutama ketika mempertimbangkan keseimbangan antara akurasi, kompleksitas model, dan efisiensi komputasi.

6. *Transfer Learning*

Transfer learning merupakan salah satu teknik dalam *machine learning* yang memungkinkan pemanfaatan pengetahuan yang telah dilatih sebelumnya untuk menyelesaikan tugas baru yang berkaitan. Berbeda dengan pendekatan *machine learning* tradisional yang melatih model dari awal untuk setiap tugas secara terpisah, *transfer learning* memanfaatkan model atau pengetahuan dari domain sumber untuk meningkatkan performa pembelajaran pada domain target [25]. Konsep utama *transfer learning* adalah memanfaatkan hubungan antara tugas

sebelumnya dengan tugas yang sedang dipelajari. Pendekatan ini membantu model menghasilkan solusi yang lebih cepat dan akurat, terutama ketika jumlah data pada domain target terbatas atau sulit diperoleh [25].

Menurut Gupta *et al.* [26] *transfer learning* merupakan metode yang menggunakan model yang telah dilatih sebelumnya sebagai dasar untuk membangun model baru pada dataset yang berbeda. Model tersebut tidak perlu dilatih dari awal karena sudah memiliki fitur yang diperoleh dari proses pelatihan sebelumnya, sehingga dapat digunakan kembali untuk menyelesaikan permasalahan lain yang memiliki karakteristik serupa. Proses *transfer learning* melibatkan pemindahan pengetahuan dari *source* domain ke target domain. *Source* domain merupakan domain tempat model awal dilatih, sedangkan target domain merupakan domain baru yang menjadi tujuan pembelajaran. Dengan memanfaatkan pengetahuan yang telah dipelajari, model dapat meningkatkan performa pada tugas baru secara lebih efisien [25].

Dalam implementasinya, *transfer learning* sering menggunakan *pretrained model*, yaitu model yang telah dilatih menggunakan dataset berskala besar seperti ImageNet. Model tersebut kemudian disesuaikan dengan dataset baru melalui proses *fine-tuning*. Pendekatan ini memungkinkan model menggunakan fitur yang telah dipelajari sebelumnya, sehingga dapat meningkatkan akurasi dan mengurangi waktu pelatihan [26]. Penggunaan *transfer learning* telah diterapkan dalam berbagai bidang, seperti klasifikasi citra, pengolahan bahasa alami, sistem rekomendasi, dan klasifikasi dokumen. Teknik ini membantu meningkatkan performa model dengan memanfaatkan informasi yang telah tersedia dari domain lain [26].

Transfer learning memiliki beberapa jenis, yaitu *inductive transfer learning*, *transductive transfer learning*, dan *unsupervised transfer learning*. *Inductive transfer learning* digunakan ketika tugas sumber dan target berbeda tetapi masih berkaitan. *Transductive transfer learning* digunakan ketika tugasnya sama namun domainnya berbeda, sedangkan *unsupervised transfer learning* digunakan ketika tidak tersedia data berlabel pada kedua domain [25]. Berdasarkan konsep tersebut, *transfer learning* menjadi salah satu metode yang efektif untuk meningkatkan kinerja model *machine learning*, khususnya ketika data pelatihan terbatas dan waktu pelatihan menjadi pertimbangan penting.

7. MobileNetV2

MobileNetV2 merupakan arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk menghasilkan model yang ringan dan efisien, khususnya pada perangkat dengan keterbatasan sumber daya komputasi. Arsitektur ini diperkenalkan oleh Sandler *et al.* [27] pada *IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR) tahun 2018 melalui publikasi berjudul *MobileNetV2: Inverted Residuals and Linear Bottlenecks*.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 3 Algoritma MobileNetV2

Pengembangan MobileNetV2 bertujuan untuk meningkatkan efisiensi jumlah parameter dan kompleksitas komputasi, yang diukur melalui operasi *multiply-adds* (MAdds), tanpa menurunkan akurasi secara signifikan [27]. Dengan pendekatan ini, arsitektur mampu mencapai keseimbangan antara performa dan efisiensi komputasi. Inovasi utama dalam MobileNetV2 terletak pada kombinasi tiga komponen arsitektural, yaitu *depthwise separable convolution*, *linear bottleneck*, dan *inverted residual structure*.

a. *Depthwise Separable Convolution*

MobileNetV2 menggunakan *depthwise separable convolution* sebagai mekanisme komputasi utama untuk meningkatkan efisiensi [27]. Teknik ini merupakan pengembangan dari konvolusi standar dengan memisahkan proses konvolusi menjadi dua tahap terpisah. Pada konvolusi standar, setiap filter berukuran $k \times k$ bekerja pada seluruh kanal input dan menghasilkan seluruh kanal output secara langsung. Kompleksitas komputasi konvolusi standar dapat dinyatakan sebagai:

$$h_i \cdot w_i \cdot d_i \cdot d_j \cdot k^2$$

Keterangan:

h_i = tinggi Feature map

w_i = lebar feature map

d_i = jumlah kanal input

d_j = jumlah kanal output

k = ukuran kernel

Kompleksitas, semakin besar jumlah kanal *input* dan *output*, maka jumlah operasi meningkat secara signifikan. Sebaliknya, pada *depthwise separable convolution*, proses konvolusi dipisahkan menjadi dua tahap. Kompleksitas komputasi metode ini menjadi:

$$h_i \cdot w_i \cdot d_i(k^2 + d_j)$$

Pemisahan ini membuat proses perhitungan jauh lebih ringan karena konvolusi spasial (*depthwise*) dilakukan secara terpisah pada setiap kanal, lalu digabungkan melalui konvolusi 1×1 (*pointwise*). Menurut Sandler *et al.* [27] ketika menggunakan kernel 3×3, pendekatan ini dapat mengurangi biaya komputasi hingga sekitar 8-9 kali dibandingkan konvolusi standar, dengan penurunan akurasi yang relatif kecil. Oleh karena itu, *depthwise separable convolution* menjadi fondasi efisiensi pada arsitektur MobileNetV2.

b. *Linear Bottleneck*

Linear bottleneck merupakan salah satu kontribusi utama yang diperkenalkan dalam MobileNetV2 [27]. Konsep ini dikembangkan berdasarkan analisis bahwa representasi fitur dalam jaringan saraf dalam sering kali berada pada sub ruang berdimensi rendah. Pada kondisi tersebut, penggunaan fungsi aktivasi non-linear secara berulang dapat menyebabkan sebagian informasi penting hilang. Oleh karena itu, MobileNetV2 menerapkan linear bottleneck untuk membantu mempertahankan informasi pada ruang fitur berdimensi rendah sehingga representasi fitur yang dihasilkan tetap efektif untuk proses klasifikasi.

Pada arsitektur CNN konvensional, fungsi aktivasi *non-linear* seperti ReLU diterapkan setelah setiap operasi konvolusi. Namun, Sandler *et al.* [27] menunjukkan bahwa penerapan *non-linearitas* pada layer dengan jumlah kanal kecil (*bottleneck*) dapat menyebabkan hilangnya informasi. Hal ini terjadi karena ReLU memetakan nilai negatif menjadi nol. Ketika fitur telah direduksi ke dimensi rendah, pemotongan tersebut berpotensi menghilangkan informasi penting secara permanen. Untuk mempertahankan kekuatan representasi, MobileNetV2 tidak menggunakan fungsi aktivasi *non-linear* pada lapisan *bottleneck* akhir. Transformasi yang digunakan pada tahap proyeksi akhir bersifat *linear* [27]. Pendekatan ini memungkinkan fitur yang telah dikompresi tetap mempertahankan informasi penting tanpa mengalami distorsi akibat *non-linearitas*.

c. *Inverted Residual Structure*

Inverted residual merupakan inovasi arsitektural utama yang membedakan MobileNetV2 dari versi sebelumnya [27]. Struktur ini memanfaatkan koneksi *residual* pada layer berdimensi kecil (*bottleneck*), bukan pada layer berdimensi

besar seperti pada *residual block* tradisional. Satu blok *inverted residual* terdiri dari tiga tahap utama. Tahap pertama adalah ekspansi dimensi menggunakan konvolusi 1×1 dengan aktivasi ReLU6 untuk memperluas jumlah kanal. Tahap kedua adalah ekstraksi fitur spasial menggunakan *depthwise convolution* berukuran 3×3 . Tahap ketiga adalah proyeksi kembali ke dimensi kecil menggunakan konvolusi *linear* 1×1 tanpa fungsi aktivasi *non-linear*.

Koneksi *residual* diterapkan apabila dimensi *input* dan *output* sama. Desain ini memisahkan proses ekspansi fitur (untuk meningkatkan kapasitas representasi) dan proses kompresi fitur (untuk menjaga efisiensi). Sandler *et al.* [27] menjelaskan bahwa pendekatan ini memberikan efisiensi memori yang lebih baik serta tetap mendukung propagasi gradien secara stabil selama pelatihan. Struktur *inverted residual* bersama *linear bottleneck* menjadikan MobileNetV2 mampu mempertahankan keseimbangan antara efisiensi komputasi dan kemampuan representasi fitur.

8. Klasifikasi Citra

Klasifikasi citra merupakan proses pengelompokan suatu gambar ke dalam kategori tertentu berdasarkan informasi visual yang terkandung di dalamnya. Dalam konteks *computer vision*, tugas ini bertujuan agar sistem komputer mampu mengenali pola pada citra dan menentukan kelas yang paling sesuai secara otomatis [28]. Perkembangan metode klasifikasi citra mengalami perubahan signifikan sejak diperkenalkannya pendekatan berbasis *deep learning*. Sebelumnya, proses klasifikasi sangat bergantung pada fitur yang dirancang secara manual (*handcrafted features*). Namun, pendekatan ini memiliki keterbatasan karena kualitas hasil sangat ditentukan oleh desain fitur yang digunakan. Seiring perkembangan teknologi, *Convolutional Neural Network* (CNN) menjadi pendekatan utama karena mampu mempelajari representasi fitur secara otomatis langsung dari data citra [29].

CNN bekerja dengan cara melakukan ekstraksi fitur secara bertahap melalui beberapa lapisan konvolusi. Pada tahap awal, jaringan menangkap pola dasar seperti tepi atau tekstur. Selanjutnya, informasi tersebut diproses lebih dalam sehingga membentuk representasi yang lebih kompleks, misalnya pola bentuk atau struktur objek tertentu [28]. Mekanisme ini memungkinkan model membangun pemahaman visual secara hierarkis tanpa memerlukan perancangan fitur secara eksplisit.

Menurut Kim *et al.* [29] CNN telah menjadi pendekatan dominan dalam analisis citra karena kemampuannya melakukan pembelajaran secara langsung dari data mentah hingga menghasilkan prediksi kelas. Model dilatih menggunakan fungsi *loss* tertentu dan diperbarui melalui algoritma *backpropagation*, sehingga seluruh proses pembelajaran berlangsung secara terintegrasi (*end-to-end*).

Dalam praktiknya, keterbatasan jumlah data sering menjadi tantangan dalam klasifikasi citra. Oleh karena itu, strategi *transfer learning* banyak digunakan untuk meningkatkan performa model. Kim *et al.* [29] menjelaskan bahwa model yang telah dilatih pada dataset besar dapat dimanfaatkan kembali sebagai dasar pembelajaran untuk tugas klasifikasi lain, baik dengan membekukan sebagian parameter maupun menyesuaikannya kembali melalui proses *fine-tuning*. Pendekatan ini membantu mempercepat proses pelatihan sekaligus meningkatkan akurasi ketika data terbatas. Dengan demikian, klasifikasi citra modern berfokus pada pembelajaran representasi visual secara otomatis melalui arsitektur jaringan saraf dalam, terutama CNN, serta didukung oleh strategi pelatihan seperti *transfer learning* untuk meningkatkan performa dan efisiensi model.

9. Metrik Evaluasi Performa

Evaluasi performa merupakan tahap penting dalam pengembangan model klasifikasi untuk mengukur sejauh mana model mampu menghasilkan prediksi yang sesuai dengan label sebenarnya pada data uji. Pengukuran ini tidak hanya melihat tingkat ketepatan prediksi, tetapi juga mempertimbangkan keseimbangan antar kelas serta efisiensi model dalam implementasi praktis. Dalam sistem klasifikasi, evaluasi umumnya dilakukan menggunakan *confusion matrix* dan beberapa metrik turunan seperti *accuracy*, *precision*, *recall*, dan *F1-score* [30]. Selain itu, dalam pengembangan model *deep learning* modern, aspek efisiensi seperti waktu inferensi dan ukuran model juga menjadi indikator penting, terutama ketika model diimplementasikan pada perangkat dengan sumber daya terbatas [31].

a. *Confusion Matrix*

Confusion matrix merupakan tabel yang digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya [30]. Dalam proses evaluasi model klasifikasi, confusion matrix menghasilkan empat komponen utama yang menjadi dasar perhitungan berbagai metrik evaluasi, yaitu:

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 4 Confusion Matrix

dengan :

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

Confusion matrix menjadi dasar dalam perhitungan berbagai metrik evaluasi lainnya. Pada penelitian ini, setiap percobaan (*run*) menghasilkan sebuah *confusion matrix* untuk masing-masing model, kemudian nilai accuracy, precision, recall, dan F1-score dihitung berdasarkan *confusion matrix* tersebut.

b. *Accuracy*

Accuracy mengukur proporsi jumlah prediksi yang benar dibandingkan dengan seluruh data yang di uji [30]. Perhitungan *accuracy* dapat dinyatakan dalam bentuk persamaan berikut:

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

Metrik ini mudah dipahami karena menunjukkan tingkat ketepatan model secara keseluruhan.

c. *Precision*

Precision menunjukkan tingkat ketepatan prediksi positif, yaitu seberapa banyak prediksi positif yang benar-benar sesuai dengan kondisi sebenarnya [30]. *Precision* dihitung sebagai perbandingan antara jumlah prediksi positif yang benar (TP) terhadap seluruh prediksi positif yang dihasilkan model (TP dan FP), yang dirumuskan sebagai:

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

d. *Recall*

Recall atau sensitivitas menunjukkan kemampuan model dalam mendeteksi seluruh data yang benar-benar termasuk dalam kelas positif. Rumus *recall* adalah:

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

Ketika nilai *recall* tinggi berarti model mampu menangkap sebagian besar data positif yang ada.

e. *F1-score*

F1-score merupakan kombinasi antara *precision* dan *recall* dalam bentuk rata-rata harmonik. Nilai *F1-score* diperoleh dari persamaan berikut:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Metrik ini digunakan untuk mengukur konsistensi performa klasifikasi pada setiap kelas secara seimbang, sehingga cocok digunakan pada penelitian dengan klasifikasi multikelas seperti penelitian ini yang melibatkan tujuh kelas denominasi uang kertas.

f. *Inference Time*

Inference time adalah waktu yang dibutuhkan model untuk menghasilkan prediksi terhadap suatu data masukan pada tahap pengujian. Dalam implementasi model *deep learning*, waktu inferensi menjadi indikator efisiensi komputasi, model dengan waktu inferensi yang lebih singkat dinilai lebih efisien dalam implementasi praktis [31].

g. Ukuran Model

Ukuran model merujuk pada jumlah parameter yang dimiliki oleh suatu jaringan saraf. Jumlah parameter berkaitan langsung dengan kebutuhan memori dan kompleksitas komputasi model. Dalam literatur arsitektur *deep learning*, jumlah parameter dan *multiply-adds* (MAdds) sering digunakan untuk menggambarkan *trade-off* antara akurasi dan efisiensi model [31].

h. Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang diciptakan oleh Guido van Rossum pada tahun 1991 [32]. Python merupakan bahasa pemrograman tingkat tinggi (*high-level programming language*) yang bersifat interpretatif, mudah dipelajari, dan memiliki sintaks yang jelas serta ringkas. Python bersifat terbuka (*open-source*) sehingga dapat digunakan dan dikembangkan secara bebas oleh berbagai kalangan. Python digunakan dalam pengembangan aplikasi, analisis data, serta pengembangan *machine learning* dan *deep learning* [20].

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama dalam proses pengolahan dataset citra uang kertas Rupiah, pembangunan model *Convolutional Neural Network* (CNN) *custom* dan *transfer learning* berbasis MobileNetV2, serta evaluasi performa model menggunakan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, Python juga digunakan dalam proses pelatihan model dan pengujian untuk memperoleh hasil klasifikasi yang optimal.

i. Google Colab

Google *Colaboratory* atau Google Colab merupakan platform berbasis cloud yang menyediakan lingkungan komputasi interaktif untuk menjalankan kode Python melalui *notebook* tanpa memerlukan instalasi perangkat lunak secara lokal. Platform ini berbasis *Jupyter Notebook* dan memungkinkan pengguna untuk menulis, menjalankan, serta mendokumentasikan kode dalam satu lingkungan kerja yang terintegrasi. Selain menyediakan CPU sebagai sumber daya komputasi bawaan, Google Colab juga menyediakan akses terhadap sumber daya komputasi seperti GPU dan TPU yang dapat digunakan untuk mempercepat proses pelatihan model *deep learning*.

Google Colab digunakan dalam bidang pendidikan dan penelitian karena kemudahan akses serta kemampuannya dalam mendukung eksperimen komputasi secara interaktif. Dalam penelitian ini, Google Colab digunakan sebagai media pengembangan dan pelatihan model *Convolutional Neural Network (CNN) custom* dan *transfer learning* berbasis MobileNetV2. Platform ini juga digunakan untuk menjalankan proses pengujian, serta evaluasi performa model secara efisien.