

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

##### 2.1.1. Data augmentation for Automated Pest classification in Mango Farms [4]

Penelitian ini membahas tentang penggunaan teknologi citra digital dalam klasifikasi hama pohon mangga untuk membantu petani mangga. Pembahasan meliputi model yang dapat digunakan untuk klasifikasi hama, *dataset* yang belum umum tersedia sehingga diperlukan pembuatan *dataset* beserta teknik augmentasi data untuk memperbanyak *dataset*, dan pembuatan aplikasi dengan *Android* untuk penggunaan sistem klasifikasi hama. Model atau metode yang digunakan untuk klasifikasi menggunakan VGG-16 dengan *Fully Convolutional Network (FCC)* dengan tingkat akurasi terbaik mencapai 76%. Untuk *dataset* sendiri dikumpulkan secara manual selama kurun waktu 6 bulan pada perkebunan mangga di Indonesia, dan mendapatkan gambar yang berjumlah 512 gambar serta meliputi 15 jenis hama yang umum ditemui pada pohon mangga beserta gambar daun mangga yang tidak terinfeksi hama. *Dataset* tersebut masih belum mencukupi untuk mendapatkan hasil yang memuaskan sehingga dilakukan *data augmentation* dengan mengubah gambar-gambar asli tersebut menjadi banyak gambar yang telah dimodifikasi dengan perubahan pada *contrast*, *affine*, *blur*, *noise*, dan lain-lain. Hasil akhir *dataset* yang telah dilatih dengan VGG-16 dapat digunakan melalui aplikasi *Android* yang dikembangkan. Salah satu kontribusi utama penelitian ini adalah *dataset* yang digunakan dalam penelitian yang akan dilakukan penulis.

### **2.1.2. Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection [6]**

Penelitian ini membahas tentang klasifikasi penyakit pada daun mangga dengan solusi *Artificial Neural Network (ANN)* yang memanfaatkan *Feature Selection* dari citra yang didapatkan. Solusi *ANN* diberikan sebagai alternatif dengan lapisan yang lebih sederhana dibanding metode *CNN* umum yang digunakan seperti AlexNet, VGG, dan ResNet. Metode *ANN* yang dikembangkan memanfaatkan citra dengan resolusi tinggi untuk dapat mendeteksi *blob* infeksi hama, dan pada penelitian dilakukan pemrosesan citra dengan meningkatkan kontras sehingga citra semakin tajam. Hasil yang didapatkan dari metode *ANN* lebih baik dibanding dengan *CNN* lain yang digunakan sebagai pembandingan, dengan terdapat pula peningkatan pada akurasi *CNN* yang digunakan jika memanfaatkan *transfer learning* dengan *pre-trained weights* diambil dari dataset public *Plant Village*, untuk hasil akurasi penelitian sebagai berikut ANN 89.41%, AlexNet 78.64%, VGG 79.92%, dan ResNet 84.88%. Untuk dataset yang digunakan penulis dari penelitian [4], merupakan citra yang diambil dengan resolusi umum kamera pada telepon pintar, sehingga dirasa lebih cocok untuk tetap menggunakan metode *CNN* yang umum dengan *transfer learning*. Dari penelitian ini penulis mengambil hasil penggunaan model *ResNet50* cocok untuk penggunaan pada *dataset* yang digunakan penulis, karena hasil *ResNet50* diatas semua model *CNN* umum diluar model baru penelitian ini.

### **2.1.3. Data Augmentation using Adversarial Networks for Tea Diseases Detection [8]**

Penelitian ini membahas tentang *data augmentation* menggunakan metode *adversarial networks* seperti *General Adversarial Networks (GAN)* dan *Deep Convolutional General Adversarial Networks (DCGAN)* untuk mengembangkan *dataset* asli tentang penyakit yang menginfeksi

tanaman teh. Hasil pengembangan *dataset* dicoba dengan berbagai metode *CNN* umum seperti AlexNet, DenseNet, ResNet, dan Xception dimana hasil terbaik didapatkan oleh metode DenseNet, dengan rata-rata terdapat peningkatan akurasi pada semua metode sebesar 1%-2% saat dicoba menggunakan *dataset* yang telah dikembangkan menggunakan *GAN* dan *DCGAN*. Penelitian ini juga menyimpulkan tidak terdapat perbedaan akurasi yang signifikan antara augmentasi data menggunakan *GAN* dengan *DCGAN*, melainkan perbedaan terdapat pada performa dimana *DCGAN* menunjukkan performa yang lebih baik dibandingkan *GAN*. Dari penelitian ini, penulis mendapatkan informasi tentang penggunaan *Generative Adversarial Network* untuk metode augmentasi, begitu pula dengan variasi dari *GAN* itu sendiri.

#### **2.1.4. Application of Transfer Learning to Detect Potato Disease from Leaf Image [9]**

Penelitian ini membahas tentang penggunaan *transfer learning* yang dapat meningkatkan akurasi pendeteksian penyakit tanaman kentang melalui citra daun. Penelitian ini dilakukan dengan mengubah citra *RGB* menjadi citra *grayscale* serta membuat *CNN* tradisional sebanyak tujuh lapisan dan dibandingkan dengan *CNN* yang memanfaatkan *transfer learning* diantaranya VGG, AlexNet, MobileNet, dan ResNet. Hasil yang didapatkan menunjukkan bahwa dengan memanfaatkan *transfer learning* terdapat peningkatan akurasi yang signifikan dibanding *CNN* tradisional. Penelitian ini memanfaatkan *dataset Plant Village* sebagai *weights* untuk *transfer learning*, dan performa keseluruhan metode sangat dipengaruhi oleh kapabilitas perangkat keras mesin dimana metode dijalankan. Dari penelitian ini penulis mengambil informasi tentang penerapan *transfer learning* untuk pelatihan model klasifikasi penyakit pada daun tanaman.

### 2.1.5. Transfer Learning Based Plant Diseases Detection Using ResNet50 [10]

Penelitian ini membahas tentang metode-metode *CNN* yang umum digunakan dan berfokus pada penggunaan ResNet dengan *transfer learning* untuk pendeteksian penyakit pada tanaman melalui citra daun. *Dataset* yang digunakan didapatkan dari GitHub dan terdiri dari 87,867 gambar daun yang tercampur antara daun sehat dengan daun yang terinfeksi. Terdapat langkah *pre-processing* yang dilakukan yakni melakukan standarisasi ukuran citra menjadi 224\*224 piksel yang merupakan ukuran citra umum untuk dimasukkan ke dalam ResNet. Dilakukan *data augmentation* pada *dataset* yang digunakan untuk memperluas ketersediaan data menggunakan metode tradisional seperti citra diputar, tinggi dan lebar citra diubah, dan lain-lain. *Training* dilakukan dengan metode ResNet50 dengan pembagian data sebesar 80% untuk *training* dan 20% untuk validasi hasil, dilakukan pula *fine-tuning* pada parameter yang digunakan untuk meningkatkan akurasi metode. Hasil akhir yang didapatkan menunjukkan metode ResNet50 mengungguli metode-metode lain seperti AlexNet dan VGG pada akurasi, serta performa ResNet50 yang lebih baik yang ditandai dengan lebih sedikitnya *epoch* untuk mencapai *convergence*. Penulis mengambil informasi tentang *benchmark* hasil ResNet50 dengan *transfer learning* yang mengungguli performa model lain.

### 2.1.6. Plant Disease Recognition on images using convolutional neural networks: A systematic review [11]

Penelitian ini berisikan kumpulan penelitian serupa yang membahas tentang pendeteksian penyakit tanaman melalui citra digital memanfaatkan *CNN*. Isi penelitian ini merupakan pembahasan 121 jurnal dengan topik serupa selama kurun waktu 10 tahun terakhir (2011 – 2021) dan isinya meliputi *framework* yang umum digunakan oleh penelitian

serupa, metode algoritma yang umum digunakan, serta pendekatan cara yang dapat dilakukan seperti *data augmentation* dan *transfer learning*. Secara keseluruhan penelitian yang dilakukan dapat mempermudah penelitian dibidang yang serupa dikemudian hari, dengan memberikan *summary review* yang menyeluruh dan berisikan data-data faktual yang dapat menjadi pertimbangan untuk pendekatan suatu penelitian pada bidang pendeteksian penyakit tanaman melalui citra digital, termasuk klasifikasi penyakit seperti yang ingin dilakukan oleh penulis. Penulis mengambil informasi tentang model yang umum digunakan, manfaat penggunaan *transfer learning*, dan metode augmentasi yang umum diterapkan pada pengolahan citra penyakit tanaman.

#### 2.1.7. Summary Tinjauan Pustaka

Dari tinjauan pustaka yang penulis gunakan, terdapat beberapa implikasi yang menjadi dasar dari penelitian yang penulis lakukan, diantaranya adalah:

- Penulis mendapatkan *dataset* dan hasil model VGG-16 dari penelitian Dr. Kusri [4], dimana hasil VGG-16 menjadi *baseline* akurasi untuk ditingkatkan.
- Penulis mendapatkan *benchmark* model *deep learning* dari penelitian [6], dan performa penggunaan *ResNet50* pada permasalahan klasifikasi penyakit daun mangga.
- Penulis mendapatkan informasi tentang aplikasi *GAN* sebagai augmentasi citra penyakit daun dari penelitian [8], dan penulis menggunakan varian *leafGAN* yang dijelaskan pada tinjauan teori.
- Penulis mendapatkan informasi penggunaan *transfer learning* dari *weights Plant Village* untuk peningkatan performa model *deep learning* dari penelitian [9].

- Penulis mendapatkan informasi *transfer learning* pada *ResNet50* secara spesifik, termasuk ketentuan *input* dan pengaturan parameter model dari penelitian [10].
- Penulis mendapatkan informasi *journal review* berkaitan aplikasi *deep learning* pada penyakit daun dari penelitian [11].

Dari kesimpulan diatas, penulis mendapatkan informasi-informasi pendukung seperti *dataset* untuk penelitian, *ResNet50* dengan *transfer learning* untuk peningkatan performa model, metode augmentasi *dataset*, dan *journal review* tentang masalah yang diteliti penulis.

Selain jurnal utama Dr. Kusriani [4] yang dijadikan referensi, jurnal lain dengan kesamaan pokok penelitian adalah jurnal [6] yang membahas pendeteksian penyakit daun mangga menggunakan *deep learning*. Perbedaan penelitian yang dilakukan oleh penulis dari jurnal [6] tersebut adalah:

- Penggunaan dataset hama mangga yang berbeda, dimana dataset yang digunakan penulis didapatkan dari Dr. Kusriani [4] dan merupakan hama mangga di Indonesia.
- Perbedaan kedua terdapat pada *pre-trained weight* untuk model *transfer learning* yang berbeda, dimana penulis berasal dari *ImageNet* dan jurnal [6] berasal dari *Plant Village*.
- Perbedaan ketiga terdapat pada penggunaan augmentasi data, penulis menggunakan augmentasi konvensional dan augmentasi *leafGAN* sedangkan jurnal [6] tidak melakukan augmentasi pada penelitiannya.
- Dan perbedaan terakhir adalah penulis tidak membuat model baru melainkan menggunakan model yang sudah ada, sedangkan jurnal [6] membuat model *ANN* baru untuk datasetnya.

## 2.2 Tinjauan Teori

### 2.2.1. Computer Vision

*Computer Vision (CV)* adalah bagian dari *Artificial Intelligence (AI)* yang memungkinkan komputer untuk mengekstrak informasi dari berbagai media visual selayaknya manusia memahami suatu hal melalui penglihatan. *Computer Vision* dimanfaatkan untuk membantu melakukan tugas manusia yang berhubungan dengan penglihatan, namun secara teratur dan terus menerus sehingga mampu melewati kapabilitas manusia bila digunakan dengan benar. Perkembangan *Computer Vision* meliputi *deep learning* dan *CNN*, serta *image classification*, *object detection*, *image segmentation*, dan lain-lain. [12]

### 2.2.2. Hama Tumbuhan

Hama tumbuhan adalah setiap spesies, *strain* atau biotipe tanaman, hewan atau patogen yang merugikan tanaman ataupun produk hasil tanaman [13]. Pada pohon mangga di Indonesia, terdapat beberapa hama yang sering muncul dan merugikan hasil panen para petani mangga karena hama ini merusak sel-sel tanaman mangga ataupun buah mangga yang sehat, beberapa diantara jenis hama tersebut adalah lalat buah, penggerek pucuk, penggerek batang, kutu putih, dan wereng mangga [2]. Dari dataset yang disediakan oleh Dr. Kusri [4] dan digunakan oleh penulis, terdapat 15 kelas hama pada pohon mangga, dengan keterangan sebagai berikut:

1. *Apoderus javanicus*, atau disebut juga *Strigapoderus javanicus* (*leaf-rolling weevil*) adalah serangga yang memotong daun mangga untuk dijadikan sebagai sarangnya [14]. Dampak hama ini terlihat seperti pada gambar 2.1 dibawah.



Gambar 2. 1 Dampak Apoderus Javanicus

2. *Aulacaspis tubercularis* (*white mango scale*), adalah hama serangga berwarna putih yang menempel pada daun mangga dan menyebabkan daun perlahan layu sampai mati [15]. Dampak hama ini terlihat seperti pada gambar 2.2 dibawah.



Gambar 2. 2 Dampak Aulacaspis Tubercularis

3. *Ceroplastes rubens* (*pink wax scale*), adalah serangga menghisap getah tanaman dan dapat menyebabkan tumbuhnya jamur sehingga mengurangi area fotosintesis daun serta kerusakan visual daun [16]. Dampak hama ini terlihat seperti pada gambar 2.3 dibawah.





Gambar 2. 3 Dampak Ceroplastes Rubens

4. *Cisaberoptus kenyae* (leaf coating mite), adalah kutu daun yang membuat lapisan putih seperti lilin pada daun, kutu menghisap getah daun menyebabkan perubahan warna serta menyebabkan daun gugur dan mati [17]. Dampak hama ini terlihat seperti pada gambar 2.4 dibawah.



Gambar 2. 4 Dampak Cisaberoptus Kenyae

5. *Dappula tertia* (bagworm moth), adalah ngengat yang menaruh telur di daun, larva yang menetas memakan daun untuk dapat melakukan transformasi, menyebabkan kerusakan pada daun yang dimakannya [18]. Dampak hama ini terlihat seperti pada gambar 2.5 dibawah.



Gambar 2. 5 Dampak Dappula Tertia

6. *Dialeuropora decempuncta* (breadfruit whitefly), adalah serangga yang menempel di daun dan menghisap getah pada daun, dapat berujung pada gugurnya daun yang disinggahi maupun ikut menyebarkan penyakit tanaman [19]. Dampak hama ini terlihat seperti pada gambar 2.6 dibawah.



Gambar 2. 6 Dampak Dialeuropora Decempuncta

7. *Erosomyia sp* (mango gall midge), adalah serangga menyerupai lalat yang menempel di daun, dimana larvanya memakan daun dari dalam, menyebabkan benjolan yang dapat terlihat, daun yang terinfeksi dapat gugur lebih cepat [20]. Dampak hama ini terlihat seperti pada gambar 2.7 dibawah.



Gambar 2. 7 Dampak Erosomyia Sp

8. *Icerya seychellarum* (Seychelles scale), adalah serangga sisir berwarna putih yang menempel pada daun, menghisap getah tanaman dan memungkinkan pertumbuhan jamur pada area yang terdampak, menyebabkan keguguran daun tanaman [21]. Dampak hama ini terlihat seperti pada gambar 2.8 dibawah.



Gambar 2. 8 Dampak Icerya Seychellarum

9. *Ischnaspis longirostris* (black thread scale), adalah serangga sisir dengan bentuk tubuh lonjong berwarna hitam sekilas mirip lintah, menghisap getah daun dan menyebabkan klorosis atau kegagalan pembentukan klorofil pada daun [22]. Dampak hama ini terlihat seperti pada gambar 2.9 dibawah.



Gambar 2. 9 Dampak Ischnaspis Longirostris

10. *Mictis longicornis* (leaf footed bug), adalah serangga berukuran sedang yang menghisap nutrisi tanaman baik itu daun, batang, maupun buah, dampak dari daun yang dihisap adalah klorosis, dan menyebabkan daun mati [23]. Dampak hama ini terlihat seperti pada gambar 2.10 dibawah.



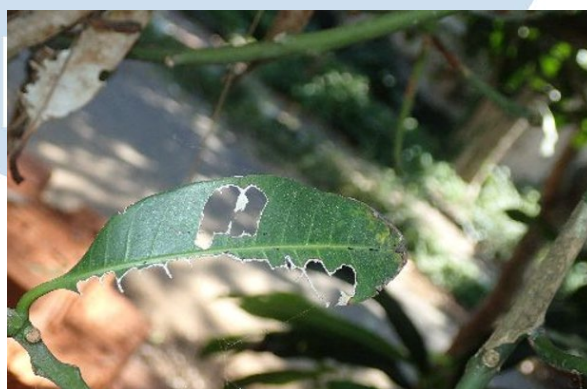
Gambar 2. 10 Dampak Mictis Longicornis

11. *Neomelicharia sparsa* (planthopper), atau umum dikenal di Indonesia sebagai wereng dan banyak ditemui pada tanaman padi adalah serangga yang menghisap getah tanaman, dan menyebabkan kerusakan pada jaringan tanaman [24]. Dampak hama ini terlihat seperti pada gambar 2.11 dibawah.



Gambar 2. 11 Dampak *Neomelicharia Sparsa*

12. *Orthaga euadrusalis* (*mango leaf webber*), adalah spesies ngengat mangga yang menaruh telur di daun mangga, dimana larva yang menetas akan memakan daun untuk nutrisi, dan dalam prosesnya merusak keseluruhan daun [25]. Dampak hama ini terlihat seperti pada gambar 2.12 dibawah.



Gambar 2. 12 Dampak *Orthaga Euadrusalis*

13. *Procontarinia matteiana* (*mango gall fly*), atau umum disebut sebagai lalat mangga adalah hama yang menyerang daun dan buah, larva yang menyerang daun dapat merusak kemampuan fotosintesis daun dan bekas lubang yang ditinggalkan dapat memungkinkan tumbuhnya jamur [26]. Dampak hama ini terlihat seperti pada gambar 2.13 dibawah.



Gambar 2. 13 Dampak Procontarinia Matteiana

14. *Procontarinia rubus* (*mango midge*), adalah spesies serupa dari lalat mangga, larva menghisap getah dan nutrisi tumbuhan, merusak jaringan dan menyebabkan klorosis [26]. Dampak hama ini terlihat seperti pada gambar 2.14 dibawah.



Gambar 2. 14 Dampak Procontarinia Rubus

15. *Valanga nigricornis* (*Javanese grasshopper*), atau disebut sebagai belalang kayu adalah serangga yang memakan tepi daun muda, menyebabkan kerusakan jaringan, penghambat pertumbuhan, hingga kematian tanaman [27]. Dampak hama ini terlihat seperti pada gambar 2.15 dibawah.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2. 15 Dampak *Valanga nigricornis*

### 2.2.3. Image Classification

*Image Classification* adalah tugas mengidentifikasi apa yang dipresentasikan oleh suatu citra digital. Digunakan suatu model tertentu yang telah dilatih untuk dapat melakukan tugas identifikasi kelas-kelas yang terdapat dari suatu gambar. Pelatihan yang dilakukan untuk model mengikuti karakteristik pembacaan kelas tertentu sesuai dengan kebutuhan pengembang model. [28]

### 2.2.4. Deep Learning

*Deep Learning (DL)* adalah perkembangan dari *Machine Learning (ML)* dimana digunakan 3 atau lebih *layer* yang berfungsi untuk mempelajari data sehingga mampu memberikan hasil prediksi yang akurat. Perbedaan antara *DL* dengan *ML* terdapat pada kemampuan *DL* untuk mempelajari karakteristik suatu data tanpa perlu supervisi manual manusia, terdapat banyak *layer* diantara dua *layer* utama yakni *input* dan *output* yang disebut dengan *hidden layer*, *hidden layer* ini berfungsi untuk mempelajari data dan memberikan prediksi yang akurat nantinya. [29]

### 2.2.5. Convolutional Neural Network

*Convolutional Neural Network (CNN)* adalah inti dari algoritma *deep learning* yang berisikan *input layer*, satu ataupun lebih *hidden layer*, dan *output layer*. Tiga *hidden layer* utama yang terdapat pada *CNN* adalah *convolutional layer*, *pooling layer*, dan *fully connected layer*. Dengan setiap *layer* yang ada, *CNN* menjadi semakin kompleks sehingga mampu mengidentifikasi berbagai bagian gambar semakin spesifik. *Layer* awal umumnya berfokus pada fitur sederhana, seperti warna dan tepi gambar. Namun saat data gambar berkembang melalui setiap *layer CNN*, maka akan mampu mengenali elemen objek lebih banyak sehingga akhirnya dapat mengidentifikasi objek dari gambar yang dimaksud. [30]

### 2.2.6. ResNet50

*Residual Network* atau *ResNet* adalah arsitektur *CNN* yang diciptakan pada tahun 2015 oleh *Microsoft Research* untuk mengatasi permasalahan *vanishing/exploding gradient* sebagai akibat dari menambahkan *hidden layer* yang terlalu banyak pada *CNN*. Teknik yang ditambahkan pada *ResNet* adalah *skip connection* dimana jika ada *layer* yang merusak kinerja arsitektur maka akan dilewati dengan *regularization*. [31]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2.x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 <sup>9</sup>	3.6×10 <sup>9</sup>	3.8×10 <sup>9</sup>	7.6×10 <sup>9</sup>	11.3×10 <sup>9</sup>

Gambar 2. 16 Arsitektur ResNet50 [32]



Dari gambar 2.16 diatas, terlihat lapisan yang terdapat dalam *ResNet50* adalah sebagai berikut:

1. Konvolusi awal memiliki kernel  $7 * 7$  dan 64 kernels lain yang berbeda dengan ukuran 2 menjadi *layer* awal.
2. Max pooling dengan ukuran *stride* 2.
3. Konvolusi selanjutnya memiliki kernel  $1 * 1,64$  dengan berikut kernel  $3 * 3,64$  dan terakhir kernel  $1 * 1,256$ . *Layer* diulang sebanyak 3 kali sehingga menjadi total 9 *layers*.
4. Selanjutnya terdapat kernel  $1 * 1,128$  dengan kernel  $3 * 3,128$  dan diakhir berupa kernel  $1 * 1,512$ . Diulang 4 kali sehingga memberi total 12 *layers*.
5. Setelah itu terdapat kernel  $1 * 1,256$  dan dua kernel dengan nilai  $3 * 3,256$  dan  $1 * 1,1024$ . Diulang sebanyak 6 kali menjadi total 18 *layers*.
6. Dan konvolusi terakhir dengan kernel  $1 * 1,512$  dan  $3 * 3,512$  serta  $1 * 1,2048$ . Diulangi 3 kali menghasilkan total 9 *layers*.
7. Dilakukan *average pooling* dan *fully connected layer* dengan 1000 *nodes* dan fungsi *softmax* sebagai *layer output*.

Dari perhitungan *layer* konvolusi dan *layer* lain diatas tanpa memasukkan *pooling* dan *activation*, didapatkan total sebanyak 50 *layers*, sehingga arsitektur ini disebut sebagai *ResNet50* [32].

### 2.2.7. Transfer Learning

*Transfer Learning* adalah penggunaan kembali model yang telah dilatih sebelumnya pada masalah baru. Dalam *machine learning*, pengetahuan yang dipelajari dari tugas sebelumnya dapat digunakan untuk meningkatkan prediksi tentang tugas baru. Keuntungan dari *transfer learning* antara lain adalah pengurangan waktu untuk pelatihan, peningkatan kinerja *CNN* pada umumnya, dan jika tidak tersedia data dalam jumlah yang besar. [33]

### 2.2.8. Data Augmentation

*Data Augmentation* adalah teknik yang dapat digunakan untuk memperluas ukuran *dataset* pelatihan secara artifisial dengan membuat data baru yang dimodifikasi dari data asli. Perluasan *dataset* ini bertujuan untuk mengatasi permasalahan *overfitting* dimana model mempelajari data secara akurat alih-alih pola dari data tersebut sehingga menyebabkan model tidak mampu mendeteksi objek yang sama pada gambar berbeda, selain itu juga bertujuan untuk mengembangkan *dataset* yang terlalu kecil dan dapat menambahkan tingkat performa dari model yang dilatih. Beberapa teknik *data augmentation* yang umum digunakan meliputi transformasi geometris, transformasi warna, pencampuran gambar, dan lain-lain. [34]

### 2.2.9. Generative Adversarial Network

*Generative Adversarial Network* adalah metode pengembangan data dengan memanfaatkan *CNN* yang berbasis pada *game theory* dimana terdapat *sub-model generator* yang berfungsi membuat gambar palsu dari gambar asli, dan *discriminator* yang berfungsi membedakan gambar asli dengan gambar hasil *generator*. Tujuan akhirnya adalah menciptakan sebanyak mungkin data yang menyerupai asli oleh *generator*, namun masih tetap dapat dibedakan oleh *discriminator*. Manfaat dari penggunaan *GAN* adalah mampu mengembangkan *dataset* dengan berisikan gambar yang menyerupai gambar asli namun tetap mampu untuk dibedakan, sehingga *dataset* akan berisikan semakin banyak data yang valid untuk pelatihan model. [35]

## 2.2.10. Metode Evaluasi

Penulis menggunakan *Confusion Matrix* sebagai metode evaluasi yang digunakan untuk melihat hasil validasi model. Pada penelitian ini, jenis yang digunakan adalah *Confusion Matrix Multi-Class* dengan matrix 16\*16 sesuai dengan jumlah kelas pada *dataset* yang digunakan [36] seperti pada ilustrasi gambar 2.17.

Dari *confusion matrix*, penulis dapat memberikan persentase tingkat *recall* dan presisi dari model, lalu dari kedua nilai tersebut dapat dihitung kembali metrik evaluasi lanjutan yakni *F1-score* dari model tersebut secara keseluruhan.

*Confusion matrix* adalah pemetaan hasil prediksi model berbanding dengan nilai asli model, dari *confusion matrix* ini didapatkan beberapa variabel seperti *true positive*, *true negative*, *false positive*, dan *false negative* [37]. Keempat variabel inilah yang digunakan untuk menghitung presisi dan *recall* dari model.

		True Class		
		Apple	Orange	Mango
Predicted Class	Apple	7	8	9
	Orange	1	2	3
	Mango	3	2	1

Gambar 2. 17 Ilustrasi Multi-Class Confusion Matrix [37]

*Precision* adalah tingkat keberhasilan suatu model dalam memprediksi objek sesuai dengan label atau kelas asli dari objek tersebut.

Seperti pada gambar dibawah, perhitungan untuk kelas apel dapat dilakukan dengan menghitung jumlah kelas apel asli yang berhasil diprediksi lalu dibagi dengan semua prediksi yang menghasilkan prediksi objek sebagai apel. Rumus *precision* adalah *true positive* dibagi dengan total *true positive* dan *false positive*, ilustrasi perhitungan *precision* dapat dilihat dari gambar 2.18 dibawah ini.

		True Class		
		Apple	Orange	Mango
Predicted Class	Apple	7	8	9
	Orange	1	2	3
	Mango	3	2	1

Gambar 2. 18 Ilustrasi Perhitungan Precision [37]

*Recall* adalah tingkat keberhasilan suatu kelas muncul dalam hasil prediksi sekalipun kelas tersebut diprediksi sebagai kelas lain. Pada gambar dibawah dapat dilihat perhitungan *recall* pada kelas apel dengan menjumlahkan hasil prediksi asli kelas apel dibagi semua kelas apel yang terprediksi sekalipun termasuk didalam kelas lain. Rumus *recall* adalah *true positive* dibagi dengan jumlah gabungan total *true positive* dan *false negative*, ilustrasi perhitungan *recall* pada *multi-class confusion matrix* dapat dilihat pada gambar 2.19 dibawah ini.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

		True Class		
		Apple	Orange	Mango
Predicted Class	Apple	7	8	9
	Orange	1	2	3
	Mango	3	2	1

Gambar 2. 19 Ilustrasi Perhitungan Recall [37]

*F1-score* adalah rerata harmonik dari *precision* dan *recall* dengan memanfaatkan rumus seperti yang terlihat pada gambar 2.20 berikut:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}fp + fn}$$

Gambar 2. 20 Rumus Perhitungan F1-score [37]

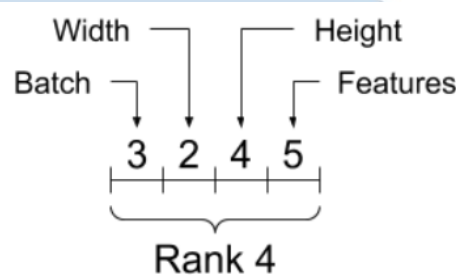
*F1-score* merupakan metrik umum yang digunakan sebagai alternatif dari akurasi untuk menilai performa suatu model.

### 2.2.11. TensorFlow Tensor versus NumPy Array

*TensorFlow* adalah *framework* pada bahasa pemrograman *python* yang dikembangkan oleh peneliti *Google* dengan tujuan mempermudah pembuatan model *machine learning*, *deep learning*, dan lain-lain. *TensorFlow* menyediakan berbagai modul pendukung untuk mempermudah pembuatan hingga pelatihan model *machine learning*, salah satunya seperti dengan penggunaan *high-level API* seperti *TensorFlow Keras*.

*TensorFlow* juga menyediakan pembentukan *dataset* ke dalam bentuk *tensor*, dan salah satu bentuk *tensor* yang umum digunakan untuk

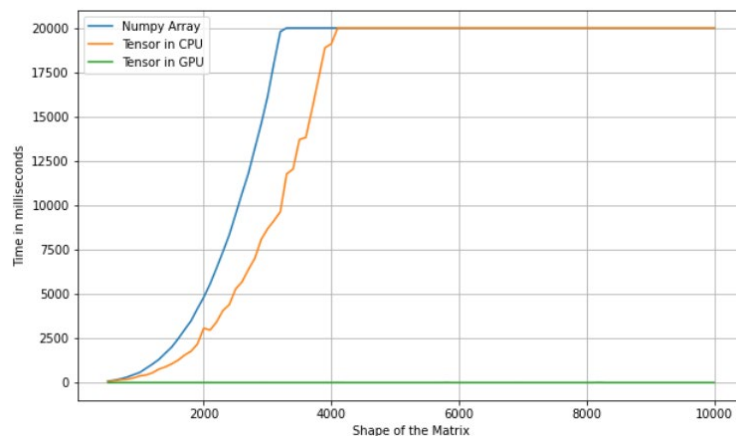
model *transfer learning* adalah  $4^{\text{th}}$  rank tensor, ilustrasi bentuk  $4^{\text{th}}$  rank tensor dapat dilihat pada gambar 2.21 dibawah ini [38].



Gambar 2. 21 Ilustrasi  $4^{\text{th}}$  Rank Tensor [38]

*NumPy* adalah *library python* yang menyediakan pembuatan *array* multi dimensi untuk digunakan dalam bermacam operasi matematika, logika, dan lain-lain [39].

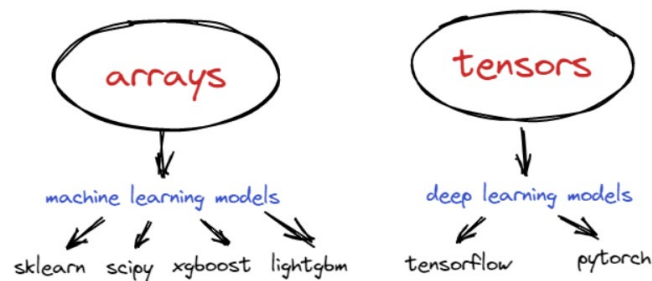
*NumPy array* sendiri umum digunakan dalam model *machine learning* karena kemudahan penggunaannya, namun untuk model *deep learning* seperti klasifikasi gambar, penggunaan *NumPy array* memiliki beberapa kekurangan jika dibandingkan dengan *tensor*, diantaranya adalah *NumPy array* tidak didukung dalam pemrosesan menggunakan *GPU* secara *native*, berbalik dengan *tensor* yang mendukung pemrosesan *GPU* untuk mempercepat pelatihan model [40].



Gambar 2. 22 Grafik Kecepatan Array dan Tensor [41]

Percobaan yang telah dilakukan [41] seperti yang terlihat dari hasil grafik pada gambar 2.22 diatas menunjukkan pemrosesan *tensor* dengan *GPU* menunjukkan kecepatan pemrosesan yang unggul dari *NumPy array* dalam berbagai ukuran *matrix*. Hasil percobaan lain untuk melihat performa *NumPy array* dengan *tensor* juga menunjukkan *tensor* jauh lebih unggul dengan kecepatan komputasi mencapai 52x lipat lebih cepat dibandingkan *NumPy array* [40].

Hal ini juga yang mendorong penggunaan *tensor* lebih umum pada model *deep learning*, karena kebutuhan *deep learning* adalah tingkat komputasi yang jauh lebih rumit dan dengan *tensor*, komputasi dapat diakselerasi *GPU* dengan sangat cepat. Terutama jika dalam penggunaan framework *deep learning* seperti *TensorFlow* maupun *PyTorch*. Sedangkan *NumPy array* dapat secara umum digunakan untuk berbagai proyek *machine learning* di *python* [42]. Gambar 2.23 dibawah ini menunjukkan *intended use* antara *NumPy array* dengan *tensor*.



Gambar 2. 23 Penggunaan Array dan Tensor [42]

### 2.2.12. Roboflow

*Roboflow* adalah perusahaan yang menyediakan layanan pengolahan data dalam bidang *computer vision*. Beberapa layanan yang ditawarkan antara lain pengolahan *dataset*, *labeling* gambar, *image preprocessing* dan *image augmentation*. *Roboflow* memberikan *tools* untuk pengguna umum dalam mengolah data untuk nantinya diaplikasikan ke dalam model *computer vision* [43].