

BAB II: TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian sebelumnya antara lain penelitian tahun 2017 yang mengklasifikasi bunga adalah penelitian yang berjudul “Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak” oleh Fitri Muwardi. Penelitian ini melakukan klasifikasi jenis bunga menggunakan metode Manhattan dan Euclidean. Hasil pengujian diperoleh tingkat akurasi paling tinggi sebesar 85% menggunakan metode klasifikasi jarak histogram *manhattan* dan ekstraksi ciri histogram, sedangkan tingkat akurasi terendah diperoleh 77% menggunakan metode jarak *Euclidean* dengan ekstraksi ciri statistic orde 1 [6].

Penelitian tahun 2018 yang berjudul “*Convolutional Neural Network based Transfer learning for Classification of Flowers*” oleh Yong Wu, Xiao Qin, Yonghua Pan, dan Chang Yuan bertujuan melakukan klasifikasi bunga menggunakan perbandingan model VGG-16, VGG-19, Inception-v3, dan ResNet50. Penelitian ini menggunakan dataset Oxford-17 dengan 17 bunga yang masing-masing terdapat 80 data dan Oxford-102 dengan 102 jenis bunga, masing-masing terdapat 40-258 gambar. Dataset dibagi menjadi 3 bagian, yaitu 75% untuk data *train*, 12,5% data *validation*, dan 12,5% data *test*. Pada penelitian ini dilakukan perbandingan model menggunakan 2 dataset, yaitu Oxford 102 dan Oxford 17. Dari hasil perbandingan akurasi yang didapatkan, bahwa model dengan *transfer learning* diperoleh akurasi yang lebih baik. *Convergence rate* pada *transfer learning* Inception-v3 dan Resnet50 lebih cepat dan *network loss* lebih kecil dibandingkan VGG-16 dan VGG-19. Pada penelitian ini, hasil akurasi menggunakan Inception-v3-*transfer* dan ResNet50-*transfer* lebih besar daripada VGG-16-*transfer* dan VGG-19, yaitu 95,88% dan 96,57% [21].

Penelitian lain dilakukan pada tahun 2019 oleh Qian Xiang, Xiaodan Wang, Ruili dengan judul “Fruit Image Classification Based on MobileNetV2 with *Transfer learning* Technique”. Penelitian ini bertujuan untuk mengklasifikasikan

buah-buahan menggunakan *transfer learning* MobileNetV2. Penelitian ini membandingkan model MobileNetV2 dengan MobileNetV1, InceptionV3, dan DenseNet121. Data yang digunakan pada penelitian ini adalah dataset yang terdiri dari 3.670 gambar dari 5 jenis buah. Ukuran yang digunakan sebagai inputan adalah 224 x 224 piksel. MobileNetV2 mendapatkan hasil akurasi yang lebih tinggi daripada MobileNetV1 dan melakukan *training* lebih cepat daripada InceptionV3 dan DenseNet121. MobileNetV2 memiliki jumlah parameter yang lebih sedikit dari MobileNetV1. *Training accuracy* yang didapatkan oleh MobileNetV2 yaitu sebesar 99% sedangkan *Validation Accuracy* yang didapatkan yaitu 85% [32].

Penelitian lain yang melakukan klasifikasi tentang bunga adalah pada tahun 2020 oleh Aditya, Toma Sarker, Md. Meraj, dan Md. Mahbubur dengan judul “Rose Diseases Recognition using MobileNet”. Penelitian ini bertujuan mendeteksi penyakit bunga mawar menggunakan model MobileNet. Penelitian dilakukan dengan menggunakan teknik *transfer learning* dan tanpa *transfer learning*. Data yang digunakan adalah berupa data 4 jenis penyakit bunga mawar yaitu Black spot, Rose dieback, Powdery mildew, dan Rust yang masing-masing terdiri dari 100 gambar dan diaugmentasi menjadi 2000 gambar. Dataset terbagi menjadi 80% data *train* dan 20% data *test*. Hasil evaluasi menggunakan F1 score dan *accuracy*, didapatkan bahwa MobileNet dengan *transfer learning* memiliki nilai akurasi 95,63%, sedangkan pada MobileNet tanpa *transfer learning* didapatkan nilai akurasi sebesar 85,83% [27].

Penelitian tahun 2021 oleh Elok Iedifitra Haksoro dan Abas Setiawan yang melakukan identifikasi jenis jamur yang dapat dikonsumsi menggunakan *transfer learning*. Model *transfer learning* yang digunakan pada penelitian ini adalah VGG19, ResNet50, MobileNets, dan MobileNetV2 untuk mendapatkan hasil akurasi model terbaik dari berbagai aturan *Learning Rate* yang digunakan pada proses *pre-training* dan *fine-tuning*. Setiap model dilatih dua kali dengan nilai *learning rate* yang berbeda yaitu 0,00001 dan 0,00005 pada tahap *pre-training* dan 0,0001 dan 0,0005 pada *fine-tune*. Dari 8 percobaan, didapatkan hasil akurasi terbaik, yaitu 92,19% menggunakan model MobileNetV2 di tahap *pre-training*

dengan *learning rate* 0,00001 dan *fine-tune* dengan *learning rate* 0,0001. Hal ini terjadi karena terdapat *inverted residual connections* dan *linear bottleneck* pada arsitektur MobileNetV2 yang dapat meningkatkan akurasi [31].

Penelitian pada tahun 2021 oleh Widi Hastomo, Sugiyanto, dan Sudjiran yang melakukan deteksi tumor menggunakan metode CNN dengan arsitektur MobileNetV2. MobileNetV2 dipilih karena menghasilkan akurasi yang tinggi dan ringan serta mampu dijalankan di komputer yang performansinya tidak terlalu tinggi. Dataset yang digunakan adalah 2.870 image yang terdiri dari 4 kelas, yaitu Glioma, Meningioma, Nos tumor, dan Pituitary. Pada penelitian ini dilakukan *training* menggunakan *epoch* sebanyak 100 dan menghasilkan akurasi sebesar 97%. Pengujian dilakukan pada keseluruhan dataset input dan diperoleh akurasi sebesar 94%. Pengukuran(*metric*) menggunakan *confusion matrix* berupa akurasi, presisi, *recall*, dan *f1-score* yang kemudian didapatkan akurasi sebesar 0.94. Sehingga dengan nilai akurasi yang sangat baik, model dapat digunakan sebagai prediksi tumor otak [30].

Kajian lain yang membahas tentang pengenalan bunga dilakukan pada tahun 2021 oleh Hanissa, Margi, dan Missa. Penelitian ini menggunakan metode CNN untuk mengklasifikasikan jenis bunga dan diimplementasikan pada aplikasi identifikasi jenis tanaman berbunga. Dataset yang digunakan berasal dari website Google Image sebanyak 460 gambar bunga yang terdiri dari 5 jenis bunga, yaitu Mawar, Matahari, Kembang Sepatu, Telang, dan *Cornflower*. *Training* dataset dijalankan 30 kali, setiap 13 langkah dihitung 1 *training* dan menghasilkan nilai akurasi data yang telah dilatih (*val_acc*) dan akurasi data yang hilang(*val_loss*) . Hasil *test accuracy* diperoleh sebesar 100% dengan salah satu hasil pengujian klasifikasi bunga mawar diperoleh akurasi 99,30% [15].

Penelitian M. Raihan dan Agung Toto tahun 2021 membangun sebuah sistem klasifikasi citra otomatis untuk genus tanaman anggrek. Metode klasifikasi yang digunakan dalam penelitian ini adalah CNN yang kemudian di-*deploy* ke dalam aplikasi berbasis website. Dalam penelitian ini, dataset yang digunakan berupa gambar/citra 5 genus tanaman anggrek, yaitu *Cattleya*, *Dendrobium*, *Oncidium*,

Phalaenopsis dan *Vanda*. Dataset terbagi menjadi 80% data *train* dan 20% data *test*. Skenario arsitektur CNN yang digunakan untuk *training* yaitu arsitektur kustom CNN dan arsitektur MobileNetV2 dengan hyperparameter *learning rate* 0,0003 dan 0,0001. Dari hasil pengujian yang dilakukan pada semua arsitektur didapatkan *score* akurasi tertinggi oleh model MobileNetV2 dimana diperoleh akurasi sebesar 90,44% dari *testing* lapangan dan 80,54% dari *testing* internet dan *F1-score* sebesar 98% [33].

Penelitian lain yang melakukan klasifikasi citra adalah penelitian yang dilakukan pada tahun 2021 oleh Mohtar Khoiruddin, Apri Junaidi, dan Wahyu Andi Saputra yang melakukan klasifikasi penyakit daun tanaman padi menggunakan algoritma CNN. Dataset yang digunakan berjumlah 6.000 yang dibagi menjadi 80% data *training*, 10% data validasi, dan 10% data *testing*. Dari hasil *training* yang dilakukan menggunakan *epoch* 25, 50, 75, dan 100 diperoleh hasil akurasi *training* terbaik di *epoch* 100 dengan akurasi 98%. Diperoleh akurasi *testing* menggunakan *confusion matrix* sebesar 98%. Algoritma CNN menghasilkan akurasi yang tinggi, dalam melakukan klasifikasi penyakit daun padi [36].

Penelitian tahun 2021 oleh Abdul Jalil Rozaqi, Andi Sunyoto, dan Muhammad Rudyanto Arief berjudul “Implementasi *Transfer learning* pada Algoritma *Convolutional Neural Network* Untuk Identifikasi Penyakit Daun Kentang” bertujuan untuk mengklasifikasikan penyakit pada daun kentang. Metode yang digunakan dalam penelitian ini adalah *transfer learning* model VGG-16, Inception-V3, dan ResNet-50. Dataset yang digunakan adalah citra daun penyakit kentang yang terdiri dari 3 kelas, yaitu *early blight*, *late blight*, dan daun sehat yang berjumlah 450 citra yang kemudian di-*preprocessing* lalu dibagi menjadi 80% untuk data *training*, 10% data *validation*, dan 10% data *testing*. Diperoleh model terbaik yaitu VGG-16 dengan hasil akurasi yang diperoleh pada proses *training* adalah sebesar 95% [24].

Tabel 2.1 Tabel penelitian sebelumnya

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
1.	Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak [6].	Penelitian ini membuat sebuah sistem pengenalan jenis bunga menggunakan metode <i>manhattan</i> dan <i>Euclidean</i> . Data yang digunakan berupa citra bunga yang terdiri dari 9 jenis data jenis bunga yaitu bunga alamanda, kamboja, kenanga, lidah mertua, sepatu, mawar, melati, matahari, dan lili putih	Penelitian ini dilakukan untuk mengidentifikasi citra jenis bunga menggunakan metode klasifikasi jarak <i>manhattan</i> dengan ekstraksi ciri histogram dan metode <i>Euclidean</i> dengan ekstraksi ciri statistic orde 1. Data yang digunakan berupa citra bunga yang terdiri dari 9 jenis data jenis bunga yaitu bunga alamanda, kamboja, kenanga, lidah mertua, sepatu, mawar, melati, matahari, dan lili putih. Data pengujian digunakan 225 sampel pada masing-masing jenis bunga.	.	Penelitian ini diawali dengan pengambilan citra bunga yang kemudian di <i>preprocessing</i> . Selanjutnya dilakukan ekstraksi ciri menggunakan statistic orde 1 dan histogram. Kemudian dilakukan pengklasifikasian menggunakan metode <i>Manhattan</i> dan <i>Euclidean</i> .	Hasil pengujian pada penelitian ini diperoleh tingkat akurasi tertinggi adalah menggunakan metode klasifikasi jarak histogram <i>manhattan</i> dan ekstraksi ciri histogram sebesar 85%, sedangkan tingkat akurasi terendah adalah sebesar 77% dengan menggunakan metode jarak <i>Euclidean</i> dan ekstraksi ciri statistik orde 1.
2.	<i>Convolutional Neural Network based Transfer learning for Classification of Flowers</i> [21].	Penelitian ini melakukan klasifikasi jenis bunga menggunakan <i>transfer learning</i> model pada CNN. Dataset yang digunakan adalah Dataset Oxford-17 yang terdiri dari 17 jenis bunga	Penelitian ini menggunakan model <i>transfer learning</i> VGG-16, VGG-19, Inception-V3, dan ResNet50 untuk melakukan perbandingan akurasi pada klasifikasi jenis bunga menggunakan	Penelitian ini tidak melakukan tahap <i>preprocessing</i> data.	Penelitian ini dimulai dengan akuisisi data yang kemudian membagi data menjadi 3, yaitu data <i>training</i> , data <i>validation</i> , dan data <i>test</i> . Selanjutnya dilakukan <i>training</i> model menggunakan VGG-16,	Hasil akurasi menggunakan <i>Inception-v3-transfer</i> dan <i>ResNet50-transfer</i> lebih besar daripada <i>VGG-16-transfer</i> dan VGG-19, yaitu 95,88% dan 96,57%.

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
		masing-masing terdapat 80 data dan Dataset Oxford-102 yang terdiri dari 102 jenis bunga masing-masing terdapat 40-258 data. Dataset terbagi menjadi 3 yaitu data <i>train</i> , <i>validation</i> , dan <i>test</i> .	2 dataset yang berbeda. Dataset yang digunakan adalah Dataset Oxford-17 yang terdiri dari 17 jenis bunga masing-masing terdapat 80 data dan Dataset Oxford-102 yang terdiri dari 102 jenis bunga masing-masing terdapat 40-258 data. Dataset terbagi menjadi 3 yaitu data <i>train</i> , <i>validation</i> , dan <i>test</i> .		VGG-19, Inception-V3, dan ResNet50 untuk dilakukan perbandingan akurasi.	
3.	Fruit Image Classification Based on MobileNetV2 with <i>Transfer learning</i> Technique [32].	Penelitian ini melakukan klasifikasi jenis buah menggunakan model MobileNetV2. Dataset yang digunakan berupa Dataset yang berisi 3.670 gambar dari 5 jenis buah, yaitu apel, pisang, carambola, jambu, dan kiwi. Dataset terbagi menjadi 3.213 data <i>train</i> dan 457 data <i>validation</i> .	Penelitian ini melakukan klasifikasi jenis buah menggunakan beberapa model <i>transfer learning</i> , yaitu MobileNetV2, MobileNetV1, InceptionV3, dan DenseNet121 untuk mendapatkan model terbaik. Dataset yang digunakan berupa Dataset yang berisi 3.670 gambar dari 5 jenis buah, yaitu apel, pisang, carambola, jambu, dan kiwi. Dataset terbagi menjadi 3.213 data <i>train</i> dan 457 data <i>validation</i> .	Penelitian ini tidak melakukan tahap <i>preprocessing</i> data.	Penelitian ini diawali dengan pengumpulan data citra buah yang kemudian dilakukan split data <i>training</i> dan <i>validation</i> . Selanjutnya membuat base model dari pre <i>trained</i> model MobileNetV2, melakukan <i>training</i> model pada <i>pre training stage</i> dan fine tuning beberapa layer menggunakan <i>dropout</i> dan <i>learning rate</i>	MobileNetV2 mendapatkan hasil akurasi yang lebih tinggi daripada MobileNetV1 dan melakukan <i>training</i> lebih cepat daripada InceptionV3 dan DenseNet121. MobileNetV2 memiliki jumlah parameter yang lebih sedikit dari MobileNetV1. <i>Training accuracy</i> yang didapatkan oleh MobileNetV2 yaitu sebesar 99% sedangkan <i>Validation Accuracy</i> yang

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
						didapatkan yaitu 85% dengan runtime 327,49 detik.
4.	Rose Diseases Recognition using MobileNet [27].	Penelitian ini melakukan deteksi penyakit pada bunga mawar menggunakan MobileNet. Data yang digunakan berupa Data gambar 4 jenis penyakit bunga mawar, yaitu Black spot, Rose dieback, Powdery mildew, dan Rust yang terdiri dari 400 gambar diaugmentasi menjadi 2000 gambar. Dataset terbagi menjadi 80% data <i>train</i> dan 20% data <i>test</i> .	Penelitian ini menggunakan model MobileNet dengan <i>transfer learning</i> dan MobileNet tanpa <i>transfer learning</i> untuk mengklasifikasi penyakit bunga mawar. Data yang digunakan berupa Data gambar 4 jenis penyakit bunga mawar, yaitu Black spot, Rose dieback, Powdery mildew, dan Rust yang terdiri dari 400 gambar diaugmentasi menjadi 2000 gambar.	Evaluasi model yang dilakukan hanya dihitung pada metrics <i>accuracy</i> dan <i>f1 score</i> saja.	Penelitian dilakukan dengan akuisisi data 4 jenis penyakit bunga mawar, kemudian dilakukan <i>preprocessing</i> data. Selanjutnya dilakukan <i>training</i> data menggunakan arsitektur MobileNet dengan <i>transfer learning</i> dan tanpa <i>transfer learning</i> . Selanjutnya dilakukan evaluasi menggunakan <i>confusion matrix</i> untuk melihat performance model yang telah dibuat.	Hasil evaluasi menggunakan F1 score dan <i>accuracy</i> , didapatkan bahwa MobileNet dengan <i>transfer learning</i> memiliki nilai akurasi 95,63%, sedangkan pada MobileNet tanpa <i>transfer learning</i> didapatkan nilai akurasi sebesar 85,83%.
5.	Pengenalan Jamur yang Dapat Dikonsumsi Menggunakan Metode <i>Transfer learning</i> Pada <i>Convolutional Neural Network</i> [31].	Penelitian ini dilakukan untuk mengenali jenis jamur yang dapat dikonsumsi atau beracun yang ada di Indonesia menggunakan metode <i>transfer learning</i> CNN. Data yang digunakan adalah 9 jenis jamur yang dikelompokkan menjadi 2 kelas, yaitu 4 jenis jamur dapat dikonsumsi dan 5 jamur tidak bisa dikonsumsi karena	Penelitian ini dilakukan untuk melakukan perbandingan model <i>transfer learning</i> VGG19, ResNet50, MobileNets, dan MobileNetV2 untuk mendapatkan hasil akurasi model terbaik dari berbagai aturan <i>Learning Rate</i> yang digunakan pada proses <i>pre-training</i> dan <i>fine tuning</i> .	Pada tahap <i>preprocessing</i> tidak disebutkan jumlah data hasil augmentasi	Pada penelitian ini dimulai dengan pengambilan data dari kaggle berupa 9 jenis jamur yang dikelompokkan menjadi 2 kelas, kemudian membagi data menjadi 70% data latih dan 30% data uji, selanjutnya dilakukan <i>preprocessing</i> dan dilakukan <i>training</i> data menggunakan <i>transfer learning</i> VGG19,	Hasil akurasi terbaik diperoleh model MobileNetV2 yaitu sebesar 92,19% di tahap <i>fine-tune</i> dengan <i>learning rate</i> 0,0001. Hasil akurasi terendah yaitu menggunakan model VGG-19 di tahap <i>pre-training</i> sebesar 86,46% dengan <i>learning rate</i> 0,0001.

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
		beracun yang masing-masing terdiri dari 642 gambar. Dataset terbagi menjadi 70% data latih dan 30% data uji.			ResNet50, MobileNets, dan MobileNetV2 untuk mendapat akurasi model terbaik. Setiap model dilatih dua kali dengan nilai <i>learning rate</i> yang berbeda yaitu 0,00001 dan 0,00005 pada tahap <i>pre-training</i> dan 0,0001 dan 0,0005 pada <i>fine-tune</i> .	
6.	<i>Convolutional Neural Network</i> Arsitektur MobileNet-V2 untuk Mendeteksi Tumor Otak [30].	Penelitian ini melakukan deteksi tumor otak untuk memprediksi pasien terhadap penyakit tumor otak. Dataset yang digunakan berisi 2.870 image yang terdiri dari 4 kelas, yaitu Glioma, Meningioma, Nos tumor, dan Pituitary.	Penelitian ini dilakukan untuk mendeteksi tumor otak untuk memprediksi pasien terhadap penyakit tumor otak menggunakan arsitektur MobileNetV2. Dataset yang digunakan berupa hasil <i>CT Scan</i> sebanyak 2870 data image otak yang terdiri dari 4 kelas.	Pada penelitian ini tidak dilakukan tahap <i>preprocessing</i> data	Penelitian dimulai dengan pengambilan dataset sebanyak 2870 yang terdiri dari 4 kelas. Kemudian dilakukan <i>training</i> model MobileNetV2. Selanjutnya pengujian atau evaluasi model menggunakan <i>confusion matrix</i> .	Hasil <i>training</i> dengan <i>epoch</i> 100 dan dataset sebanyak 2.870 didapatkan nilai akurasi sebesar 97% dan hasil akurasi hasil pengujian (<i>testing</i>) menggunakan <i>Confusion matrix</i> adalah 94% dengan pengukuran <i>metrics</i> akurasi, presisi, <i>Recall</i> , dan <i>f1-score</i> .
7.	Implementasi Deep Learning Flower Scanner Menggunakan Metode <i>Convolutional Neural Network</i> [15].	Penelitian ini mengklasifikasikan jenis bunga menggunakan metode CNN. Dataset yang digunakan adalah 460 gambar dengan 5 jenis bunga, yaitu Mawar, Matahari, Kembang Sepatu, Telang, dan <i>Cornflower</i> . Hasil dari	Penelitian ini menggunakan metode CNN untuk mengklasifikasi jenis bunga. Dataset yang digunakan adalah 460 gambar dengan 5 jenis bunga, yaitu Mawar, Matahari, Kembang	Pada penelitian ini tidak dilakukan tahap <i>preprocessing</i> data	Penelitian dimulai dengan pengumpulan dataset bunga sebanyak 460 gambar. Selanjutnya dilakukan <i>training</i> model sebanyak 30 kali. Kemudian mengekspor model ke ekstensi .tflite untuk diimpor ke dalam pembuatan aplikasi	Hasil <i>training</i> dataset dengan 460 gambar dan <i>epoch</i> 30, didapatkan nilai akurasi data yang telah dilatih (<i>val_acc</i>) sebesar 0,97 dan nilai akurasi data yang hilang (<i>val_loss</i>) sebesar 0,087. Hasil

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
		pembuatan model klasifikasi akan diimplementasikan ke dalam pembuatan aplikasi pengenalan jenis bunga.	Sepatu, Telang, dan <i>Cornflower</i> .		pengenalan jenis bunga. Selanjutnya dilakukan testing pada aplikasi klasifikasi jenis bunga.	<i>test accuracy</i> diperoleh sebesar 100% dengan salah satu hasil pengujian klasifikasi bunga mawar diperoleh akurasi 99,30%.
8.	Klasifikasi Genus Tanaman Angrek Menggunakan Metode <i>Convolutional Neural Network</i> (CNN) [33].	Penelitian ini membuat sebuah sistem untuk klasifikasi citra genus tanaman angrek berbasis website menggunakan metode CNN. Dataset yang digunakan berupa data citra 5 genus tanaman angrek, yaitu <i>Cattleya</i> , <i>Dendrobium</i> , <i>Oncidium</i> , <i>Phalaenopsis</i> dan <i>Vanda</i> sebanyak 6685 citra. Dataset terbagi menjadi data <i>train</i> dan data <i>test</i> , dimana data <i>train</i> dibagi lagi menjadi 80% data <i>train</i> dan 20% data <i>validation</i> .	Pada penelitian ini digunakan arsitektur CNN kustom dan MobileNetV2 untuk mengklasifikasikan genus tanaman angrek.		Proses klasifikasi pada penelitian ini dimulai dengan pengumpulan dataset citra tanaman angrek, kemudian dilakukan augmentasi data. Selanjutnya, splitting data <i>train</i> dan data <i>test</i> yang kemudian data <i>train</i> dipecah menjadi data <i>train</i> dan data <i>validation</i> . Selanjutnya dilakukan <i>training</i> menggunakan 4 arsitektur CNN yang berbeda dengan menerapkan <i>hyperparameter</i> berupa <i>learning rate</i> yang selanjutnya dilakukan <i>testing</i> dan evaluasi terhadap data <i>test</i> untuk dipilih model terbaik yang dikirimkan ke aplikasi.	Dari hasil <i>training</i> dan <i>testing</i> menghasilkan 2 model CNN terbaik, yaitu MobileNetV2 + <i>Dropout</i> dengan <i>hyperparameter Adam Optimizer</i> dan <i>Learning rate</i> 0,0001 yaitu akurasi <i>testing</i> lapangan 90,44% dan <i>testing</i> internet 80,54%, ukuran model 101MB. Sedangkan MobileNetV2 Orisinal didapatkan <i>testing</i> lapangan 83,12% dan <i>testing</i> internet 80% dengan ukuran 9 MB menggunakan <i>Adam Optimizer</i> dan <i>Learning rate</i> 0,0003.
9.	Klasifikasi Penyakit Daun Padi	Penelitian ini melakukan klasifikasi penyakit daun padi menggunakan	Penelitian ini menggunakan algoritma CNN untuk		Penelitian diawali dengan pengumpulan dataset citra daun padi dari <i>Kaggle</i>	Hasil akurasi <i>training</i> terbaik diperoleh <i>epoch</i> 100 dengan

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Menggunakan <i>Convolutional Neural Network</i> [36].	algoritma CNN. Dataset yang digunakan berjumlah 240 dari 3 kelas penyakit daun padi yang diaugmentasi menjadi 6.000 citra dan dibagi menjadi 80% data <i>train</i> , 10% data validasi, dan 10% data testing	mengklasifikasikan daun padi dan melihat pengaruh <i>epoch</i> terhadap akurasi yang dihasilkan.		yang selanjutnya dilakukan <i>preprocessing</i> . Kemudian dilakukan perancangan model CNN dengan proses <i>training</i> menggunakan <i>epoch</i> 25, 50, 75, dan 100. Lalu diuji menggunakan data <i>testing</i> .	hasil akurasi 98% dan dilakukan testing menggunakan <i>confusion matrix</i> dengan akurasi testing 98%.
10.	Implementasi <i>Transfer learning</i> pada Algoritma <i>Convolutional Neural Network</i> Untuk Identifikasi Penyakit Daun Kentang [24].	Penelitian ini melakukan klasifikasi penyakit daun kentang menggunakan metode CNN. Dataset yang digunakan berupa Data citra daun kentang yang terdiri dari 3 kelas, yaitu <i>Early blight</i> , <i>late blight</i> , dan daun sehat yang berjumlah 450 Dataset terbagi menjadi 80% data <i>training</i> , 10% data <i>validation</i> , dan 10% data <i>testing</i> .	Penelitian ini melakukan klasifikasi penyakit daun kentang dengan menggunakan arsitektur CNN sederhana dan beberapa model <i>transfer learning</i> Inception-V3, VGG-16, dan ResNet-50.	Penelitian ini tidak dilakukan proses augmentasi data.	Penelitian ini dimulai dengan pengumpulan data citra daun kentang yang kemudian di- <i>preprocessing</i> . Selanjutnya dilakukan klasifikasi menggunakan CNN dan <i>transfer learning</i> model. Kemudian dilakukan pengujian model menggunakan <i>confusion matrix</i>	Hasil percobaan klasifikasi menggunakan metode <i>transfer learning</i> antara model VGG-16, Inception-V3 dan ResNet-50 menghasilkan bahwa model VGG-16 memiliki hasil yang paling baik, dibuktikan dengan performa model yang stabil dan akurasi yang paling tinggi dengan jumlah 95%

Berdasarkan penelitian terdahulu, *transfer learning* MobileNetV2 mampu menghasilkan akurasi yang tinggi untuk melakukan berbagai klasifikasi. Selain memiliki akurasi yang tinggi, MobileNetV2 mempercepat proses *training* model [34]. Terdapat penelitian-penelitian terdahulu yang membahas tentang klasifikasi bunga dengan menggunakan berbagai dataset dan metode diantaranya yaitu

metode *Manhattan* dan *Euclidean* [6], CNN [15] dan VGG-16, VGG-19, Inception-V3, serta ResNet50 [21]. Terdapat beberapa penelitian dalam berbagai kasus menggunakan MobileNetV2 di berbagai bidang, seperti klasifikasi jenis jamur [31], deteksi tumor otak [30], dan klasifikasi genus tanaman anggrek [33], serta berbagai klasifikasi lainnya. *Transfer learning* MobileNetV2 dapat mempercepat proses pelatihan. Jumlah parameter juga berkurang oleh *depth separable convolutions* dan *inverted residual linear bottlenecks* pada MobileNetV2 [33]. Oleh karena itu, pada penelitian ini akan mengimplementasikan *transfer learning* MobileNetV2 untuk klasifikasi jenis bunga pada *dry flower*.

2.2. Landasan Teori

Berikut ini merupakan teori-teori yang menunjang pada penelitian klasifikasi bunga yang akan dilakukan.

2.2.1. *Dry flower*

Bunga atau dalam bahasa latin disebut *flos* adalah alat reproduksi seksual pada tanaman berbunga (*divisio Magnoliophyta* atau *Angiospermae*, "tanaman berbiji tertutup") [15]. Bunga adalah bagian tumbuhan yang dihasilkan dari perubahan pucuk (batang dan daun) sehingga bentuk, warna, dan susunannya menyesuaikan dengan manfaat tumbuhan tersebut [6]. Bunga adalah bagian dari tumbuhan yang digunakan untuk menghasilkan biji dimana penyerbukan dan pembuahan suatu tanaman terjadi di dalam bunga yang kemudian akan berkembang membentuk buah [16].

Terdapat ratusan ribu spesies dan varietas bunga dengan berbagai bentuk, warna, dan struktur yang menarik [4]. Keragaman bentuk yang indah dan keestetikan yang dimiliki, menjadikan bunga banyak digunakan sebagai kerajinan tangan seperti buket bunga atau dekorasi ruangan [7]. Bunga yang dapat digunakan sebagai kerajinan tangan adalah bunga alami dan buatan dimana bunga alami terbagi ke dalam beberapa kategori, yaitu *fresh flower* (bunga segar), *dried flower* (bunga kering), dan *preserved flower* (bunga awetan) [8].

Dried flower atau bunga kering adalah bunga segar yang diawetkan dengan dikeringkan atau didehidrasi secara alami dengan cara digantung sampai benar-benar kering [9]. Kelebihan yang dimiliki bunga kering adalah tidak membutuhkan air dan tidak bisa layu sehingga sering dijadikan sebagai kerajinan untuk dekorasi rumah dan *bouquet* bunga, tetapi kelemahannya yaitu tangkai yang mudah patah, kelopaknya mudah gugur, dan warnanya cepat pudar [11]. Beberapa jenis bunga pada bunga kering (*dry flower*) diantaranya adalah *pampas*, *lagurus*, *cotton flower*, *baby breath*, dan *setaria* [14].

Pada penelitian ini, akan dilakukan klasifikasi terhadap 5 kelas jenis bunga pada *dry flower*, diantaranya sebagai berikut:

2.2.1.1. *Pampas*

Cortaderia selloana atau pampas grass adalah rumput hias berukuran besar yang berasal dari Amerika Selatan. *Pampas* pertama kali diperkenalkan di Montpellier Botanical Garden pada tahun 1857 [37]. Ciri-ciri yang dimiliki oleh tanaman ini adalah tinggi, ramping, dan panjangnya mencapai lebih dari satu meter. Bagian paling mencolok pada tanaman ini adalah bulu-bulu putih keperakan yang tinggi dan lembut [38]. Pada gambar 2.1 dapat dilihat contoh bunga *pampas*.



Gambar 2.1 *Pampas* [14]

2.2.1.2. *Lagurus*

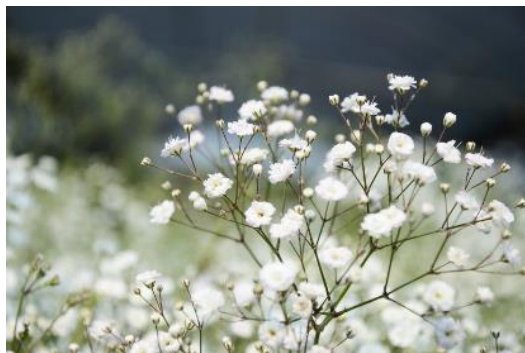
Lagurus ovatus atau biasa disebut *bunny tail* merupakan rumput tahunan dari keluarga *Poaceae* yang berasal dari pantai wilayah Mediterania. *Lagurus* dibudidayakan sebagai tanaman hias dan umumnya digunakan dalam rangkaian bunga kering. *Lagurus* telah dinaturalisasi di beberapa wilayah di Australia dan Inggris serta dianggap sebagai spesies invasif di beberapa tempat di luar daerah asalnya. *Lagurus* membentuk rumpun yang memiliki tinggi sekitar 1-2 kaki. Bentuk rumput ini memiliki bunga yang oval seperti paku yang lembut dengan bulu [39]. Contoh bunga *lagurus* dapat dilihat pada gambar 2.2.



Gambar 2.2 *Lagurus* [40]

2.2.1.3. *Baby Breath*

Semua bunga *baby breath* berasal dari Eropa, Asia, dan Afrika Utara. *Baby breath* diperkenalkan ke Amerika Serikat pada akhir 1800-an. *Baby breath* memiliki nama ilmiah *Gypsophila paniculata*, dimana *paniculata* mencerminkan bagaimana cara bunga ini tumbuh sedangkan *Gypsophila* berasal dari habitat alami tanaman ini yang sering menampilkan tanah yang kaya akan gypsum (kapur) [41]. *Gypsophila* merupakan tanaman hias bunga yang berasal dari keluarga anyelir, *Caryophyllaceae*. *Baby breath* banyak dibudidaya dan menjadi komoditas kualitas ekspor di beberapa negara seperti Peru. *Baby breath* memiliki daya tahan yang lama dan lebih awet sehingga dapat dijadikan sebagai dekorasi karangan bunga [42]. *Baby breath* memiliki kelopak yang berukuran kecil yang tumbuh dengan 3 varian warna yaitu merah muda, kuning, dan putih [43]. Pada gambar 2.3 menunjukkan bunga *baby breath*.



Gambar 2.3 Bunga *baby breath* [40]

2.2.1.4. *Cotton Flower*

Gossypium hirsutum atau yang dikenal dengan bunga kapas (*cotton flower*) adalah sub semak tahunan dari keluarga mallow yang bervariasi pada cara pertumbuhannya. *Gossypium* memiliki tinggi mencapai sekitar 5 kaki dengan 2 cabang, yaitu cabang yang menghasilkan bunga dan buah. *Cotton flower* memiliki ciri bunga yang besar berbentuk cangkir kuning atau putih dan diikuti oleh buah kapas yang berisi serat dan biji [44]. Gambar 2.4 merupakan contoh dari bunga *cotton flower*.



Gambar 2.4 *Cotton Flower* [45]

2.2.1.5. *Setaria*

Setaria viridis adalah spesies rumput yang dikenal di sebagian besar benua sebagai spesies *Setaria italica*. *Setaria* adalah rumput tahunan yang memiliki batang tegak dengan panjang 1-2 meter [46]. *Setaria viridis* memiliki bentuk yang mirip dengan lagurus yaitu memiliki bulu yang lembut, tetapi pada *setaria* hanya memiliki sedikit bulu [14]. Contoh bunga *setaria* dapat dilihat pada gambar 2.5.

Gambar 2.5 *Setaria* [14].

2.2.2. *Machine Learning*

Machine learning atau pembelajaran mesin adalah cabang dari *Artificial Intelligence* yang digunakan untuk menggantikan atau meniru perilaku manusia menyelesaikan masalah atau melakukan otomatisasi [47]. *Machine learning* merepresentasikan data-data menggunakan algoritma pembelajaran dalam menangani dan memprediksi data yang besar [48]. *Machine learning* memiliki perbedaan dengan *traditional programming*, dimana pada pemrograman tradisional, data dan program diperlukan sebagai input oleh komputer agar menghasilkan output. Sedangkan pada *machine learning*, komputer membutuhkan data dan output agar menghasilkan sebuah program [49]. Pada gambar 2.6 menunjukkan perbedaan konsep dari *machine learning* dan *traditional programming*.

Gambar 2.6 Konsep *Machine Learning* dan *Traditional Programming* [50]

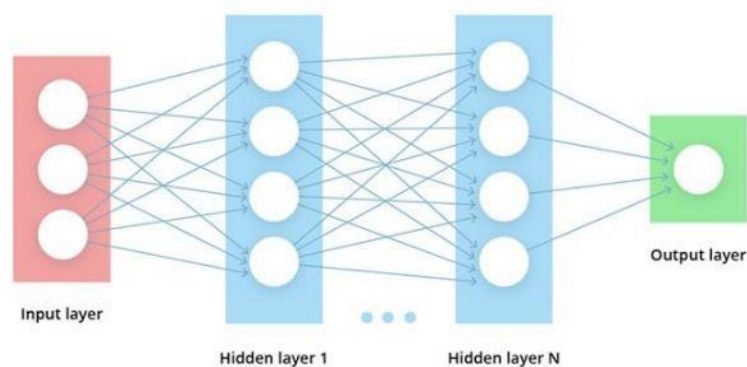
Machine learning memiliki ciri khas yaitu proses pelatihan, pembelajaran, atau *training*. Data pelatihan diperlukan agar dapat diimplementasikan teknik-teknik *machine learning*. Terdapat 2 data yang biasanya digunakan pada algoritma

machine learning, yaitu data *training* dan data *testing*. Data pelatihan atau data *training* digunakan untuk melatih algoritma dan data uji atau *testing* digunakan untuk menentukan kinerja algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat [18].

Machine learning digunakan untuk klasifikasi dan prediksi. Klasifikasi adalah salah satu metode pada *machine learning* untuk mengklasifikasikan objek berdasarkan karakteristik tertentu. Prediksi atau regresi digunakan untuk memprediksi output dari data input berdasarkan data yang sudah dilatih [48]. Cara pembelajaran pada *machine learning* ada beberapa skenario yang dapat dilakukan, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Supervised Learning* adalah pembelajaran dimana data *training* berisi keluaran yang telah ditentukan dan diberi label. *Unsupervised Learning*, data pembelajaran yang tidak diberi label. *Reinforcement Learning* adalah fase pembelajaran dan tes saling dicampur [51].

2.2.3. *Deep Learning*

Deep Learning adalah cabang dari *machine learning* yang merupakan pengembangan Jaringan Saraf Tiruan (JST) [15]. Komputer dapat mengklasifikasikan suatu objek melalui gambar, suara, teks, atau video dengan menggunakan teknik *deep learning* [52]. Berdasarkan gambar 2.7, *Deep Learning* memiliki 3 lapisan atau *layer*, yaitu *Input Layer*, *Hidden Layer*, dan *Output Layer*. *Input Layer* berisi node-node yang menyimpan sebuah nilai masukan pada fase latih dan hanya dapat berubah ketika diberikan nilai masukan baru. Semua proses pelatihan dan pengenalan dijalankan pada *hidden layer*, dimana jumlah lapisan ini tergantung dari arsitektur yang akan dirancang untuk menemukan komposisi algoritma yang tepat sehingga dapat meminimalisir *error* pada *output*. Sedangkan *output layer* berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada lapisan *hidden layer* berdasarkan *input* yang diterima [53].

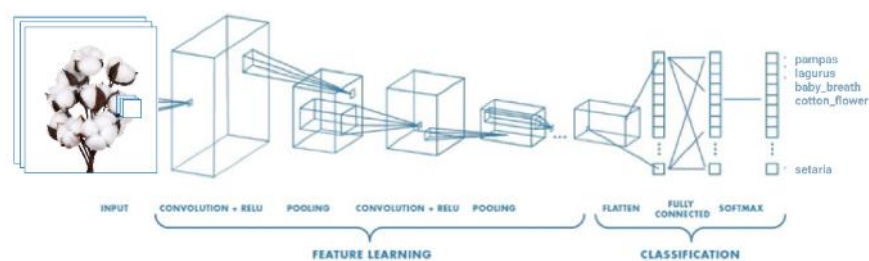


Gambar 2.7 Layer pada *Deep Learning* [54].

Deep Learning memungkinkan model komputasi yang terdiri dari beberapa *processing layer* untuk mempelajari representasi data dengan berbagai macam tingkat abstraksi [20]. *Deep Learning* telah dimanfaatkan untuk memecahkan masalah seperti *computer vision*, *speech recognition*, dan *natural language processing* [51].

2.2.4. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah salah satu algoritma *Deep Learning* yang berasal dari pengembangan *Multi Layer Perceptron (MLP)* dan dirancang untuk mengolah data dua dimensi seperti gambar atau suara [15]. CNN adalah metode *Supervised Learning* yang digunakan untuk mengklasifikasikan data yang diberi label karena *supervised learning* melatih data dan variabel target. CNN banyak digunakan untuk mengenali atau mendeteksi objek [33].



Gambar 2.8 Arsitektur *Convolutional Neural Network (CNN)*

Struktur CNN terdiri dari ekstraksi fitur dan klasifikasi seperti yang ditunjukkan pada gambar 2.8. Proses ekstraksi fitur dari citra dilakukan pada *convolutional layer* atau lapisan konvolusional selanjutnya mengurangi dimensi dan waktu komputasi di *pooling layer*. Arsitektur ini dapat mencapai bentuk regulasinya sendiri. Hasil ekstraksi fitur kemudian dimasukkan ke dalam lapisan *softmax* untuk proses klasifikasi [4].

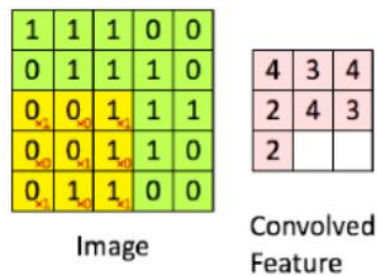
Proses pada CNN dimulai dengan input data berupa citra yang ditangkap dari setiap piksel citra berukuran panjang x lebar x 1 untuk citra *grayscale* dan panjang x lebar x 3 untuk citra dengan warna RGB. Kemudian dilakukan ekstraksi fitur dari citra dengan melakukan “*encoding*” menjadi *features* berupa angka-angka yang merepresentasikan citra gambar tersebut. Proses *feature extraction* atau ekstraksi fitur memiliki 2 bagian utama, yaitu *convolutional layer* dan *pooling layer* yang dapat disesuaikan dengan kebutuhan. Semakin dalam arsitektur, semakin baik hasil akurasi klasifikasi [51].

Hasil dari proses *feature extraction* adalah *feature map* yang berbentuk array multidimensi, sehingga perlu menjalankan proses *flatten* menjadi vector sebagai input dari *fully connected layer*. Pada proses *classification* terdapat beberapa *hidden layer*, *activation function*, dan *loss function*. *Softmax classifier* mengubah angka atau log menjadi probabilitas yang berjumlah satu dan menghasilkan vektor yang mewakili probabilitas dari daftar label yang potensial [33]. Berikut ini penjelasan mengenai *layer* pada arsitektur CNN.

2.2.4.1. *Convolutional Layer*

Convolutional layer merupakan *layer* pertama yang mengekstraksi fitur dari citra *input* dengan melakukan operasi konvolusi pada *output* dari lapisan sebelumnya [55]. *Convolutional layer* menjadi lapisan utama yang paling penting pada CNN. Secara matematis, konvolusi adalah mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dari sebuah *filter* seperti *filter 3x3* digunakan pada citra 8x8, *filter* akan menghitung pada lokasi piksel yang sekarang, kemudian bergeser ke kanan, menghitung kembali sampai semua posisi piksel

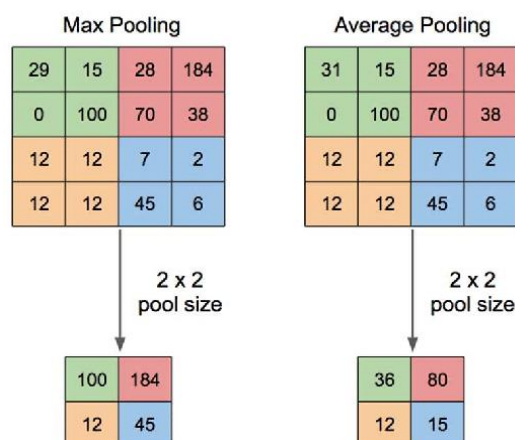
terhitung [17]. Hasil konvolusi berbentuk transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada *layer* menspesifikasikan *kernel* konvolusi yang digunakan, sehingga dapat dilatih berdasarkan input pada CNN [15]. Proses konvolusi dapat dilihat pada gambar 2.9.



Gambar 2.9 Proses Konvolusi CNN [53]

2.2.4.2. Pooling Layer

Pooling layer akan mereduksi ukuran data citra yang terlalu besar yang diterima dari convolutional layer sehingga data menjadi lebih kecil dan mudah dikelola serta mempertahankan informasi penting [55]. Layer ini akan memotong input menjadi beberapa grid. Pooling layer memiliki beberapa tipe seperti *max pooling* dan *average pooling*. Metode yang paling sering digunakan adalah metode *max pooling*. *Max pooling* mengambil nilai maksimal dari setiap grid, sedangkan *average pooling* mengambil nilai rata-rata dari bagian-bagian piksel pada citra [19]. Perbedaan proses *max pooling* dan *average pooling* dapat dilihat pada gambar 2.10.



Gambar 2.10 Proses Max Pooling dan Average Pooling [56]

2.2.4.3. *ReLU*

ReLU bertujuan untuk mengaplikasikan fungsi aktivasi yang kemudian hasilnya digunakan pada *layer* selanjutnya. *Layer* ini berfungsi untuk meningkatkan sifat ketidaklinearan dari jaringan secara keseluruhan tanpa mempengaruhi bidang reseptif *convolutional layer* [17].

2.2.4.4. *Fully Connected Layer*

Fully Connected Layer memiliki tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. *Fully Connected Layer* meratakan matriks menjadi vektor dengan mentransformasi setiap neuron menjadi satu dimensi sebelum dimasukkan ke dalam *fully connected layer* [53].

2.2.4.5. *Aktivasi Softmax*

Softmax classifier digunakan untuk mengklasifikasi lebih dari 2 kelas. *Softmax* memberi kemungkinan menghitung probabilitas dari semua label, dimana nilai riil pada vektor diubah menjadi nilai antar 0 dan 1 yang jika semua dijumlahkan akan bernilai 1 [33].

2.2.4.6. *Global Average pooling Layer*

Global average pooling merupakan *average pooling* yang dapat mereduksi tensor dengan size $w \times w \times d$ ke dalam bentuk $1 \times 1 \times d$ dimana diambil nilai rata-rata pada tiap *pool* [57]. *Global average pooling (GAP) layer* digunakan setelah *convolutional layer* terakhir. *Global average pooling* melakukan perhitungan nilai rata-rata dari semua elemen pada *feature map*. GAP dapat menggantikan *fully connected layer* untuk mengurangi penyimpanan yang dibutuhkan oleh matriks berat yang besar dari *fully connected layer* [58].

2.2.5. *Transfer learning*

Transfer learning adalah metode atau teknik untuk mengklasifikasikan dataset yang baru menggunakan model yang sudah dilatih sebelumnya (*pre-trained model*) terhadap suatu dataset, sehingga tidak perlu dilakukan *training* data dari awal [59].

Transfer learning memiliki arsitektur *convolution* dan *pooling layer* yang lebih dalam daripada CNN sederhana, sehingga memungkinkan untuk mengekstraksi lebih banyak fitur dan menghasilkan informasi citra yang lebih baik [24]. Keuntungan utama menggunakan *transfer learning* adalah meningkatkan kinerja dasar, menghemat waktu pembelajaran model, dan memperoleh kinerja akhir yang lebih tinggi [60].

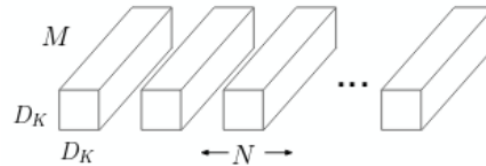
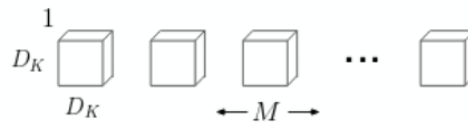
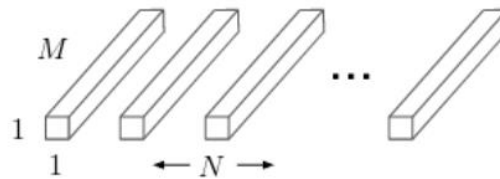
Ada 2 tahap utama yang dilakukan pada *transfer learning*, yaitu *pre-training* dan *fine-tuning*. Proses *pre-training* adalah menggunakan *base model* yang bobotnya dibekukan untuk mempercepat proses *training*, sedangkan *fine-tuning* dilakukan untuk *training* pada sebagian lapisan pada *base model* untuk memperoleh akurasi yang baik terhadap model yang dihasilkan dari *pretraining* [31].

Ada beberapa *pre-trained model* yang bersaing untuk mendapatkan score akurasi tertinggi yang dilatih terhadap ImageNet. Dataset Imagenet berisi lebih dari 14 juta gambar dari 1000 kelas [33]. Model tersebut diantaranya yaitu VGG16, VGG19, ResNet50, MobileNet, MobileNetV2, AlexNet [26][61]. Dalam penelitian ini, MobileNetV2 akan digunakan sebagai *pre-trained model* yang telah dilatih menggunakan ImageNet.

2.2.6. MobileNetV2

MobileNetV2 merupakan pengembangan dari model versi-versi sebelumnya, yaitu MobileNet [30]. Arsitektur MobileNet memiliki perbedaan mendasar dengan arsitektur CNN pada umumnya, yaitu penggunaan lapisan atau *layer* konvolusi dengan ketebalan filter yang sesuai dengan ketebalan input image, sehingga dapat menghemat ukuran model yang dibuat [34]. Teknik yang digunakan oleh MobileNet adalah konvolusi kedalaman terpisah atau *depthwise separable convolution* (DSP) yang membagi *layer* konvolusi menjadi *depthwise convolution* dan *pointwise convolution* seperti yang ditunjukkan pada gambar 2.11 [29][30]. Tujuan dari *layer* ini adalah untuk mereduksi komputasi (parameter) sehingga menghasilkan ukuran model yang lebih kecil [33]. Lapisan *depthwise convolution* merupakan lapisan pertama yang melakukan penyaringan ringan dengan

menerapkan *single convolutional filter* untuk setiap input channel. Sedangkan *pointwise convolution* atau 1×1 convolution merupakan lapisan kedua yang bertanggung jawab dalam membangun fitur baru dengan mengkomputasi kombinasi linear dari hasil *depthwise convolution* [31].

(a) *Standard Convolution Filters*(b) *Depthwise Convolution Filters*(c) *1x1 Convolutional Filters (Pointwise Convolution)*

Gambar 2.11 Konvolusi standard (a) dibagi menjadi dua lapisan: depthwise convolution (b) dan pointwise convolution (c) untuk membuat filter terpisah secara mendalam (depthwise) [62]

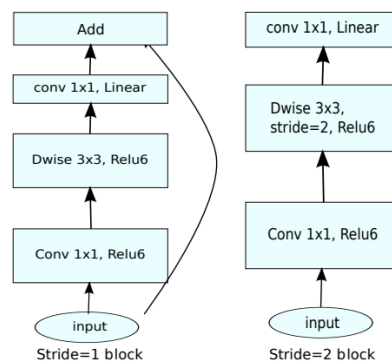
MobileNetV2 merupakan versi kedua dari MobileNet yang dirilis pada tahun 2017 [63]. MobileNetV2 juga menggunakan *Depthwise Separable Convolution* sebagai lapisan utamanya. MobileNetV2 memiliki perbedaan dengan versi sebelumnya yaitu adanya penambahan 2 fitur baru berupa *linear bottleneck* dan *shortcut connections* antar *bottleneck* [30]. Antar model pada *bottleneck* terdapat input dan output sedangkan *layer* bagian dalam mengenkapsulasi kemampuan model mengubah input dari konsep tingkat yang lebih rendah (piksel) ke descriptor tingkat yang lebih tinggi (kategori gambar). *Linear bottlenecks* berfungsi untuk mencegah kerusakan informasi, sedang *shortcut connections* memungkinkan proses

training lebih cepat dan meningkatkan akurasi [28]. Struktur *bottleneck* pada MobileNetV2 dapat dilihat pada tabel 2.2.

Tabel 2.2 Blok Penyusun MobileNetV2 [28]

Input	Operator	Output
$h \times w \times k$	1x1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise $s = s$, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Berdasarkan blok penyusun MobileNetV2, detail arsitektur yang dimiliki yaitu sebanyak 32 filter pada *fully convolutional layer*, yang diikuti dengan 19 *layer residual bottleneck*, lalu ReLU6 sebagai variabel non-linear karena memiliki keunggulan pada komputasi rendah. Kernel yang digunakan adalah standar kernel yaitu 3 x 3, selain itu batch normalization dimanfaatkan pada tahap *training* [28].



(d) Mobilenet V2

Gambar 2.12 Building Bottleneck MobileNetV2 [33]

Tabel 2.3 Arsitektur MobileNetV2 [34]

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

Input	Operator	t	c	n	s
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 3320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Transformasi fitur terjadi dari channel tk ke k dengan s adalah *stride*, dan faktor ekspansi t . Bottleneck menambahkan 1×1 convolutional layer di depan *depthwise convolutional layer* dan menggunakan *linear activation* setelah *pointwise convolution layer* untuk mencapai tujuan *downsampling* dengan mengatur parameter di *depthwise convolutional layer*. Keseluruhan dari arsitektur MobileNetV2 terlihat pada tabel 2.3 dimana *conv2d* adalah standar *convolution*, *avgpool* adalah *average pooling*, c adalah *output channel* dan n adalah perulangan [32].

MobileNetV2 adalah jaringan yang menyediakan pengembangan model *computer vision* yang dapat disesuaikan untuk perangkat *mobile*. MobileNetV2 dapat mengurangi jumlah operasi dan memori dengan tetap menjaga akurasi yang sama [31]. Selain memiliki akurasi yang cukup tinggi, MobileNetV2 juga memiliki jumlah *training* parameters yang kecil dibanding arsitektur CNN lainnya, sehingga kebutuhan komputasinya lebih ringan. MobileNetV2 hanya memiliki model size 14MB saja, namun masih memiliki performansi yang baik [33]. Berdasarkan tabel 2.2, MobileNetV2 memiliki top-1 *accuracy* sebesar 71%, top-5 *accuracy* sebesar 90%, dan parameter sejumlah 3.538.984.

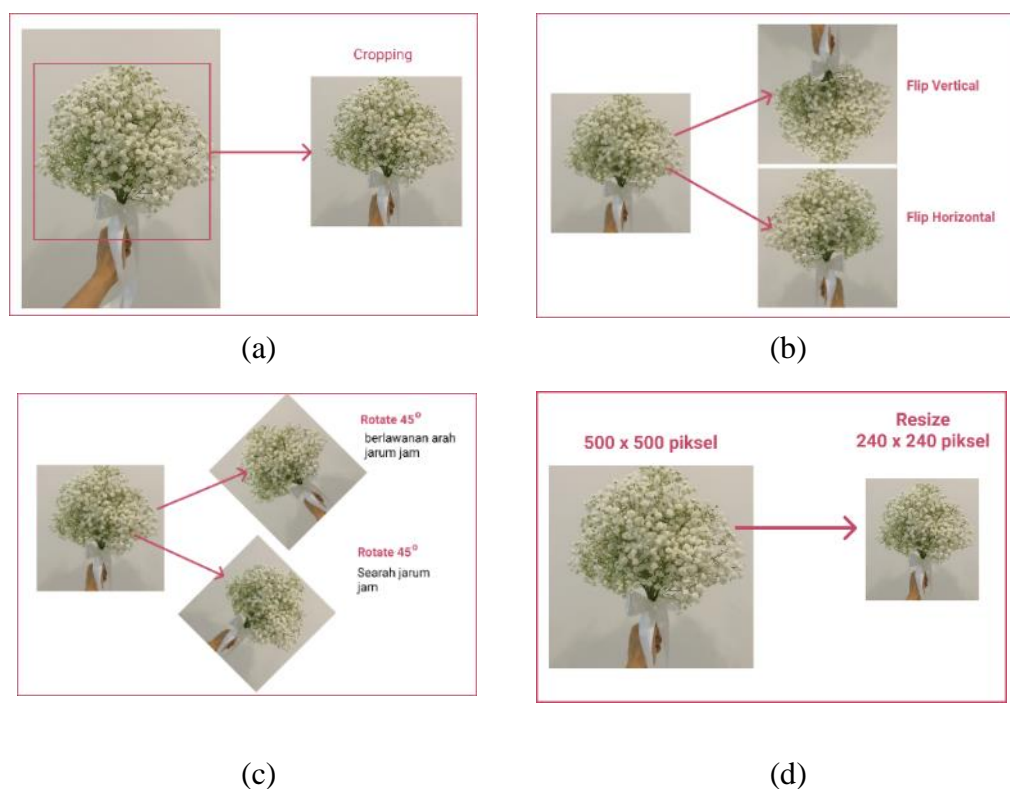
Tabel 2.4 MobileNetV2 berdasarkan website Keras [35]

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
MobileNetV2	14MB	0.713	0.901	3.538.984	88

2.2.7. Augmentasi Data

Augmentasi data memiliki tujuan menambah variasi data sehingga mencegah overfitting pada saat proses pelatihan [4]. Data diaugmentasi dengan membuat

salinan sumber data tanpa mengubah label pada setiap bagian dari data tersebut [64]. Beberapa operasi yang digunakan pada augmentasi data diantaranya meliputi *cropping*, *flipping*, *rotating*, dan *resizing*. *Cropping* dilakukan untuk memotong satu bagian dari citra. *Flipping* yaitu perubahan orientasi citra secara horizontal, vertical, atau keduanya. *Rotating* yaitu memutar citra terhadap titik pusat dalam derajat, baik searah atau berlawanan jarum jam. *Resizing* adalah mengubah resolusi citra [33]. Contoh operasi augmentasi data citra dapat dilihat pada gambar 2.13.



Gambar 2.13 Operasi augmentasi (a) *Cropping* (b) *Flip Vertical* dan *Horizontal* (c) *Rotate* (d) *Resize*

2.2.8. Accuracy dan Loss

Proses pelatihan atau *training* dilakukan untuk melatih model yang telah dirancang terhadap dataset citra pada CNN. Pada proses *training* dilakukan menggunakan dataset *train* dan data *validation*. Data *train* adalah data yang digunakan untuk mengubah *weights* atau bobot dari model tersebut untuk melatih model tersebut. Sedangkan data *validation* adalah data yang belum pernah dipelajari oleh model tersebut. Dari hasil training model yang dilakukan dapat

diperoleh *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss*. *Accuracy* adalah nilai yang digunakan untuk mengetahui tingkat keberhasilan model yang telah dibuat. Sedangkan nilai *loss* adalah suatu ukuran sebuah error yang dibuat network untuk meminimalisir kesalahan prediksi. *Training accuracy* adalah nilai dari perhitungan akurasi pada data *training* dan prediksi dari model yang telah dibuat. Sedangkan *validation accuracy* adalah nilai perhitungan akurasi dari data *validation* dan prediksi dari model yang telah dibuat dengan input data dari *validation* dataset tersebut. *Training loss* adalah nilai dari penghitungan loss function dari training dataset dan prediksi dari model. Sedangkan *validation loss* adalah nilai penghitungan loss function dari data *validation* dan prediksi dari model yang telah dilatih dengan input data dari data *validation* [65].

2.2.9. *Overfitting & Hyperparameter*

Underfitting dan *overfitting* model adalah ketika model tidak dapat melakukan prediksi dengan benar. *Underfitting* terjadi ketika model tidak dapat memprediksi data *training* maupun data *testing* dengan benar. Sedangkan *overfitting* model terjadi ketika model yang dibuat lebih fokus mempelajari data *training*, sehingga tidak melakukan prediksi dengan tepat pada data *testing*. *Underfitting* model memiliki *loss* yang tinggi dan akurasi rendah, sedangkan *overfitting* memiliki *loss* rendah dan akurasi rendah. Model yang baik adalah ketika memiliki *loss* yang rendah dan akurasi tinggi. *Overfitting* dapat diatasi dengan penyesuaian dengan *hyperparameter* [33].

Hyperparameter adalah sebuah parameter yang digunakan untuk mengontrol saat proses *training* untuk mendapatkan hasil *score* akurasi yang baik. Beberapa *hyperparameter* tersebut diantaranya yaitu :

2.2.9.1. *Input shape*

Ukuran citra yang disarankan pada beberapa arsitektur model CNN adalah 224x224x3 piksel dimana 3 merupakan *channel* RGB [33].

2.2.9.2. *Batch size*

Batch size digunakan untuk memecah dataset ke dalam beberapa bagian kecil. Beberapa nilai yang umumnya digunakan sebagai *batch size* yaitu 16, 32, 64, 128, atau 256 [33].

2.2.9.3. *Epoch*

Epoch adalah putaran penuh pada saat *training* terhadap seluruh dataset. Tidak ada keterangan yang pasti terkait berapa jumlah *epoch* yang optimal, karena akan berbeda untuk kumpulan data yang berbeda [33].

2.2.9.4. *Optimizer*

Optimizer adalah metode yang bertujuan untuk menurunkan nilai *loss* sehingga menghasilkan akurasi skor terbaik dengan meng-*update* bobot (*weight*). *Adam Optimizer* adalah salah satu *optimizer* terbaik karena cepat dalam mencapai *loss* minimum [33].

2.2.9.5. *Learning Rate*

Learning rate adalah parameter yang mengontrol seberapa cepat atau lambat model mempelajari masalah ketika proses *training*. Semakin kecil *learning rate* yang digunakan, maka waktu yang diperlukan semakin lama untuk mencapai bobot yang optimal. Jika terlalu besar nilai *learning rate* yang digunakan, maka bobot yang optimal dapat terlewatkan. Beberapa *learning rate* yang umum digunakan diantaranya yaitu 0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3 [33].

2.2.10. *Confusion matrix* dan *Classification Report*

Evaluasi performa menjadi langkah penting dalam *life cycle* model *deep learning* untuk mengukur kinerja model. Untuk mengevaluasi model klasifikasi teknik yang dapat digunakan yaitu *Confusion matrix* dan *Classification Report* [33].

2.2.10.1. *Confusion matrix (CM)*

Confusion matrix adalah tabel NxN dimana N merupakan jumlah kelas / label / kategori) yang berisi jumlah prediksi yang benar dan salah dari model klasifikasi. Tujuan dari *Confusion matrix* yaitu membandingkan nilai aktual dengan nilai prediksi. Keluaran nilai prediksi dari program adalah bernilai Positif atau Negatif, sedangkan nilai sebenarnya atau nilai aktual adalah berupa nilai *True* dan *False* [26]. *Confusion matrix* mengembalikan nilai dengan 4 kategori, yaitu *True Positive* (TP), *True Negatif* (TN), *False Positive* (FP), dan *False Negatif* (FN). TP adalah prediksi positif dan nilai sebenarnya positif. TN adalah prediksi negatif dan nilai sebenarnya negatif. TP dan TN merupakan jumlah klasifikasi yang bernilai benar. FP adalah prediksi positif dan nilai sebenarnya negatif. FP disimpulkan ketika prediksi yang dihasilkan tidak tepat atau bernilai *yes* (positif) saat prediksi yang diharapkan adalah *no*(negatif). FN adalah prediksi negatif dan nilai sebenarnya positif [66].

2.2.10.2. *Classification Report*

Metrics yang digunakan untuk mengukur kinerja model dihitung menggunakan data yang didapatkan dari *Confusion matrix*. *Metrics* yang dihasilkan adalah *Accuracy*, *Precision*, *Recall*, dan *F1-score*. *Accuracy* adalah metrics yang menggambarkan seberapa akurat model mengklasifikasikan dengan benar. Sedangkan *precision* menggambarkan akurasi antara kasus yang diprediksi positif dan juga hasilnya *True Positif* (TP) pada data sebenarnya. *Recall* adalah keberhasilan model menemukan kembali informasi atau proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar. *F1-score* adalah perbandingan rata-rata *precision* dan *recall* yang dibobotkan [33]. Perhitungan *confusion matrix* ditunjukkan pada tabel 2.5.

Tabel 2.5 *Confusion Matrix*

Klasifikasi		Prediksi	
		Positif	Negatif
Aktual	Positif	<i>True Positive</i> (TP)	<i>True Negative</i> (TN)
	Negatif	<i>False Positive</i> (FP)	<i>False Negative</i> (FN)

Dari tabel 2.5 dapat dihitung dapat diperoleh hasil *accuracy*, *precision*, *recall* dan *f1-score* menggunakan persamaan (2,1), (2,2), (2,3), dan (2,4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2,1)$$

$$Precision = \frac{TP}{FP+TP} \quad (2,2)$$

$$Recall = \frac{TP}{TP+FN} \quad (2,3)$$

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (2,4)$$

Keterangan :

True Positive (TP) = prediksi positif dan nilai sebenarnya positif

True Negative (TN) = prediksi negatif dan nilai sebenarnya negatif

False Positive (FP) = prediksi positif dan nilai sebenarnya negatif

False Negative (FN) = prediksi negatif dan nilai sebenarnya positif

2.2.11. *Tensorflow*

Tensorflow merupakan platform *open source end-to-end* library machine learning yang dibuat oleh tim Google Brain [67]. *Tensorflow* adalah sebuah *library* yang bersumber terbuka untuk komputasi numerik dan pembelajaran mesin berskala besar. *Tensorflow* dapat melatih dan menjalankan jaringan saraf yang mendalam untuk klasifikasi digit tulisan tangan, pengenalan gambar, kata yang disematkan, dan lain-lain. *Tensorflow* mendukung berbagai aplikasi, prediksi

produksi pada skala besar, yang berfokus pada pelatihan dan inferensi pada *deep neural network* [16].

2.2.12. Python

Python merupakan bahasa pemrograman tingkat tinggi yang banyak digunakan oleh akademisi maupun umum [68]. Python digunakan dengan berbagai tujuan, mulai dari membangun aplikasi berbasis desktop atau website, membuat permainan (games), hingga untuk keperluan statistik, Internet of Things (IoT), analisis, mengolah dan visualisasi data, serta saat ini umum digunakan untuk membangun proyek AI [33].

2.2.13. Google Colaboratory

Google Colab atau *Google Colaboratory* merupakan *executable document* yang cocok digunakan untuk menulis dan mengeksekusi kode *machine learning*, data analisis, atau pendidikan [69]. *Google Colab* memiliki keunggulan yaitu tidak memerlukan konfigurasi, layanan gratis *General Processing Units* (GPU) yang merupakan prosesor untuk mengolah tampilan grafik, dan dapat berbagi dengan mudah ke sesama rekan penelitian. Aplikasi *deep learning* dapat dibangun menggunakan Colab dengan library seperti Keras, Tensorflow, Pytorch, dan OpenCV [70].