

KLASIFIKASI SPESIES TANAMAN MONSTERA BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Muhamad Restu Assegaf¹, Agung Toto Wibowo²

^{1,2} Universitas Telkom, Bandung

restuassegaf@students.telkomuniversity.ac.id¹, agungtoto@telkomuniversity.ac.id²

Abstrak

Tanaman hias mencakup semua tumbuhan yang sengaja di tanam untuk memperindah suatu ruang atau tempat seperti menghias ruangan, kebun, upacara-upacara tertentu, karangan bunga dan lain sebagainya. Tanaman hias daun yang populer saat ini salah satunya adalah Monstera dengan nama latin Philodendron atau yang dikenal dengan sebutan janda bolong yang dimana tanaman ini sangat tidak biasa yang dimana bilahnya yang berlubang dan mencolok. Karena keunikannya ini banyak peminat dari tanaman Monstera ini membuat banyak pedagang nakal atau orang tidak bertanggung jawab untuk melakukan penipuan terhadap jual beli tanaman hias. Cara mengetahui perbedaan pada tanaman monster perlu dibuat sebuah system yang dapat mengklasifikasikan tanaman montera berdasarkan daunnya menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur VGG 16 digunakan pada penelitian ini dikarenakan memiliki kelebihan pada aspek ukuran yang kecil, kecepatan yang lebih cepat, penggunaan pembelajaran residual untuk konvergensi yang lebih cepat, generalisasi yang lebih baik, dan memecahkan masalah degradasi.

Kata Kunci: Klasifikasi Monstera, *Convolutional Neural Network (CNN)*, VGG 16.

Abstract

Ornamental plants include all plants that are intentionally planted to beautify a space or place such as decorating a room, garden, certain ceremonies, flower garlands, and so on. One of the leaf ornamental plants that are currently trending is the Latin name Monstera Philodendron or known as the perforated widow which is a very unusual plant where the blades are perforated and striking. Because of this uniqueness, many enthusiasts of this Monstera plant make many naughty traders or people not responsible for committing fraud against the sale and purchase of ornamental plants. How to find the differences of monstera plants, it is necessary to use the convolutional neural network (CNN) method and the VGG 16 architecture used in this research to make a system that can classify monstera plants according to leaves, because it has the Advantages are small size, high speed and residual learning. Faster convergence, better generalization and resolution of degradation problems.

Keywords: *Monstera Classification, Convolutional Neural Network (CNN), VGG16.*

1. Pendahuluan

1.1. Latar Belakang

Tanaman hias mencakup semua tumbuhan yang sengaja ditanam untuk memperindah suatu ruang atau tempat seperti menghias ruangan, kebun, upacara-upacara tertentu, karangan bunga dan lain sebagainya. Tanaman hias memiliki beberapa jenis yang salah satunya adalah tanaman hias daun. Ada banyak jenis tanaman hias, salah satunya adalah tanaman hias daun. Tanaman hias daun sedang menjadi trend saat ini. Harga tanaman hias daun tergantung pada umur dan tinggi tanaman yang bisa mencapai ratusan ribu hingga jutaan rupiah.

Monstera merupakan tanaman hias daun dengan nama latin Philodendron atau yang dikenal dengan sebutan janda bolong [1]. Tanaman ini sangat tidak biasa yang dimana bilahnya yang berlubang dan mencolok [2]. Pada beberapa daun tanaman Monstera memiliki serangkaian performansi yang besar dengan jarak yang teratur dan

meluas dari pelepah ke tepi daun dan bisa menembus tepi daun. *Monstera species Adansonii, Acuminata, dan Borsigiana* merupakan yang populer di Indonesia.

Karena keunikannya ini banyak orang awam yang tidak mengetahui jenis dari spesies tanaman monstera yang ada dijual di berbagai tempat, baik itu di kebun, pinggir jalan, dan lain sebagainya. Biasanya monstera dapat di bedakan berdasar lebar daun, jumlah lubang, tekstur daun, dan gelombang tepi daun [2]. Namun hanya sedikit orang yang paham dengan hal tersebut yang membuat kebanyakan orang susah dalam membedakan tanaman monstera berdasarkan spesiesnya.

Sebagai catatan, pada penelitian ini hasil akhir berupa model CNN. Sehingga untuk dapat melakukan klasifikasi citra monstera tidak bisa langsung begitu saja, melainkan perlu pemahaman terkait bagaimana menggunakan model CNN.

Cara mengetahui perbedaan pada tanaman *Monstera* perlu dibuat sebuah sistem yang dapat mengklasifikasikan tanaman monstera berdasarkan daunnya. Citra daun pada *Monstera* dipilih sebagai pengklasifikasian karena daun pada monstera sangat unik namun saat masih berumur muda daun *Monstera* belum memiliki lubang pada bagian daunnya, kemudian diolah menjadi data yang akan diklasifikasikan menggunakan metode *Convolutional Neural Network* (CNN). CNN merupakan metode ekstraksi fitur dari citra berupa jaringan syaraf tiruan [3]. CNN banyak digunakan sebagai metode pengenalan data yang berbentuk citra karena hasil akurasi dan kinerjanya yang tidak diragukan lagi pada penyimpanan informasi spasial dari data citra sehingga menghasilkan hasil klasifikasi yang baik.

Arsitektur VGG 16 digunakan pada penelitian ini [4] dikarenakan memiliki kelebihan pada aspek ukuran yang kecil, kecepatan yang lebih cepat, penggunaan pembelajaran residual untuk konvergensi yang lebih cepat, generalisasi yang lebih baik, dan memecahkan masalah degradasi.

1.2. Topik dan Batasan

Penelitian untuk tugas akhir ini adalah mencari model terbaik untuk klasifikasi tumbuhan monstera menggunakan metode *Convolutional Neural Network* (CNN), dan menganalisis performansi model yang diperoleh.

Batasan pada penelitian ini adalah system hanya dapat mengenali tiga label dari spesies tanaman monstera diantaranya *Adansonii, Acuminata, dan Borsigiana*. Dataset yang diambil berupa citra daun tanaman *Monstera* yang berumur muda hingga berumur sedang, yang dimana bentuk daun menjadi fokusnya. Pengambilan gambar citra diambil dari posisi bagian atas daun. Background untuk data train hanya memiliki 4 variasi background, sedangkan untuk data test tidak perlu variasi karena hanya untuk digunakan sebagai uji coba keakuratan.

1.3. Tujuan

Tujuan penelitian ini yaitu mencari model terbaik untuk mengklasifikasi tanaman *Monstera* dengan menggunakan metode *Convolutional Neural Network* (CNN) agar dapat membedakan dari tiga spesies yang sudah ditentukan pada batasan sebelumnya yakni *Adansonii, Acuminata, dan Borsigiana* dan menganalisis performansi dari model yang didapat.

2. Studi terkait

2.1. *Monstera*

Monstera adalah tanaman tropis dari keluarga *Araceae* yang berasal dari Meksiko serta beberapa bagian Amerika Selatan. *Monstera* terkenal dengan daun berongga alaminya yang mencolok. Pada beberapa daun tanaman *Monstera* memiliki serangkaian perforasi yang besar dengan jarak yang teratur dan meluas dari pelepah ke tepi daun dan bisa menembus tepi daun [2].

Tanaman monstera biasanya ditanam di luar ruangan sebagai tanaman hias dengan banyak ruang yang dimana tanah yang diperlukan tanah yang subur dan juga gembur. Jika ditanam pada tanah tidak dengan wadah atau yang sering disebut dengan pot lebih baik ditanam didekat pohon atau benda apapun yang memanjang agar tanaman monstera dapat merambat pada benda tersebut. Hanya saja jika ditanam ditanah akan membuat cukup merepotkan dalam budidaya yang dimana menjaga kelembaban tanah sedikit lembab [5].

Pada tanaman monstera dapat dilakukan pengklasifikasian untuk dapat membedakan antara monstera dengan spesies *Acuminata, Adansonii, dan Borsigiana* menggunakan metode CNN dengan arsitektur VGG16 yang dimana data diambil dari lapangan. Lalu dengan kumpulan data tersebut akan didapat hasil tingkat keakuratan dalam mengidentifikasi objek pada spesies tanaman monstera.

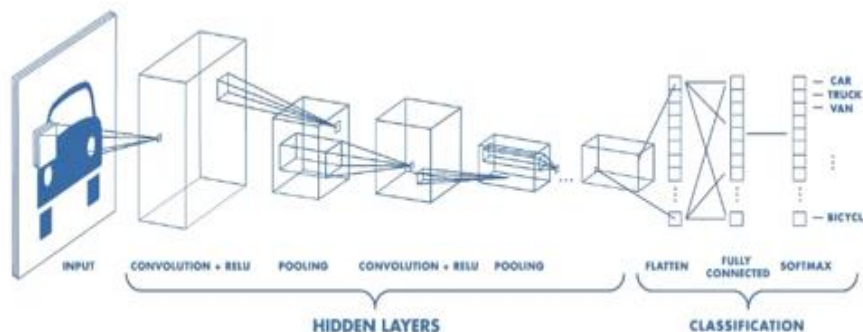
Pada jurnal [6] membahas tentang pengklasifikasian bibit untuk meningkatkan produksi tanaman sekaligus mengurangi biaya produksi. Penelitian mengidentifikasi bibit Epigeal dan Hypogeal menggunakan metode CNN dan arsitektur VGG16 yang dimana dataset diambil dari “ImagNet” dan “Aarhus University Signal, University of Denmark Selatan”. Kumpulan data yang berisi sekitar 5.000 gambar dengan 960 tumbuhan unik milik 12 spesies yang terdiri dari *Fat Hen*, *Small-flowered Cranesbill*, *Maize*, *Silky-bent*, *Sugar beet*, *Common wheat*, *Cleavers*, *Common Chickweed*, *Scentless Mayweed*, *Black-grass*, dan *Charlock*, *Shepherds Purse*, penulis menghasilkan 99,48% tingkat keakuratan dalam mengidentifikasi objek pada bibit tanaman.

2.2. Pengolahan Citra Digital

Pengolahan citra adalah proses mengolah dan menganalisis citra dengan bantuan komputer, dimana bentuk citra dapat berupa foto. [7]. Pengolahan citra berupa gambar (*image processing*) merupakan suatu proses yang melakukan *input* berupa gambar dan memiliki *output* berupa gambar juga [8].

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network atau yang disingkat dengan CNN adalah suatu metode proses pada citra digital yang dimana dapat memproses citra pada gambar atau suara [9]. Metode CNN sering digunakan pada penelitian sebagai pengklasifikasian dan pendeteksi pada suatu objek gambar. CNN sendiri memiliki struktur yang dimulai dari *input* sebagai tahap awal, proses ekstrasi fitur yang terdiri dari beberapa lapisan yakni *hidden layer*, *convolutional layer*, fungsi aktivasi yang biasa di sebut dengan ReLU, dan yang terakhir adalah pooling, dan diakhiri output sebagai tahap terakhir pada CNN. Proses klasifikasi terdiri dari *fully-connected* dan fungsi aktivasi (*softmax*) yang outputnya adalah hasil klasifikasi.



Gambar 1. Tahapan Convolutional Neural Network

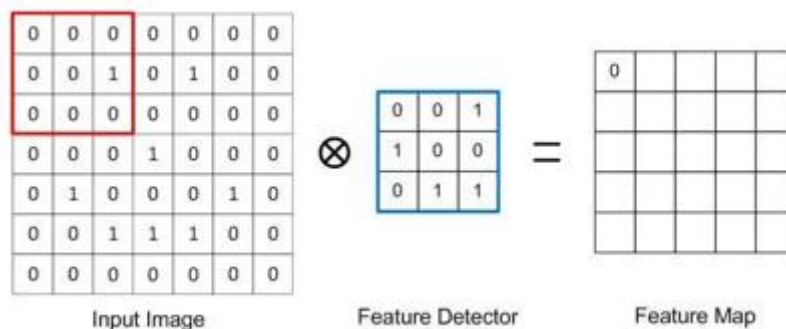
Sumber Gambar: <https://towardsdatascience.com/>

a. Input Layer

Input layer merupakan tahapan awal pada CNN yang dimana pada tahap ini memasukkan data awal yang berupa citra kedalam sistem agar dapat di proses oleh lapisan CNN selanjutnya.

b. Convolution Layer

Convolution layer merupakan tahapan yang berlaku sebagai filter atau kernel yang memiliki bobot acak diawal yang dimana bobot ini akan di perbaharui pada saat melakukan *training*. Tujuan dari *convolution layer* untuk mengekstrak fitur dari citra input. Filter ini akan dikalikan untuk berpindah dari sudut kiri atas ke sudut kanan bawah atau seluruh gambar. *Output* dari *convolution layer* biasa disebut *feature map*.



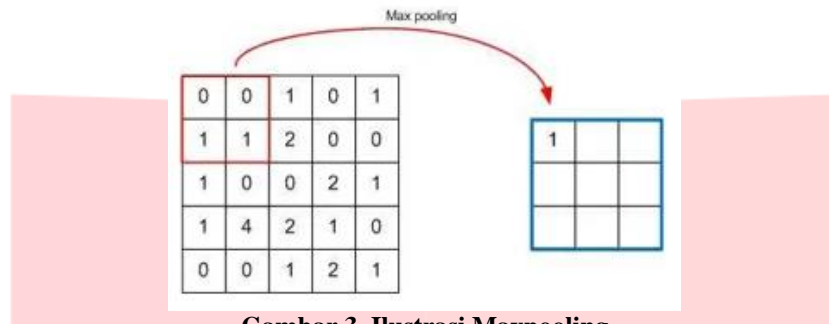
Gambar 2. Ilustrasi Convolution Layer

c. Fungsi Aktivasi: ReLU

Fungsi Aktivasi ReLU (*Rectified Linear Unit*) dilakukan setelah melakukan proses *convolution*. Nilai-nilai pada *feature map* akan di seleksi lagi yang dimana nilai yang bernilai negatif akan dijadikan nilai 0 oleh aktivasi ReLU.

d. Pooling Layer

Pooling Layer merupakan sebuah filter yang bergerak di area *feature map* secara menyeluruh. Digunakan *Maxpooling* pada *layer* ini untuk diambil nilai terbesar dari hasil *feature map*. Tujuan *pooling layer* adalah mereduksi ukuran sebuah data citra yang dimana ukuran gambar akan lebih kecil dari sebelumnya agar dapat mempercepat komputasi.



Gambar 3. Ilustrasi Maxpooling

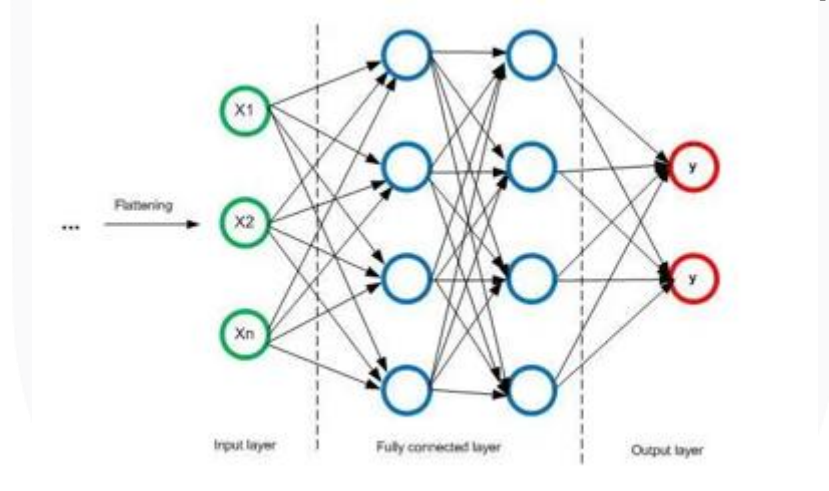
Sumber Gambar: <https://towardsdatascience.com/>

e. Flatten Layer

Flatten Layer adalah tahapan yang dimana mengubah bentuk *matrix* pada *feature map* menjadi *vektor* (*array satu dimensi*) sehingga dapat digunakan sebagai *input* di tahap *fully connected*.

f. Fully Connected Layer

Pada *Fully Connected Layer* terdapat *node-node* yang terhubung satu sama lain antara *node* depan dan *node* belakang yang disebut dengan (*forward propagation*) dan (*back propagation*). Hal ini berperan dalam penentuan pengklasifikasian suatu objek berdasarkan probabilitas yang dimana bergantung pada setiap *epoch* nya.



Gambar 4. Ilustrasi Fully Connected Layer

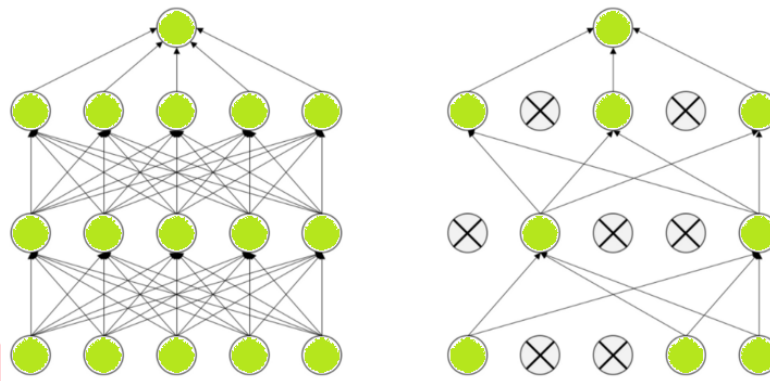
Sumber Gambar: <https://towardsdatascience.com/>

g. Fungsi Aktivasi: Softmax

Softmax biasa digunakan pada kasus klasifikasi lebih dari 2 kelas (*multi-class*) yang mana prosesnya berlangsung pada *output layer* pada tahapan *fully connected*. Kelebihan menggunakan *Softmax* yaitu rentang probabilitasnya berkisar dari 0 hingga 1. Tujuan proses Aktivasi *Softmax* untuk menghitung probabilitas pada setiap kelas dan menentukan kelas untuk *input* citra yang diproses sebelumnya.

h. Dropout Regularization

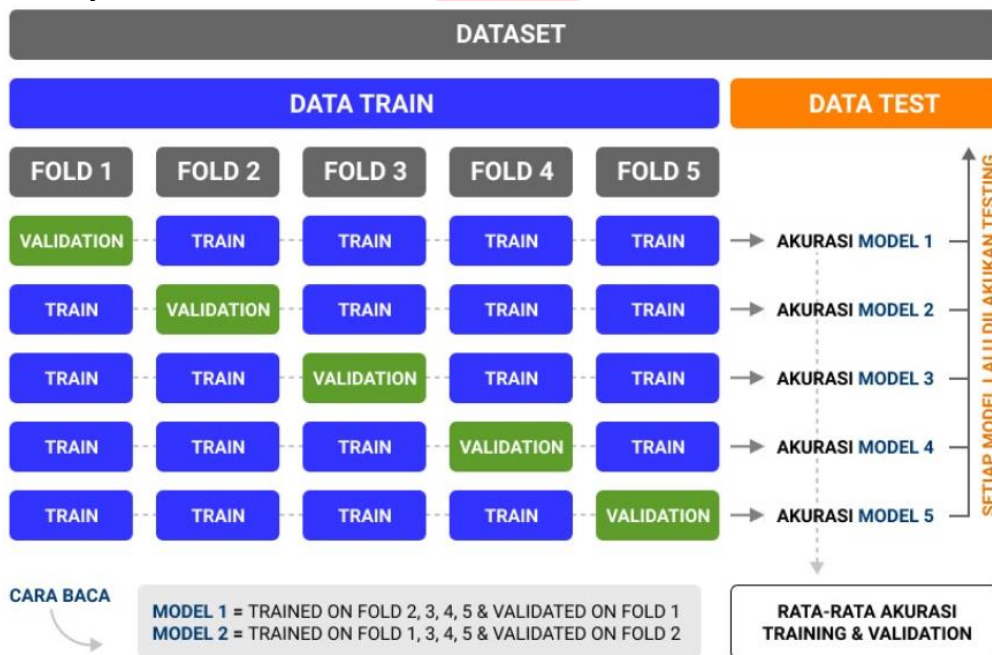
Dropout merupakan sebuah teknik untuk mengurangi *overfitting* agar dapat mempercepat proses pelatihan (*training*) yang dimana beberapa *neuron* yang dipilih secara acak tidak digunakan selama *training*.



Gambar 5. Ilustrasi Dropout

2.4. Cross Validation : K-Fold

Cross validation adalah sebuah metode statistik, biasa digunakan untuk mengevaluasi model dimana data *train* akan di bagi menjadi dua bagian, yaitu data *train* dan data *validation* dengan tujuan mengevaluasi dan membandingkan model yang dibangun. Model yang akan dilatih menggunakan data *train* dan data yang akan divalidasi menggunakan data *validation*. Cross validation akan membagi subset sebanyak K dan dilakukan perulangan sebanyak K.

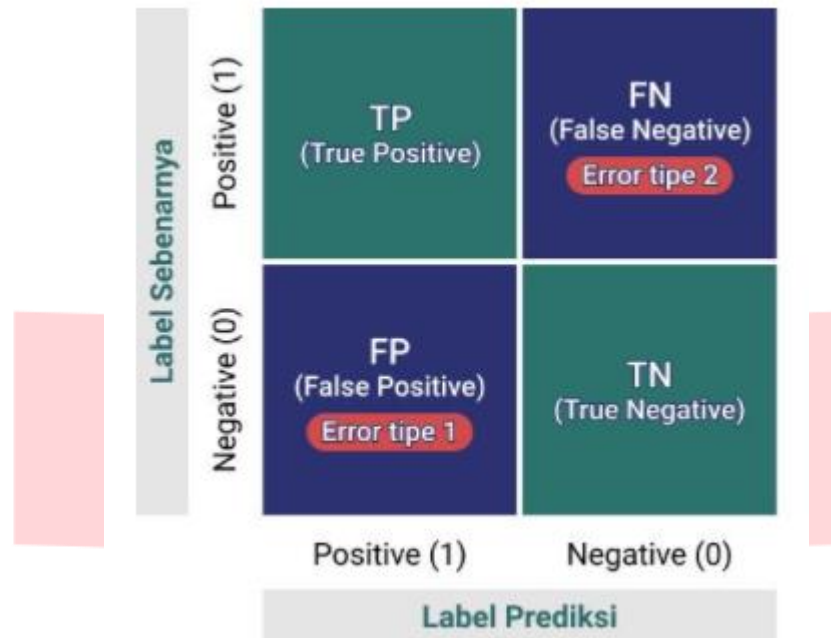


Gambar 6. ilustrasi Cross Validation : K-Fold

Sumber Gambar: <https://satishgunjal.com/kfold/>

2.5. Confusion Matrix

Confusion Matrix digunakan untuk membandingkan nilai aktual dengan nilai prediksi. Baris pada *matrix* mewakili *class* yang sebenarnya, sedangkan kolom mewakili *class* yang diprediksi. Nilai yang dikembalikan oleh Confusion Matrix dibagi ke dalam 4 kategori sebagai berikut:



Gambar 7. Confusion Matrix

Keterangan kategori pada gambar di atas adalah *True Positive* (TP) sebagai prediksi positif dan nilai sebenarnya positif, *True Negative* (TN) sebagai prediksi negatif dan nilai sebenarnya negative, *False Positive* (FP) sebagai prediksi positif dan nilai sebenarnya negative, dan *False Negative* (FN) sebagai prediksi negatif dan nilai sebenarnya positif.

2.6. Classification Report

Classification Report merupakan data dari *Confusion Matrix* yang dapat digunakan untuk menghitung *matrix* yang dapat mengukur kinerja model, yang terdiri dari 4 *matrix* sebagai berikut :

Classification Report Metrics	
$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$	
$precision = \frac{TP}{TP + FP}$	
$recall = \frac{TP}{TP + FN}$	
$f1score = 2 \times \frac{recall \times precision}{recall + precision}$	

Gambar 8. Classification Report

- Accuracy : Seberapa akurat model dalam mengklasifikasikan.
- Precision : Akurasi antara data yang diminta dengan hasil prediksi.
- Recall : Keberhasilan model dalam menemukan kembali sebuah informasi.
- F1-Score : Perbandingan rata-rata Precision dan Recall yang dibobotkan.

2.7. Underfitting & Overfitting

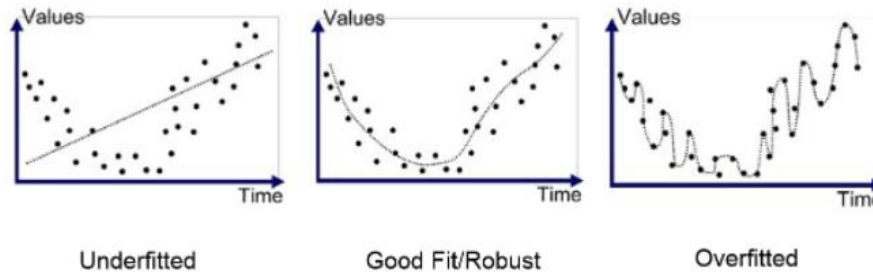
Ketika membuat sebuah model CNN terjadi suatu kondisi yang disebut sengan *underfitting* dan *overfitting* dimana model yang mengalami kondisi tersebut tidak akan mampu memprediksi suatu kelas dengan benar [10]

a. Underfitting

Underfitting adalah suatu kondisi dimana model tidak dapat melihat logika di balik data, yang mengakibatkan tidak mungkin untuk membuat prediksi pada dataset *training* maupun dataset *testing* dengan tepat. biasanya memiliki loss tinggi dan akurasi rendah.

b. Overfitting

Overfitting adalah suatu kondisi dimana memiliki loss rendah dan akurasi rendah yang disebabkan hanya terlalu berfokus pada data training tertentu saja.



Gambar 9. Underfitting, Good Fitting dan Overfitting

Sumber Gambar: <https://medium.com/greyatom/>

Model yang baik dapat menginterpretasikan data tanpa terpengaruh oleh *noise* dan memiliki *loss* rendah dengan akurasi tinggi. Model tidak akan cocok dengan semua data, tetapi dapat menjelaskan kelompok data.

2.8. Hyperparameter

Hyperparameter merupakan sebuah parameter yang nilainya ditentukan oleh perancang model CNN. teknik yang digunakan *Hyperparameter* adalah *experimental* untuk mendapatkan hasil *score* akurasi yang baik.

a. Input Shape

Sebelum melakukan input shape ada baiknya citra yang akan digunakan tidak *blur*, gelap, atau gangguan noise lainnya sebelum di augmentasi sebelum model CNN mempelajari features pada citra gambar hal ini bertujuan agar model CNN tidak mengalami *overfitting*. Selain itu ukuran gambar yang terlalu kecil bisa menjadi faktor CNN tidak bisa mempelajari features pada citra dengan baik.

b. Batch Size

Pada saat *training* jika dilakukan sekaligus akan memberatkan proses pelatihan secara komputasi. Untuk mengatasi hal ini batch size berfungsi untuk memecah dataset menjadi beberapa bagian kecil.

c. Epoch

Epoch berfungsi untuk mengatur jumlah algoritma dalam memperbarui bobot dari keseluruhan dataset. Jumlah epoch menentukan seberapa banyak pelatihan terhadap model dengan bobot yang berbeda-beda yang bertujuan agar generalisasi data menjadi lebih bagus. Semakin banyak epoch maka akan semakin banyak model akan melakukan training.

d. Jumlah Filter & Filter Size

Jumlah filter pada *Convolutional Layer* bisa disesuaikan, secara umum banyak peneliti yang menggunakan nilai 2^n dan untuk ukuran filter pada *Convolutional Layer* dan *Pooling Layer* bisa disesuaikan juga yang umumnya banyak peneliti menggunakan ukuran filter *Convolutional* yaitu 3×3 atau 5×5 dengan *stride*=1 atau *stride*=2, dan untuk *Pooling* yaitu 2×2 dengan *stride*=2.

e. Stride

Stride merupakan parameter yang menentukan jumlah pergeseran pada filter. Contohnya jika nilai *stride*=1 maka filter akan bergeser sebanyak 1 piksel secara horizontal lalu vertikal terus berulang sehingga seluruh area terkena efek filter.

f. Padding

Padding berfungsi untuk menentukan jumlah piksel yang mengandung nilai 0, kemudian menambahkannya ke setiap sisidari *input* dimensi map. Hal ini bertujuan untuk memanipulasi dimensi *output* dari proses *convolutional* atau *pooling*, sehingga dimensi *output* dapat diukur agar tetap sama seperti dimensi *input*.

g. Optimizer

Biasanya para ahli menggunakan *optimizer* : *Adam Optimizer* karena *Optimizer* ini terkenal memiliki hasil yang baik dan dianggap sebagai *optimizer* yang cepat dalam mencapai *loss* minimum (*converge*). Misalnya, dalam kasus klasifikasi citra *optimizer adam* dapat membuat akurasi menjadi lebih baik [11].

h. Learning Rate

Learning Rate merupakan parameter yang mengontrol cepat atau lambatnya proses model dalam mempelajari data saat pelatihan (*training*). Hal ini sangat berpengaruh terhadap hasil dari skor akurasi pada model yang dilatih.

i. Jumlah Hidden Layer & Jumlah Node/Neuron

Jumlah *Hidden Layer* pada tahap *Fully Connected* bisa disesuaikan, biasanya para ahli menggunakan hidden layer dimulai dari 2 *hidden layer*[12]. Kemudian untuk *neuron* pada setiap *Hidden Layer* bisa disesuaikan juga, para ahli biasanya menggunakan nilai 2^n (misalnya: 8/16/32/64/128/256) [10].

2.9. VGG16

Pada tahun 2015, Kaiming He, Xiangyu Zhang, Shaoqing Ren, dan Jian Sun dari Microsoft Research membuat arsitektur dengan akurasi yang lebih bagus dengan kesalahan lebih sedikit yaitu 3,57% pada set pengujian ImageNet yang melebihi kadalaman VGG [12] dengan kedalaman hingga 152 lapisan. Pencapaian mereka membuat mereka memenangkan tempat pertama di Tugas klasifikasi ILSVRC 2015. Tidak hanya itu, mereka juga melakukan analisis pada CIFAR-10 dengan 100 dan 1000 lapisan.

Pada jurnal [13] membahas tentang metode pengenalan ikan dengan metode CNN dan arsitektur VGG16. Dataset ikan dalam penelitian ini terdiri dari 50 spesies, masing-masing spesies terdapat 15 gambar yang dimana 10 gambar untuk penelitian dan 5 gambar untuk pengujian. Pada penelitian ini dilatih 4 model dengan hasil penelitian menunjukkan *RGB color space image* 92.4%, *canny filter image* 80.4%, *blending image* 75.6%, dan *blending image* dicampur dengan *RGB image* 96.4%.

Tabel 1. Arsitektur dari VGG16

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

3. Sistem yang Dibangun

3.1. Deskripsi Umum

Penelitian yang dilakukan pada penelitian ini untuk mengklasifikasi tanaman monstera berdasarkan citra daun menggunakan metode CNN dengan arsitektur VGG16 untuk mengklasifikasikan 3 spesies dari monstera.

Peneliti melakukan pengambilan dataset dilapangan untuk dijadikan dataset penelitian. Setelah itu dataset yang sudah diambil sebelumnya akan melalui tahap augmentasi yang dilanjutkannya pada pembagian dataset untuk data *train* dan data *test*. Data *train* akan dibagi menjadi 2 skenario yakni data *train* dengan *background* dan data *train* tanpa *background* yang dimana selanjutnya data *train* akan dibagi menjadi 80% data *train* dan 20% data *validation* dengan menggunakan *K-Fold Cross Validation* dengan nilai $K=5$ untuk mengevaluasi model. Pada




penelitian ini hanya menggunakan arsitektur VGG16 yang diberikan *dropout* berbeda (0.1 , 0.3 , dan 0.5). Arsitektur VGG16 akan dilatih dengan skenario dataset dengan *background* dan skenario dataset tanpa *background*. Selain itu, juga akan diterapkan *hyperparameter* berupa *learning rate* dengan nilai 0.0001 dan 0.0003. Total ada 60 (12 x 5 *fold*) model yang didapat dari :

- Skenario dataset dengan *background* dan skenario dataset tanpa *background* = 2
- *Learning rate* dengan nilai 0.0001 dan 0.0003 = 2
- *Dropout* berbeda (0.1 , 0.3 , dan 0.5) = 3
- *K-Fold Cross Validation* = 5




Tabel Data 2. Train dengan Background

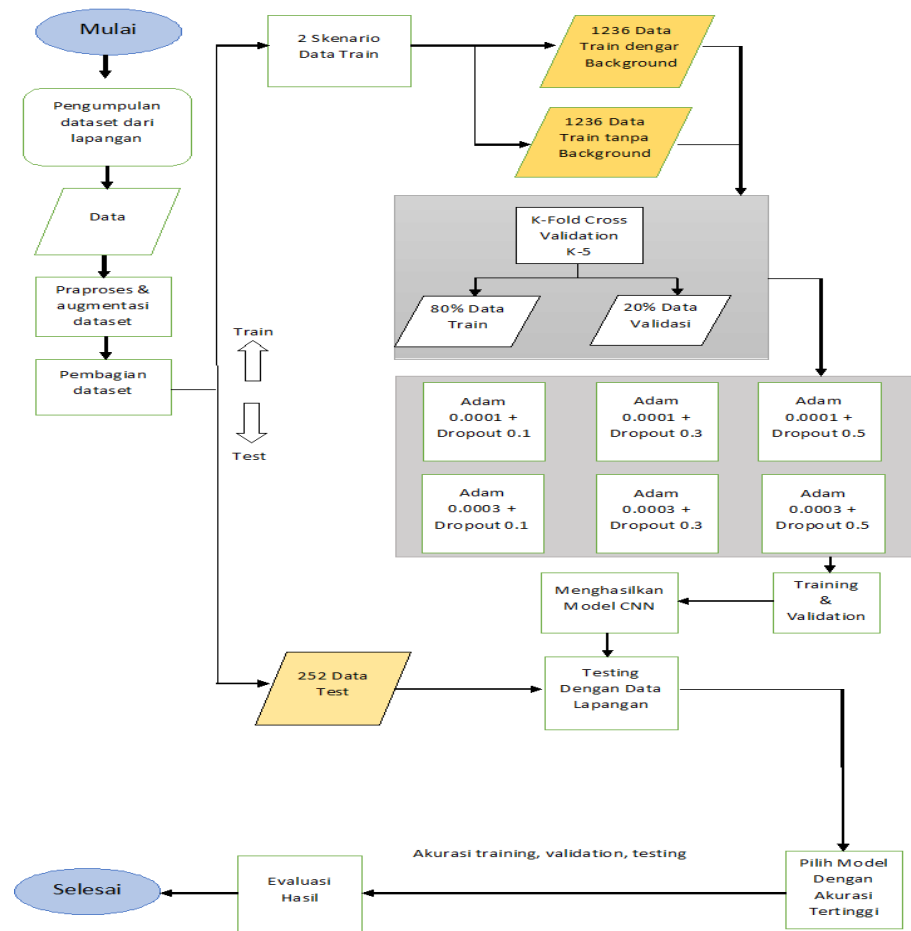
Data Train dengan Background		
Acuminata	Adansonii	Borsigiana
		

Tabel Data 3. Train tanpa Background

Data Train tanpa Background		
Acuminata	Adansonii	Borsigiana
		

Tabel 4. Data Test

Data Test		
Acuminata	Adansonii	Borsigiana
		



Gambar 10. Deskripsi Umum

3.2. Pengumpulan DATASET

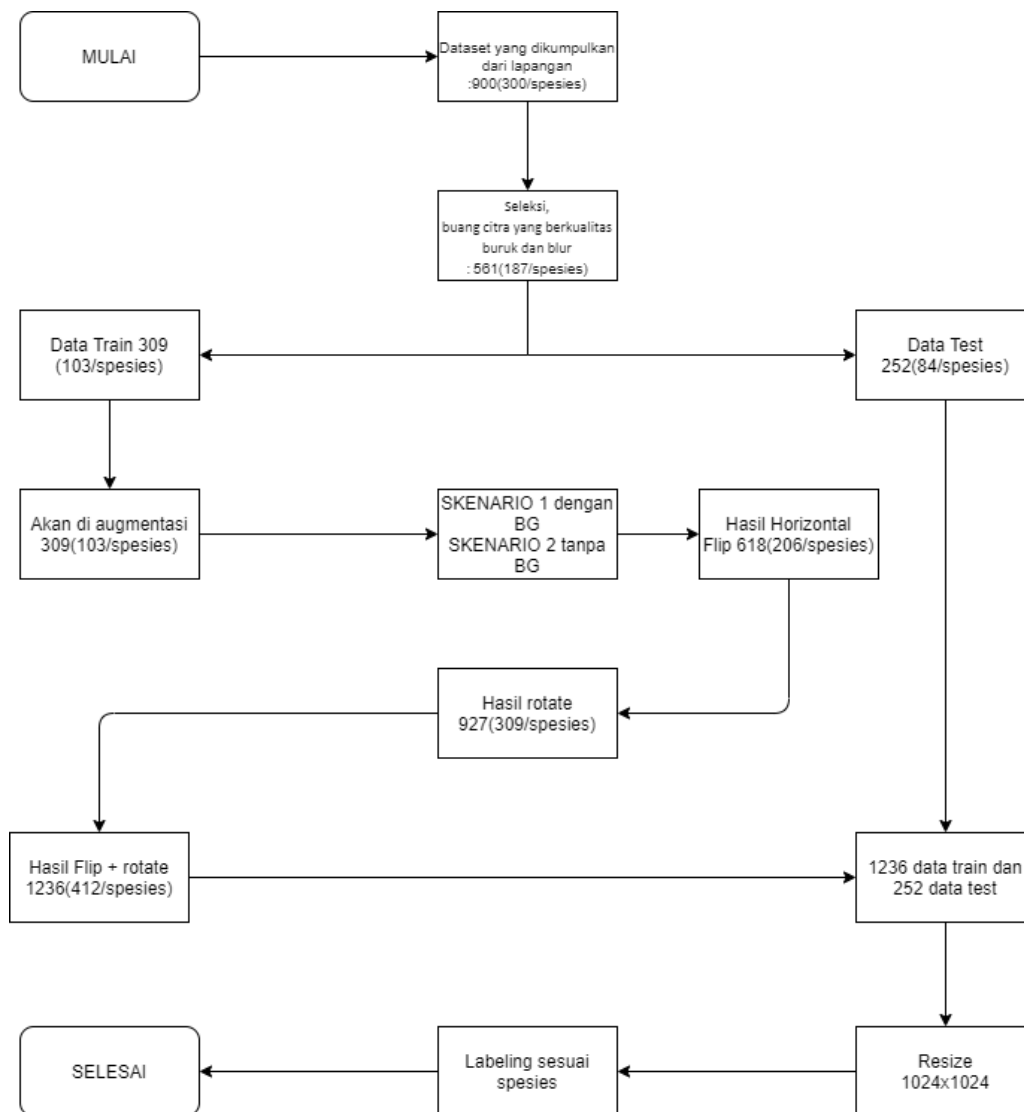
Dataset yang digunakan dalam penelitian ini berupa gambar/citra dari tiga spesies tanaman *Monstera* yaitu *Acuminata*, *Adansonii*, dan *Borsigiana*. Dataset diperoleh dengan mengambil dataset secara langsung di lapangan yang dimana penulis dan beberapa rekan penelitian mengunjungi sebuah tempat budidaya berbagai macam tanaman yaitu Grace Rose Farm. Ada 900 citra yang diperoleh, yang terdiri dari 300 citra per spesies (*Acuminata*, *Adansonii*, dan *Borsigiana*).

Pengambilan data dilakukan dengan cara mengambil foto daun dari 3 spesies monstera yakni *Acuminata*, *Adansonii*, dan *Borsigiana* yang dimana pengambilan gambar diambil dari bagian permukaan daun. Dilakukan teknik *burst* untuk pengambilan gambar dari ke 3 spesies monstera tersebut.

Pengambilan foto diambil di dua tempat yaitu Bandung dan Lombok. HandPhone yang digunakan untuk mengambil foto adalah Xiaomi Mi 6 Versi 6/128 GB dengan *Chipset* Qualcomm MSM8998 Snapdragon 835 (10 nm), RAM & Memori: 6 GB & 128 GB, kamera Dual 12 MP + 12 MP, dan Baterai Li-Po 3350 mAh. Jarak pengambilan gambar antara HandPhone yang digunakan sepanjang 15cm. Pengambilan gambar dilakukan pada jam 11.00 pagi dan jam 14.00 siang.

3.3. Pra Proses dan Augmentasi Dataset

Pra Proses dan augmentasi dataset dilakukan terlebih dahulu sebelum melakukan pembagian dataset menjadi *data train* dan *data test*. dataset yang berjumlah 900 akan diseleksi agar tidak mendapatkan citra yang buruk/blur yang dimana dapat mempengaruhi proses pembelajaran model CNN. didapat 561 citra yang terdiri dari 496 citra per spesies (*Acuminata*, *Adansonii*, dan *Borsigiana*). Karna penulis melakukan pengujian skenario perbedaan background dataset (tanpa *background* dengan ada *background*) dilakukan sebagai berikut :



Gambar 11. Tahapan Pra Proses dan Augmentasi

Pada pra proses dilakukan 2 tahap yaitu tahap *crop* dan tahap *resize* yang dimana bertujuan untuk memperkecil ukuran pada gambar karena ukuran gambar asli sangat besar jika digunakan pada saat melakukan proses training. Selain itu proses dalam melakukan training akan cepat karena ukuran gambar yang sudah diperkecil.

3.3.1. *Resize*







Tahap *resize* dilakukan agar gambar asli diubah ukuran pixelnya menjadi lebih kecil agar pada proses komputasi dalam pencarian dapat berjalan lebih cepat.

3.3.2. *Cropping*

Tahap *crop* atau potong dilakukan bertujuan untuk menghilangkan bagian-bagian yang tidak penting pada gambar asli agar pada saat proses klasifikasi hanya tertuju pada objek yang diinginkan.

Pada gambar diatas dimiliki 561 citra yang akan dibagi menjadi 309 data train dan 252 data test. 309 data train akan dilakukan augmentasi yang dimana dibagi menjadi 2 skenario yakni skenario dengan *background* dan skenario tanpa *background*. Penghilangan *background* dilakukan dengan menggunakan *EditPhotosForFree* yang berbasis *web*.

Tabel 5. Skenario dengan *background* dan skenario tanpa *background*

Dengan <i>Background</i>	Tanpa <i>Background</i>
 Acuminata	 Acuminata
 Adansonii	 Adansonii
 Borsigiana	 Borsigiana

Augmentasi adalah proses dimana gambar dimodifikasi dengan cara rotasi dan *flipping* pada penelitian ini sehingga komputer mendeteksi sebagai gambar yang berbeda. Ini dilakukan untuk meningkatkan akurasi dari model CNN tahap augmentasi terdiri dari :

3.3.3. *Flipping*

Pada proses ini penulis melakukan operasi *flipping* yaitu suatu perubahan orientasi citra secara Horizontal dengan tujuan menambah variatif suatu data.

3.3.4. *Rotate*

Proses rotasi, yaitu suatu operasi memutar suatu citra terhadap titik pusatnya terhadap arah jarum jam atau berlawanan jarum jam. Hal ini dilakukan bertujuan menambah data agar semakin variatif sehingga model dapat mempelajari data baru dengan mudah.

3.3.5. *Flipping + Rotate*

Setelah dicerminkan terhadap citra itu sendiri kemudian akan dilakukan rotasi terhadap arah jarum jam atau berlawanan jarum jam.

3.3.6. Data Test / Data Lapangan

Sama halnya dengan data Train, data Testing lapangan perlu dilakukan pemilihan data dari data yang blur namun bedanya pada data Train dilakukan proses Augmentasi data hanya dengan melakukan operasi *Resize* saja, dan data Testing hanya berjumlah 84 dalam setiap spesiesnya. Hal ini dikarenakan data Testing adalah data yang akan diujikan kepada data Train nanti untuk menentukan seberapa akurat data yang telah dipelajari dengan data baru yang tidak dilakukan pelatihan (data Train).

3.4. K-Fold Cross Validation

Untuk melihat performansi model CNN dalam pelatihan (*training*) yang akan dijalankan nanti, digunakanlah *K-Fold Cross Validation*. Dalam penelitian ini akan digunakan nilai $K=5$, ini berarti 80% data *train* dan 20% data *validation*.

3.5. Perancangan CNN

Pada penelitian ini menggunakan arsitektur VGG16 dengan menggunakan *dropout*. Pada arsitektur yang digunakan akan dilatih pada data (Training & validation) terhadap skenario dataset dengan *background* dan skenario dataset tanpa *background*. Untuk memaksimalkan hasil akurasi kemudian perlu mengatur hyperparameter seperti *learning rate*, *dropout*, dan lain-lain. Karena penulis mengatur *fold* = 5 sehingga model yang dihasilkan berjumlah 5 model yang dimana setiap model akan diujikan ke data test sehingga di dapatlah model terbaik untuk setiap versi data dan skenario yang diuji.

4. Evaluasi

4.1. Skenario Pengujian & Hasil Penelitian

Proses *training* dilakukan dengan menggunakan dua skenario yang sudah diolah sebelumnya :

- Skenario 1 → 1236 data train dengan *background* (412 /spesies)
- Skenario 2 → 1236 data train tanpa *background* (412 /spesies)

Training dilakukan menggunakan teknik *K-Fold Cross Validation* dengan nilai $K=5$, ini berarti 80% data *train* dan 20% data *validation*. *Hyperparameter* yang digunakan :

- Input Shape Citra : 224x224x3
- Batch Size : 64
- Epoch : 10
- Optimizer : Adam
- Learning Rate: 0.0001, 0.0003

Arsitektur CNN yang digunakan adalah VGG16 dengan *dropout* yang berbeda-beda (0.1 , 0.3 , dan 0.5). Terdapat total 60 model CNN lengkap beserta *score* hasil *testing* dari beberapa alat ukur model yakni *precision*, *recall*, *f1-score* dan *accuracy* dari lapangan dapat dilihat pada lampiran.

Setelah mendapatkan nilai *score* hasil *testing* dari beberapa alat ukur model, *precision*, *recall*, *f1-score* dan *accuracy*, maka akan dicari rata-rata formansi algoritma VGG16 terhadap data testing dari 5 *fold k-fold* untuk setiap model di dapat sebagai berikut :

Tabel 6. Rata-rata score testing dari 5 fold

skenario		Rata-rata dari 5 fold			
		precision	recall	F1-score	akurasi
Dengan BG	adam 0.0001 + Dropout 0.1	86.67%	83.40%	83.47%	83.49%
	adam 0.0001 + Dropout 0.3	84.20%	78.13%	77.47%	78.17%
	adam 0.0001 + Dropout 0.5	82.00%	78.07%	77.20%	77.93%
	adam 0.0003 + Dropout 0.1	86.20%	83.73%	84.33%	83.73%
	adam 0.0003 + Dropout 0.3	85.80%	82.67%	82.47%	82.70%
	adam 0.0003 + Dropout 0.5	84.13%	78.93%	78.67%	78.89%
Tanpa BG	adam 0.0001 + Dropout 0.1	69.13%	69.33%	66.87%	69.36%
	adam 0.0001 + Dropout 0.3	71.87%	72.00%	69.40%	72.14%
	adam 0.0001 + Dropout 0.5	72.20%	71.13%	68.40%	71.11%
	adam 0.0003 + Dropout 0.1	69.87%	69.00%	66.00%	69.04%
	adam 0.0003 + Dropout 0.3	69.87%	70.53%	67.13%	70.63%
	adam 0.0003 + Dropout 0.5	68.60%	67.13%	61.60%	67.06%

Dilihat dari tabel rata-rata *score testing* dari ke 5 *fold* pada *precision* untuk yang terbaik dengan *background* ada pada model dengan *learning rate* 0.0001 dengan *dropout* 0.1. Sedangkan untuk yang tanpa *background* ada pada *learning rate* 0.0001 dengan *dropout* 0.5. Untuk , *recall*, *f1-score* dan *accuracy* memiliki letak model yang sama, pada dengan *background* ada pada *learning rate* 0.0003 dengan *dropout* 0.1 dan untuk tanpa *background* ada *learning rate* 0.0001 dengan *dropout* 0.3.

4.2. Hasil Model Terbaik dengan Background

Dari hasil pelatihan (*training*) dan pengujian (*testing*), didapat dua model CNN dengan *score* akurasi tertinggi. Pada tahap ini akan dilakukan analisis terhadap kedua model tersebut dengan menggunakan *Confusion Matrix* dan *Classification Report*.

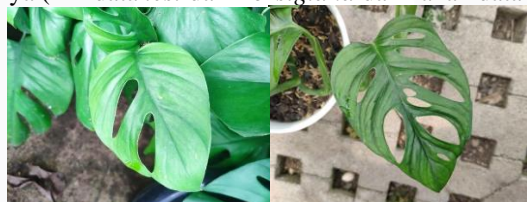
1. learning rate 0.0003 dengan dropout 0.3



Gambar 12. Confusion Matrix dan Classification Report model CNN terbaik ke-1

Berdasarkan informasi pada *Confusion Matrix* dan *Classification Report* diatas, didapatkan beberapa kesimpulan sebagai berikut :

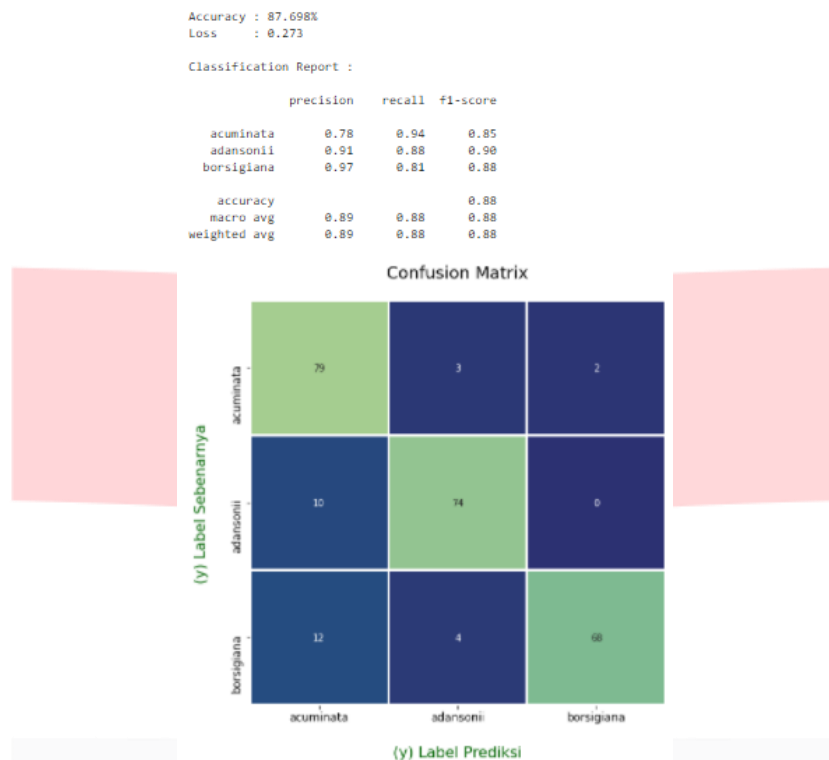
- Hasil *testing* paling banyak salah prediksi yaitu *Borsigiana*. Dimana dari 84 citra *Borsigiana* 10 citra diantaranya dikenali sebagai *Acuminata* yang disebabkan oleh kemiripan beberapa dataset *Borsigiana* dengan dataset *Acuminata*. Contohnya (kiri data *test* dari *Borsigiana* dan kanan data *train* dari *Acuminata*) :



- Hasil *testing* paling banyak salah prediksi ke 2 yaitu *Adansonii*. Dimana dari 84 citra *Adansonii* 7 citra diantaranya dikenali sebagai *Acuminata* yang disebabkan oleh kemiripan beberapa dataset *Adansonii* dengan dataset *Acuminata*. Contohnya (kiri data *test* dari *Adansonii* dan kanan data *train* dari *Acuminata*) :



2. learning rate 0.0003 dengan dropout 0.1



Gambar 13. Confusion Matrix dan Classification Report model CNN terbaik ke-2

Berdasarkan informasi pada *Confusion Matrix* dan *Classification Report* diatas, didapatkan beberapa kesimpulan sebagai berikut :

- Hasil *testing* paling banyak salah prediksi yaitu *Borsigiana*. Dimana dari 84 citra *Borsigiana* 12 citra diantaranya dikenali sebagai *Acuminata* yang disebabkan oleh kemiripan beberapa dataset *Borsigiana* dengan dataset *Acuminata*. Contohnya (kiri data *test* dari *Borsigiana* dan kanan data *train* dari *Acuminata*) :



- Hasil *testing* paling banyak salah prediksi ke 2 yaitu *Adansonii*. Dimana dari 84 citra *Adansonii* 10 citra diantaranya dikenali sebagai *Acuminata* yang disebabkan oleh kemiripan beberapa dataset *Adansonii* dengan dataset *Acuminata*. Contohnya (kiri data *test* dari *Adansonii* dan kanan data *train* dari *Acuminata*) :



5. Kesimpulan

Dari hasil penelitian ini didapat model CNN terbaik yaitu model CNN dengan *hyperparameter learning rate* 0.0003 dengan *dropout* 0.3 dan *learning rate* 0.0003 dengan *dropout* 0.1. Skenario dengan *background* menghasilkan akurasi yang lebih baik dibandingkan dengan akurasi tanpa *background*.

Dengan model CNN yang telah dihasilkan tersebut, masyarakat luas terlebih orang awam dengan spesies monstera dapat memanfaatkan model untuk dilakukan klasifikasi, dengan catatan perlu pemahaman terkait bagaimana cara menjalankan model CNN.

Pada jurnal [14] disimpulkan bahwa penggunaan data *train* tanpa *background* memiliki akurasi yang lebih bagus, dikarenakan memang tidak ada noise pada *background* nya. Namun hasil pada percobaan pada penelitian ini adalah didapatkan akurasi yang lebih bagus pada data *train* dengan *background* dikarenakan data *train* dengan *background* dan data test memiliki *background* yang serupa yang menyebabkan saat melakukan testing hasil akurasi yang lebih bagus ada pada data *train* dengan *background*. Kesimpulannya pada studi kasus/dataset yang digunakan pada penelitian ini adalah data *train* dengan *background* memiliki akurasi yang lebih bagus dibandingkan data *train* tanpa *background*.

Saran

Saran untuk pengembangan kedepannya adalah:

- Memperbanyak jumlah dataset yang lebih bervariasi untuk meningkatkan akurasi optimal.
- Menguji dataset dengan data test *external (Internet)*.
- Proses dalam pengambilan dataset untuk lebih diperhatikan untuk mendapatkan akurasi yang optimal.
- Mencoba untuk menggunakan arsitektur yang lebih modern untuk kedepannya.

REFERENSI

- [1] D. D. Nugraha, "TRANSFORMASI BENTUK DAUN MONSTERA SEBAGAI MOTIF BATIK DALAM BUSANA KASUAL," *Comput. Human Behav.*, vol. 63, no. May, pp. 9–57, 2019, [Online]. Available: <http://dx.doi.org/10.1016/j.chb.2016.05.008>.
- [2] A. H. L. A. N. Gunawardena, K. Sault, P. Donnelly, J. S. Greenwood, and N. G. Dengler, "Programmed cell death and leaf morphogenesis in *Monstera obliqua* (Araceae)," *Planta*, vol. 221, no. 5, pp. 607–618, 2005, doi: 10.1007/s00425-005-1545-1.
- [3] E. P. N. A. Wijaya, H. Armanto, L. Zaman, and T. Informasi, "Klasifikasi akasara jawa dengan cnn," *J. Tek.*, vol. 13, no. 2, pp. 61–64, 2020.
- [4] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," *2018 IEEE 8th Annu. Comput. Commun. Work. Conf. CCWC 2018*, vol. 2018-Janua, pp. 169–175, 2018, doi: 10.1109/CCWC.2018.8301729.
- [5] I. M. Andrade, S. J. Mayo, D. Kirkup, and C. Van Den Berg, "Comparative morphology of populations of *Monstera Adans.* (Araceae) from natural forest fragments in Northeast Brazil using elliptic Fourier Analysis of leaf outlines," *Kew Bull.*, vol. 63, no. 2, pp. 193–211, 2008, doi: 10.1007/s12225-008-9032-z.
- [6] S. S. A.-N. Belal A.M. Ashqar, Bassem S. Abu-Nasser, "Plant Seedlings Classification Using Deep Learning," *Int. J. Acad. Inf. Syst. Res.*, vol. 46, no. 3, pp. 745–749, 2019.
- [7] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 1, no. 01, pp. 1–6, 2016, doi: 10.29100/jipi.v1i01.18.
- [8] K. H. Mahmud, Adiwijaya, and S. Al Faraby, "Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2127–2136, 2019.
- [9] E. N. Arrofiqoh and H. Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *Geomatika*, vol. 24, no. 2, p. 61, 2018, doi: 10.24895/jig.2018.24-2.810.
- [10] M. M. Taslim, K. Gunadi, and A. N. Tjondrowiguno, "Deteksi Rumus Matematika pada Halaman Dokumen Digital dengan Metode Convolutional Neural Network," *J. Infra*, vol. 7, no. 2, pp. 123–129, 2019, [Online]. Available: www.arxiv.org.
- [11] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th Int. Symp. Qual. Serv. IWQoS 2018*, pp. 1–2, 2019, doi: 10.1109/IWQoS.2018.8624183.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [13] P. Hridayami, I. K. G. D. Putra, and K. S. Wibawa, "Fish species recognition using VGG16 deep convolutional neural network," *J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 124–130, 2019, doi: 10.5626/JCSE.2019.13.3.124.
- [14] M. Rajnoha, R. Burget, and L. Povoda, "Image Background Noise Impact on Convolutional Neural Network Training," *Int. Congr. Ultra Mod. Telecommun. Control Syst. Work.*, vol. 2018-Novem, pp. 1–4, 2019, doi: 10.1109/ICUMT.2018.8631242.

LAMPIRAN
LAMPIRAN 1. Tabel hasil score testing Precision

skenario		Pericision				
		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dengan BG	adam 0.0001 + Dropout 0.1	86.33%	85.67%	88.00%	86.67%	86.67%
	adam 0.0001 + Dropout 0.3	84.67%	83.33%	83.33%	87.67%	82.00%
	adam 0.0001 + Dropout 0.5	78.67%	83.00%	82.33%	84.33%	81.67%
	adam 0.0003 + Dropout 0.1	85.00%	87.33%	88.33%	88.67%	81.67%
	adam 0.0003 + Dropout 0.3	85.33%	84.33%	90.33%	87.00%	82.00%
	adam 0.0003 + Dropout 0.5	84.33%	82.00%	83.33%	86.33%	84.67%
Tanpa BG	adam 0.0001 + Dropout 0.1	65.67%	69.67%	70.00%	71.00%	69.33%
	adam 0.0001 + Dropout 0.3	67.00%	74.33%	71.33%	68.67%	78.00%
	adam 0.0001 + Dropout 0.5	75.00%	72.00%	74.33%	73.33%	66.33%
	adam 0.0003 + Dropout 0.1	64.33%	77.00%	74.33%	67.00%	66.67%
	adam 0.0003 + Dropout 0.3	73.00%	69.33%	65.67%	68.33%	73.00%
	adam 0.0003 + Dropout 0.5	71.33%	76.00%	67.33%	63.33%	65.00%

LAMPIRAN 2. Tabel hasil score testing Recall

skenario		Recall				
		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dengan BG	adam 0.0001 + Dropout 0.1	84.00%	77.33%	86.33%	84.67%	84.67%
	adam 0.0001 + Dropout 0.3	74.67%	79.00%	73.67%	86.33%	77.00%
	adam 0.0001 + Dropout 0.5	72.33%	80.00%	80.00%	82.67%	75.33%
	adam 0.0003 + Dropout 0.1	81.00%	85.33%	87.33%	87.67%	77.33%
	adam 0.0003 + Dropout 0.3	83.00%	78.00%	89.67%	85.00%	77.67%
	adam 0.0003 + Dropout 0.5	81.67%	77.33%	81.00%	80.67%	74.00%
Tanpa BG	adam 0.0001 + Dropout 0.1	66.33%	70.67%	68.33%	70.67%	70.67%
	adam 0.0001 + Dropout 0.3	67.33%	74.33%	71.33%	69.67%	77.33%
	adam 0.0001 + Dropout 0.5	74.67%	71.33%	70.00%	72.00%	67.67%
	adam 0.0003 + Dropout 0.1	65.67%	72.33%	74.33%	65.33%	67.33%
	adam 0.0003 + Dropout 0.3	72.67%	70.33%	67.00%	69.33%	73.33%
	adam 0.0003 + Dropout 0.5	71.00%	63.00%	68.00%	66.67%	67.00%

LAMPIRAN 3. Tabel hasil score testing F1-Score

skenario		F1-Score				
		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dengan BG	adam 0.0001 + Dropout 0.1	84.00%	77.00%	86.33%	85.00%	85.00%
	adam 0.0001 + Dropout 0.3	74.33%	77.67%	72.67%	86.33%	76.33%
	adam 0.0001 + Dropout 0.5	71.33%	79.33%	78.00%	82.67%	74.67%
	adam 0.0003 + Dropout 0.1	81.00%	85.00%	91.00%	87.67%	77.00%
	adam 0.0003 + Dropout 0.3	83.00%	77.67%	90.00%	85.00%	76.67%
	adam 0.0003 + Dropout 0.5	81.33%	76.33%	80.67%	80.67%	74.33%
Tanpa BG	adam 0.0001 + Dropout 0.1	61.67%	68.33%	65.00%	70.33%	69.00%
	adam 0.0001 + Dropout 0.3	62.33%	74.00%	68.00%	65.67%	77.00%
	adam 0.0001 + Dropout 0.5	74.67%	67.33%	66.00%	72.00%	62.00%
	adam 0.0003 + Dropout 0.1	59.67%	72.33%	74.33%	62.00%	61.67%
	adam 0.0003 + Dropout 0.3	71.33%	67.67%	61.33%	63.67%	71.67%
	adam 0.0003 + Dropout 0.5	69.33%	52.00%	66.33%	59.33%	61.00%

LAMPIRAN 4. Tabel hasil score testing Accuracy

skenario		Akurasi				
		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dengan BG	adam 0.0001 + Dropout 0.1	84.12%	76.98%	86.50%	84.92%	84.92%
	adam 0.0001 + Dropout 0.3	74.60%	78.96%	73.41%	86.50%	77.38%
	adam 0.0001 + Dropout 0.5	72.22%	79.76%	79.76%	82.54%	75.39%
	adam 0.0003 + Dropout 0.1	80.95%	85.31%	87.30%	87.69%	77.38%
	adam 0.0003 + Dropout 0.3	83.33%	78.17%	89.68%	84.92%	77.38%
	adam 0.0003 + Dropout 0.5	81.74%	77.38%	80.95%	80.55%	73.81%
Tanpa BG	adam 0.0001 + Dropout 0.1	66.27%	70.63%	68.65%	70.63%	70.63%
	adam 0.0001 + Dropout 0.3	67.46%	74.60%	71.42%	69.84%	77.38%
	adam 0.0001 + Dropout 0.5	74.60%	71.42%	69.84%	71.82%	67.85%
	adam 0.0003 + Dropout 0.1	65.87%	72.22%	74.20%	65.47%	67.46%
	adam 0.0003 + Dropout 0.3	72.61%	70.63%	67.06%	69.44%	73.41%
	adam 0.0003 + Dropout 0.5	70.63%	63.09%	68.25%	66.66%	66.66%

LAMPIRAN 5. Tabel hasil Score Accuracy

skenario		Fold 1			Fold 2			Fold 3			Fold 4			Fold 5		
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test
Dengan BG	adam 0.0001 + Dropout 0.1	100.00%	100.00%	84.12%	99.90%	99.60%	76.98%	100.00%	99.19%	86.50%	100.00%	100.00%	84.92%	99.80%	97.98%	84.92%
	adam 0.0001 + Dropout 0.3	98.48%	100.00%	74.60%	98.18%	99.19%	78.96%	98.58%	98.79%	73.41%	99.49%	100.00%	86.50%	98.58%	97.98%	77.38%
	adam 0.0001 + Dropout 0.5	92.51%	99.60%	72.22%	92.52%	98.38%	79.76%	91.71%	98.79%	79.76%	92.92%	99.19%	82.54%	92.62%	97.57%	75.39%
	adam 0.0003 + Dropout 0.1	100.00%	100.00%	80.95%	100.00%	99.19%	85.31%	99.70%	99.60%	87.30%	100.00%	100.00%	87.69%	99.90%	97.57%	77.38%
	adam 0.0003 + Dropout 0.3	98.99%	100.00%	83.33%	99.80%	98.79%	78.17%	99.29%	99.19%	89.68%	98.99%	100.00%	84.92%	99.70%	97.98%	77.38%
	adam 0.0003 + Dropout 0.5	95.24%	100.00%	81.74%	90.09%	98.38%	77.38%	94.24%	98.79%	80.95%	92.21%	99.60%	80.55%	94.84%	97.98%	73.81%
Tanpa BG	adam 0.0001 + Dropout 0.1	95.04%	94.76%	66.27%	95.45%	93.12%	70.63%	94.84%	88.66%	68.65%	95.65%	89.07%	70.63%	94.24%	95.95%	70.63%
	adam 0.0001 + Dropout 0.3	86.54%	92.74%	67.46%	83.42%	90.69%	74.60%	89.08%	90.28%	71.42%	87.77%	87.85%	69.84%	84.93%	95.14%	77.38%
	adam 0.0001 + Dropout 0.5	71.86%	85.89%	74.60%	71.89%	88.66%	71.42%	71.08%	84.21%	69.84%	69.46%	81.78%	71.82%	69.97%	91.50%	67.85%
	adam 0.0003 + Dropout 0.1	95.95%	93.55%	65.87%	96.66%	93.93%	72.22%	95.75%	96.36%	74.20%	94.94%	92.31%	65.47%	96.97%	95.55%	67.46%
	adam 0.0003 + Dropout 0.3	90.49%	91.13%	72.61%	90.50%	91.09%	70.63%	89.69%	91.90%	67.06%	89.08%	91.50%	69.44%	86.96%	95.55%	73.41%
	adam 0.0003 + Dropout 0.5	71.05%	93.15%	70.63%	66.23%	79.35%	63.09%	75.63%	85.02%	68.25%	69.67%	82.59%	66.66%	67.54%	91.90%	66.66%

LAMPIRAN 6. Dokumentasi Penelitian pengambilan *dataset*

