



**IDENTIFIKASI PENYAKIT TANAMAN TOMAT
MENGGUNAKAN ALGORITMA *CONVOLUTIONAL
NEURAL NETWORK* DAN PENDEKATAN *TRANSFER
LEARNING***

Skripsi

**diajukan sebagai salah satu persyaratan untuk memperoleh gelar
Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan
Komputer**

Oleh
Maulana Muhammad Fathul Alim
NIM.5302416053

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2020**

PERSETUJUAN PEMBIMBING

Nama : Maulana Muhammad Fathul Alim
NIM : 5302416053
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer
Judul : Identifikasi Penyakit Tanaman Tomat Menggunakan
Algoritma *Convolutional Neural Network* dan Pendekatan
Transfer Learning

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer Fakultas Teknik Universitas Negeri Semarang.

Semarang, 10 Agustus 2020

Pembimbing



Dr. Ir. Subiyanto, S.T., M.T.

NIP. 197411232005011001

PENGESAHAN

Skripsi dengan judul Identifikasi Penyakit Tanaman Tomat Menggunakan Algoritma *Convolutional Neural Network* dan Pendekatan *Transfer Learning* telah dipertahankan di depan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES pada 10 Agustus 2020.

Oleh

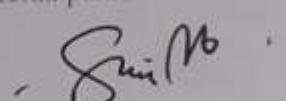
Nama : Maulana Muhammad Fathul Alim

NIM : 5302416053

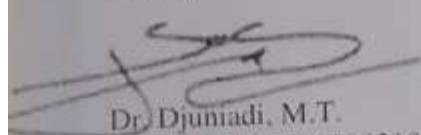
Program Studi : S1-Pendidikan Teknik Informatika dan Komputer

Panitia:

Ketua panitia

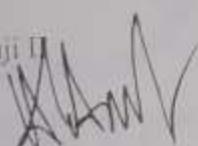

Ir. Ulfah Mediati Arief, M.T., IPM
NIP. 196605051997022001

Pengaji I

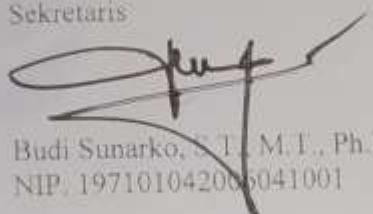

Dr. Djumiadi, M.T.

NIP. 196306281990021001

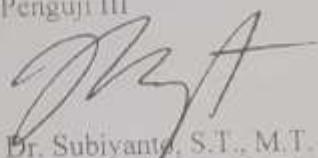
Pengaji II


Alfa Faridh Suni, S.T., M.T.
NIP. 198210192014041001

Sekretaris


Budi Sunarko, S.T., M.T., Ph.D.
NIP. 197101042006041001

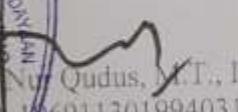
Pengaji III


Dr. Subiyanto, S.T., M.T.
NIP. 197411232005011001

Mengetahui

Fakultas Teknik UNNES




Dr. Nur Qudus, M.T., IPM
NIP. 196911301994031001

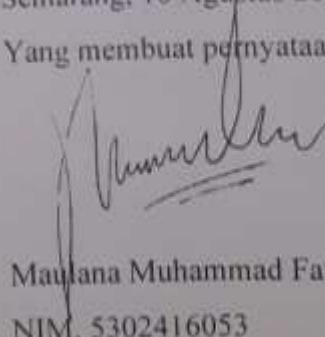
PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/atau doctor) baik di Universitas Negeri Semarang maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing dan masukan tim penguji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebut nama pengarang dan dicantumkan dalam Daftar Pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang, 10 Agustus 2020

Yang membuat pernyataan



Maulana Muhammad Fathul Alim

NIM. 5302416053

MOTTO DAN PERSEMBAHAN

“Dialah Yang telah menurunkan air hujan dari langit untuk kalian, sebagianya menjadi minuman dan sebagianya (menyuburkan) tumbuh-tumbuhan, yang pada (tempat tumbuhnya) kalian menggembalakan ternak kalian”

(Q.S. An-Nahl:10)

Persembahan:

- Keluarga tercinta yang menjadi motivasi, penyemangat, pemberi dukungan dalam menyelesaikan skripsi ini.
- Pengasuh Pondok Pesantren Durrotu Ahlissunnah Waljamaah Banaran Semarang selaku orang tua dan guru spiritual.
- Dosen pembimbing mata kuliah skripsi, Dr. Subiyanto, S.T., M.T. atas arahan dan bimbingannya selama penelitian sehingga skripsi ini dapat terselesaikan.
- Teman-teman satu bimbingan dan tim *Unnes Electrical Engineering Student Research Group* (UEESRG) yang telah banyak membantu dalam penyelesaian skripsi ini.
- Teman-teman Jurusan Teknik Elektro angkatan tahun 2016 yang telah berjuang bersama dalam perkuliahan.
- Teman-teman Pondok Pesantren Durrotu Ahlissunnah Waljamaah yang banyak memberikan dukungan dalam menyelesaikan skripsi.

RINGKASAN

Maulana Muhammad Fathul Alim, 2020, Identifikasi Penyakit Tanaman Tomat Menggunakan Algoritma *Convolutional Neural Network* dan Pendekatan *Transfer Learning*, Dr. Subiyanto, S.T.,M.T., Pendidikan Teknik Informatika dan Komputer.

Penyakit tanaman menjadi ancaman yang menghambat produksi tomat global. Pengendalian penyakit tanaman tomat dengan sistem pengenalan otomatis sangat diperlukan untuk mencegah kegagalan panen. Terinspirasi oleh keberhasilan *deep learning* dalam melakukan identifikasi berdasarkan citra digital, penerapan *deep learning* juga akan memberikan dampak yang signifikan dalam mengidentifikasi penyakit tanaman tomat.

Penelitian ini mengadopsi algoritma *Convolutional Neural Network* (CNN) untuk melakukan identifikasi penyakit tanaman tomat berdasarkan citra digital. Beberapa model CNN seperti VVG, ResNet, dan DenseNet dikomparasikan performanya dalam melakukan identifikasi penyakit tanaman tomat secara otomatis. Eksperimen dilakukan dengan menggunakan dataset PlantVillage yang terdiri dari 22.930 gambar daun tomat yang terbagi menjadi 10 kelas.

Metode yang diusulkan mengeksplorasi fitur pada citra dengan memecahnya menjadi beberapa bagian kecil dan bertumpuk kemudian menyatukannya kembali untuk diolah dengan fungsi aktivasi. Evaluasi performa dilakukan dengan menghitung *accuracy*, *precision*, *recall*, *f1-score*, dan AUC menggunakan metode *confusion matrix*. Model CNN terbaik dari komparasi adalah ResNet-50 dengan 96% *accuracy*, 97% *precision*, 96% *recall*, 97% *f1-score*, dan 97,9% AUC. ResNet-50 mampu melakukan prediksi benar sebanyak 4402 terhadap 4585 data uji. Hasil eksperimen menunjukkan bahwa model CNN dapat menjadi alat yang berguna dalam mengidentifikasi dan mengklasifikasikan penyakit tanaman tomat.

Kata kunci: Penyakit tanaman tomat, *Convolutional Neural Network*, *Transfer Learning*

PRAKATA

Puji syukur penulis sampaikan ke hadirat Allah SWT karena atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Identifikasi Penyakit Tanaman Tomat Menggunakan Algoritma *Convolutional Neural Network* dan Pendekatan *Transfer Learning*”. Skripsi ini merupakan tugas akhir yang diajukan untuk memenuhi syarat dalam memperoleh gelar Sarjana Pendidikan pada Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang. Penulis menyadari bahwa penulisan ini tidak akan terwujud tanpa adanya bantuan dan dorongan dari berbagai pihak. Oleh karena itu peneliti menyampaikan ucapan terimakasih kepada:

1. Dr. Subiyanto, S.T.,M.T., selaku dosen pembimbing yang telah memberikan bimbingan dan arahan dalam penyusunan skripsi ini.
2. Budi Sunarko, S.T., M.T., Ph.D., Koordinator Program Studi PTIK Unnes.
3. Ir. Ulfah Mediaty Arief, M.T., Ketua Jurusan Teknik Elektro Unnes.
4. Dr. Nur Qudus, M.T., Dekan Fakultas Teknik Unnes.
5. Prof. Dr. Fathur Rokhman, M.Hum. , Rektor Universitas Negeri Semarang atas kesempatan yang diberikan kepada penulis untuk menempuh studi di Universitas Negeri Semarang.
6. Segenap dosen jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang yang telah memberikan banyak ilmu pengetahuan.
7. Orang tua dan keluarga penulis yang telah memberikan dukungan dan doa.
8. Tim UEESRG yang memberikan banyak saran masukan dan bantuan dalam berbagai hal berkaitan dengan penelitian.
9. Rekan-rekan jurusan Teknik Elektro Unnes Angkatan 2016.
10. Keluarga Pondok Pesantren Durrotu Ahlissunnah Waljamaah.
11. Berbagai pihak yang telah memberi bantuan dalam menyelesaikan skripsi ini.

Penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak khususnya bagi penulis sendiri dan masyarakat serta pembaca pada umumnya.

Semarang, 10 Agustus 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERSETUJUAN PEMBIMBING	ii
PENGESAHAN.....	iii
PERNYATAAN KEASLIAN.....	iv
MOTTO DAN PERSEMBAHAN.....	v
RINGKASAN	vi
PRAKATA	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah	3
1.3. Rumusan Masalah	4
1.4. Batasan Masalah	4
1.5. Tujuan Penelitian.....	5
1.6. Manfaat Penelitian.....	5
BAB II	7
2.1. Kajian Pustaka.....	7
2.2. <i>Computer Vision</i>	12
2.3. Citra Digital.....	13
2.4. Identifikasi Penyakit Tomat	16
2.4.1. Bacterial Spot	16
2.4.2. <i>Early Blight</i> (Pembusukan Dini)	16
2.4.3. <i>Late Blight</i> (Busuk Daun)	17
2.4.4. <i>Leaf Mold</i> (Bercak Daun karena Jamur).....	17
2.4.5. <i>Septoria Leaf Spot</i>	18
2.4.6. Spider Mites Two Spider Mite	18
2.4.7. Target Spot	19

2.4.8. Tomato Mosaic Virus.....	19
2.4.9. Tomato Yellow Leaf Curl Virus	19
2.5. <i>Artificial Neural Network</i>	21
2.6. <i>Convolutional Neural Network (CNN)</i>	29
2.6.1. <i>Convolutional Layer (Conv. Layer)</i>	31
2.6.2. <i>Pooling layer</i>	32
2.6.3. <i>Fully-Connected Layer</i>	33
2.7. <i>Transfer Learning</i>	34
2.7.1. VGG.....	35
2.7.2. ResNet.....	36
2.7.3. DenseNet.....	37
BAB III.....	39
3.1. Waktu dan Tempat Penelitian.....	39
3.2. Alat dan Bahan Penelitian	39
3.3. Desain Penelitian.....	41
3.4. <i>Modelling CNN</i>	42
3.4.1. Instalasi Model CNN.....	43
3.4.2. Persiapan Dataset.....	46
3.4.3. <i>Training Model</i>	48
3.4.4. <i>Testing Model</i>	50
3.5. Teknik Pengumpulan Data	51
3.6. Teknik Analisis Data.....	53
BAB IV	56
4.1. Performa Pelatihan dan Validasi	57
4.2. Performa Pengujian Model.....	62
4.3. Komparasi Performa Model	70
4.4. Desain Aplikasi.....	72
BAB V.....	75
5.1. Simpulan	75
5.2. Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN.....	83

Lampiran 1. Kode Program (Konfigurasi Bahasa Pemrograman Python)	83
Lampiran 2. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model VGG-16)	84
Lampiran 3. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model VGG-19)	86
Lampiran 4. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model ResNet-50)	88
Lampiran 5. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model ResNet-101)	90
Lampiran 6. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model DenseNet-121)	92
Lampiran 7. Kode Program (Persiapan Data, Pelatihan, Pengujian, dan Evaluasi Model DenseNet-169)	94
Lampiran 8. Kode Program (Plotting Grafik Pelatihan)	96
Lampiran 9. Blok Diagram Sistem	98
Lampiran 10. Berkas Administrasi	100

DAFTAR TABEL

Table 3.1 Jadwal Penelitian.....	39
Table 3.2 Alat dan Bahan Penelitain	40
Table 3.3 Dataset Penyakit Tomat.....	40
Tabel 3.4 Karakteristik dari Setiap Model CNN	46
Tabel 3.5 Parameter Pelatihan Model CNN	50
Table 4.1 <i>Performance Metric</i> dari Model CNN	67
Table 4.2 Komparasi Hasil dengan Penelitian Terdahulu	73

DAFTAR GAMBAR

Gambar 2.1 Koordinat Citra Digital.....	14
Gambar 2.2 Representasi Citra dalam Bentuk Matriks	14
Gambar 2.3 Penyakit Tanaman Tomat	20
Gambar 2.4 Jaringan Syaraf Manusia.....	21
Gambar 2.5 Jaringan Syaraf Tiruan (JST)	22
Gambar 2.6 Fungsi Aktivasi ReLU	24
Gambar 2.7 <i>Multi-Layer Perceptron</i> (MLP).....	26
Gambar 2.8 <i>Konsep Machine Learning</i>	27
Gambar 2.9 Ilustrasi <i>Convolutional Neural Network</i> (CNN).....	30
Gambar 2.10 Arsitektur <i>Convolutional Neural Network</i> (CNN).....	31
Gambar 2.11 Ilustrasi Proses Konvousi Citra.....	32
Gambar 2.12 <i>Max-Pooling</i>.....	33
Gambar 2.13 Arsitektur Jaringan VGG.....	36
Gambar 2.14 Arsitektur ResNet Dibandingkan dengan VGG dan <i>Feed Forward Neural Network</i>.....	37
Gambar 2.15 Arsitektur <i>DenseNet</i> dengan Model <i>DenseBlocks</i>.....	38
Gambar 3.1 <i>Flowchart</i> Penelitian	40
Gambar 3.2 <i>Flowchart</i> Pelatihan dan Pengujian CNN	43
Gambar 3.3 Rangkuman Informasi Model VGG-16.....	46
Gambar 4.1. <i>Training Accuracy</i> pada Model yang Dilatih.....	59
Gambar 4.2 <i>Training Loss</i> pada Model yang Dilatih.....	60
Gambar 4.3 <i>Validation Accuracy</i> pada Model yang Dilatih.....	61
Gambar 4.4 <i>Validation Loss</i> pada Model yang Dilatih.....	62
Gambar 4.5 <i>Confusion Metric</i> dari Setiap Model CNN.....	66
Gambar 4.6 Diagram Batang <i>Performance Metrics</i> dari Tiap Model CNN ...	68
Gambar 4.7 Waktu Pelatihan Model CNN.....	71

Gambar 4.8 Tampilan Awal Aplikasi	72
Gambar 4.9 Tampilan Aplikasi Ketika Model dan Gambar Telah Dipilih ...	73
Gambar 4.11 Tampilan Ketika Aplikasi Mendeteksi Penyakit Pada Daun yang Dipilih.....	74

BAB I

PENDAHULUAN

1.1. Latar Belakang

Tingginya permintaan pasar terhadap buah tomat, baik tomat segar maupun tomat olahan membutuhkan produktifitas yang tinggi pada sektor pertanian tomat (Adhikari, Oh and Panthee, 2017). Terlepas dari tanaman tomat yang dapat tumbuh di berbagai iklim dunia (Adhikari, Oh and Panthee, 2017), penyakit yang menyerang tanaman tomat merupakan ancaman yang dapat menghambat produksi tomat global dan berefek pada kerugian pasca panen yang signifikan (Ali *et al.*, 2019a). Sangat penting bagi petani untuk mendeteksi penyakit pada tanaman lebih awal untuk mengontrol penyebaran penyakit (Martinelli *et al.*, 2015). Pemberian tindakan sangat bergantung pada pengenalan penyakit yang menyerang tanaman. Metode tradisional untuk pengenalan penyakit pada tanaman, berdasar pada interpretasi visual dari gejala yang muncul dan memerlukan analisis laboratorium yang mana metode tersebut membutuhkan tenaga ahli di bidang plantologi dan waktu yang memadai untuk menyelesaikan diagnosis (Kumlachew Alemu, 2015).

Dalam ilmu Biologi, daun digunakan sebagai indikator status kesehatan pada tanaman melalui pengamatan visual warna daun yang berkaitan langsung terhadap kandungan klorofil didalamnya (Ali *et al.*, 2019a). *Computer Vision* memegang peran dalam dunia teknologi informasi digital sebagai metode identifikasi yang menggunakan data gambar visual (Hameed, Chai and Rassau,

2018). *Computer Vision* bertanggung jawab untuk penerapan metode yang memungkinkan komputer memahami konten pada gambar, yang mana interpretasi ini membutuhkan ekstraksi karakteristik tertentu dari suatu gambar (Gomes and Leta, 2012).

Sistem cerdas berdasarkan *Computer Vision* sudah selayaknya menjadi bagian dari manajemen produksi pertanian untuk meningkatkan produktifitas (Tian *et al.*, 2019). Pengenalan penyakit tanaman dengan *computer vision* merupakan salah satu bentuk aplikasinya. Seiring perkembangan teknologi, banyak penelitian mengembangkan sistem cerdas dengan menggunakan jaringan syaraf tiruan yang meniru cara kerja otak manusia. Algoritma *Convolutional Neural Network* yang merupakan bagian dari *deep learning*, melatih jaringan yang berupa lapisan-lapisan konvolusi untuk mengestraksi fitur penting pada citra (Ferentinos, 2018).

Beberapa penelitian terdahulu menggunakan pendekatan *scratch* dalam melatih algoritma CNN. Metode *scratch* yang merupakan metode tunggal yang mana mengharuskan peneliti untuk mengatur jenis layer dan parameter yang digunakan. Berbeda dengan *scratch*, pendekatan *Transfer Learning* bukan merupakan metode tunggal, namun beberapa metode serupa yang dikomparasikan satu sama lain untuk mendapatkan akurasi yang maksimal atau yang biasa dikenal sebagai *fine-tuning* (Lee *et al.*, 2020), dan terbukti bahwa pelatihan *Convolutional Neural Network* dengan pendekatan *fine-tunning* atau *transfer learning* lebih baik dari pada pelatihan model pendekatan *scratch* (Mohanty, Hughes and Salathé, 2016). Namun belum banyak juga peneliti yang menggunakan pendekatan *transfer learning* untuk malakukan pelatihan CNN.

1.2. Identifikasi Masalah

Pengendalian penyakit pada tanaman sangat perlu untuk dilakukan guna mencegah gagal panen. Metode *computer vision* banyak digunakan oleh peneliti dalam melakukan identifikasi penyakit pada tanaman. Di masa lalu, berbagai metode klasifikasi otomatis telah digunakan untuk mengklasifikasikan data penginderaan jarak jauh dan pengamatan tanaman. Motode pembelajaran mesin seperti SVM, *Fuzzy Logic*, *Decision Tree*, *K-means*, KNN, dan Jaringan Syaraf Tiruan telah diterapkan dalam penelitian pertanian.

Penelitian mengenai identifikasi penyakit tanaman dilakukan oleh (Rumpf *et al.*, 2010) pada tahun 2010 dengan menggunakan algoritma SVM dan mendapatkan nilai akurasi sebesar 86%. Algoritma ini mampu melakukan klasifikasi dengan tiga kelas penyakit yang berbeda dan dengan pengenalan stadium kaparahan penyakit algoritma SVM mendapatkan akurasi klasifikasi antara 65% sampai 90%. Seiring perkembangan teknologi, banyak dikenalkan algoritma-agoritma baru dengan kemampuan yang lebih baik. Metode *neural network* meniru cara kerja dari otak manusia sehingga bisa mengenali suatu objek dengan memperhatikan fitur unik dari setiap objek.

Algoritma *Convolutional Neural Network* memecah citra menjadi piksel-piksel kecil yang bertumpuk dan menyatukannya kembali menjadi sebuah citra dengan fitur yang lebih jelas. (Rangarajan and Purushothaman, 2018) melakukan klasifikasi penyakit pada daun tomat menggunakan algoritma *Convolutional Neural Network* dengan pendekatan *transfer learning*. Arsitektur jaringan yang digunakan adalah *AlexNet* dan *VGG16*. Algoritma CNN dengan model jaringan

AlexNet dengan kinerja model yang dievaluasi memberikan hasil akurasi yang lebih tinggi mencapai 97,49%.

1.3. Rumusan Masalah

Metode *deep learning* dengan algoritma *Convolutional Neural Network* melakukan konvolusi dengan memecah citra menjadi piksel-piksel kecil yang bertumpuk. Hasil pemecahan kemudian disatukan kembali dengan menerapkan proses konvolusi dengan lapisan *filter*. Hal ini diakukan untuk mendapatkan fitur citra yang lebih spesifik, dan proses diulang beberapa kali dengan menerapkan fungsi aktivasi. Pendekatan *scratch* mengharuskan untuk membentuk susunan *layer* secara manual, metode ini tidak akan menghasilkan suatu model yang baik apabila dilakukan dengan sumber daya yang terbatas. Pelatihan *Convolutional Neural Network* dengan pendekatan *scratch* berkemungkinan menimbulkan *overfitting* dan masalah konvergensi, oleh karenanya dikembangkanlah pendekatan *transfer learning*. Pendekatan ini melakukan *fine-tuning* terhadap *pre-trained model* dalam melakukan pelatihan untuk mencegah *overfitting*.

1.4. Batasan Masalah

Agar pembahasan dalam penelitian ini lebih fokus, maka diterapkan batasan-batasan sebagai berikut.

1. Pada penelitian ini menggunakan algoritma CNN dengan penerapan *transfer learning* sebagai subjek yang dikaji.

2. Algoritma CNN digunakan untuk identifikasi penyakit pada daun tomat.
3. System hanya dirancang untuk mengenali penyakit pada daun tomat dengan memiliki 9 jenis penyakit dan kondisi sehat.
4. Data yang digunakan dalam penelitian ini adalah data sekunder dataset *PlantVillages* yang berisi citra daun penyakit tanaman tomat.
5. Hasil pengelompokan hanya didasarkan pada sistem klasifikasi dengan komputer tanpa bantuan dari ahli penyakit tanaman.
6. Uji akurasi dilakukan dengan menggunakan *confusion matriks* untuk menghitung tingkat akurasi yang didapatkan oleh masing-masing model.

1.5. Tujuan Penelitian

Tujuan penelitian ini adalah untuk mendapatkan model CNN dengan performa terbaik dalam identifikasi penyakit tanaman tomat menggunakan algoritma CNN dan pedekatan *transfer learning*.

1.6. Manfaat Penelitian

Manfaat dari penelitian ini dikategorikan berdasarkan penerapannya. Ada dua jenis kemanfaatan yang diharapkan, yakni secara teoritis untuk kepentingan penelitian dan penerapan praktis, diantaranya sebagai berikut

a. Manfaat Teoritis

Hasil penelitian ini diharapkan bisa menyumbangkan pemikiran dalam memperkaya wawasan mengenai konsep penggunaan algoritma CNN dalam

mengaplikasikannya pada bidang pertanian, terkhusus pada pengenalan dan klasifikasi jenis penyakit tomat melalui citra digital daun tomat.

b. Manfaat Praktis

Hasil penelitian ini diharapkan dapat menyumbangkan pemikiran dalam pemecahan salah satu masalah dibidang pertanian yakni mengenai tahap awal identifikasi penyakit yang menyerang tanaman tomat. Selanjutnya hasil penelitian ini diharapkan bisa menjadi acuan bagi penyusunan program pengenalan penyakit tomat oleh lembaga yang terkait.

BAB II

KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1. Kajian Pustaka

Sebuah penelitian yang berjudul *Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance* (Rumpf *et al.*, 2010) bertujuan untuk pendekripsi dini penyakit bercak daun *Cercospora*, karat daun dan embun tepung pada daun bit gula dengan menggunakan pencitraan *hyperspectral*. Peneliti menggunakan algoritma SVM untuk melakukan pekerjaan klasifikasi. Hasil yang didapatkan dari penelitian tersebut adalah algoritma SVM mampu melakukan klasifikasi pada tiga jenis penyakit dan kondisi sehat mencapai diatas 86%. Dan dengan kondisi jenis dan stadium penyakit, akurasi klasifikasi antara 65% sampai 90%.

Dalam penelitian lainnya yang berjudul *Using Deep Learning for Image-Based Plant Disease Detection* (Mohanty, Hughes and Salathé, 2016) juga melakukan identifikasi jenis penyakit yang menyerang pada tanaman. Dataset yang digunakan adalah 54.306 gambar daun tanaman yang sakit dan sehat yang dikumpulkan di bawah kondisi yang terkendali. Algoritma CNN digunakan untuk mengidentifikasi 14 jenis tanaman dan 26 klasifikasi penyakit. Model *GoogleNet* yang terlatih mencapai akurasi 99,35%.

Dalam penelitian lainnya yang berjudul *Deep Learning for Tomato Diseases: Classification and Symptoms Visualization* (Brahimi, Boukhalfa and

Moussaoui, 2017) menggunakan algoritma CNN untuk mengidentifikasi penyakit pada daun tomat. Penelitian tersebut menggunakan dataset daun tomat dengan jumlah 14.828 gambar daun tomat yang terinfeksi dengan sembilan klasifikasi penyakit dan sehat. Hasil dari penelitian ini memperoleh tingkat akurasi 99,18% dengan menggunakan arsitektur *GoogleNet* yang dibandingkan dengan *AlexNet*.

Sebuah penelitian berjudul *Identification of rice diseases using deep convolutional neural networks* (Lu *et al.*, 2017) bertujuan untuk mengidentifikasi penyakit pada tanaman padi menggunakan *Deep Learning* dengan algortma CNN. Penelitian dilakukan dengan menggunakan dataset 500 gambar alami dari daun dan batang padi yang sakit dan sehat yang diambil dari sawah percobaan, CNN dilatih untuk mengidentifikasi 10 jenis penyakit padi umum. Model yang diusulkan dalam penelitian ini mencapai tingkat akurasi 95,48% dengan menggunakan model arsitektur jaringan *AlexNet*.

Penelitian lain yang berjudul *Potato Disease Classification Using Convolution Neural Networks* (Oppenheim and Shani, 2017) juga melakukan identifikasi jenis penyakit pada kentang dengan menggunakan algoritma CNN. Penelitian tersebut menggunakan dataset berjumlah 400 gambar kentang yang terkontaminasi dengan berbagai bentuk, dan ukuran yang diperoleh di bawah kondisi pencahayaan normal yang terkendali. Hasil penelitian yang didapatkan mencapai tingkat akurasi 96% menguunakan asitektur jaringan VGG dengan sebaran data untuk latih 90% dan data uji 10%.

Suatu penelitian berjudul *Identification of plant disease infection using soft-computing: Application to modern botany* (Kiani and Mamedov, 2018) bertujuan untuk mengenali penyakit yang menyerang pada tanaman dengan menggunakan citra daun. Penelitian dilakukan dengan pengambilan data di tempat pertanian terbuka dengan kondisi pencahayaan siang hari. Penelitian ini dilakukan menggunakan algoritma *classifier fuzzy-base* dan memberikan hasil perkiraan untuk pengambilan keputusan pada kondisi daun itu sehat atau tidak. Hasil akurasi yang diberikan yakni 96% untuk efisiensi model tersegmentasi dan 93% untuk tersegmentasi menggunakan perkiraan klasifikasi instan manusia sebagai tolok ukur, ini berarti algoritma *fuzzy-base* dalam melakukan pengenalan penyakit pada tanaman memiliki akurasi lebih tinggi daripada manusia.

Dalam penelitian berjudul *A comparative study of fine-tuning deep learning models for plant disease identification* (Chebet et al., 2018) melakukan klasifikasi penyakit tanaman berbasis gambar. Algoritma yang dipakai yakni CNN dengan menerapkan beberapa arsitektur meliputi VGG16, *Inception-V4*, *ResNet* dengan 50, 101 dan 152 lapisan dan *DenseNets* dengan 121 lapisan. Data yang digunakan untuk percobaan adalah 38 kelas penyakit yang beserta kondisi sehat dari 14 tanaman yang berbeda. Dalam percobaan menggunakan algoritma CNN ini, arsitektur jaringan *DenseNets* memiliki kecenderungan untuk secara konsisten meningkatkan ketidakakuratan tanpa tanda-tanda *overfitting* dan penurunan kinerja. Selain itu, *DenseNets* membutuhkan jumlah parameter yang jauh lebih sedikit dan waktu komputasi yang masuk akal untuk mencapai kinerja canggih. Ini mencapai skor akurasi pengujian 99,75% untuk mengalahkan arsitektur lainnya.

Suatu penelitian berjudul *Tomato crop disease classification using pre-trained deep learning algorithm* (Rangarajan and Purushothaman, 2018) melakukan klasifikasi penyakit pada daun tomat menggunakan algoritma CNN dengan arsitektur jaringan *AlexNet* dan *VGG16*. Data yang digunakan berasal dari *PlantVillage* dengan enam kelas penyakit dan kelas sehat yang berjumlah 13.262 gambar daun tomat. Hasil penelitian ini memberikan akurasi sebesar 97,29% untuk *VGG16* dan 97,49% untuk *AlexNet*. Kinerja model telah dievaluasi dengan memodifikasi jumlah gambar, mengatur berbagai ukuran *mini-batch* dan memvariasikan berat dan tingkat pembelajaran bias.

Penelitian lain yang berjudul *Deep learning models for plant disease detection and diagnosis* (Ferentinos, 2018) juga melakukan identifikasi penyakit pada tanaman dengan menggunakan algoritma CNN dengan pendekatan *transfer learning*. Penelitian tersebut menggunakan dataset 87.848 gambar, berisi 25 tanaman berbeda. Penelitian ini menggunakan lima jenis jaringan CNN, yakni *AlexNet*, *AlexNetOWTBn*, *GoogLeNet*, *Overfeat*, dan *VGG*. Hasil yang didapatkan mencapai tingkat akurasi 99,48% menggunakan arsitektur jaringan *VGG*.

Dalam penelitian yang berjudul *Depthwise separable convolution architectures for plant disease classification* (Kc et al., 2019) melakukan klasifikasi penyakit tanaman berdasarkan gambar daun. Peneliti memperkenalkan algoritma CNN dengan model arsitektur konvolusi yang terpisah. Jaringan yang digunakan adalah *MobileNet* yang dimodifikasi dengan dibandingkan pada jaringan *AlexNet* dan *VGG*. Proses *training* dan *testing* dilakukan pada *dataset PlantVillage* yang tersedia untuk umum terdiri dari 82.161 gambar yang mengandung 55 kelas

penyakit tanaman dan kondisi sehat. Arsitektur jaringan *Reduced MobileNet* mencapai akurasi klasifikasi 98,34% dengan parameter 29 kali lebih sedikit dibandingkan dengan *VGG* dan 6 kali lebih rendah dari *MobileNet*.

Dalam penelitian yang berjudul *New perspectives on plant disease characterization based on deep learning* (Lee *et al.*, 2020) bertujuan untuk melakukan identifikasi penyakit pada daun tomat menggunakan algoritma CNN dengan pendekatan pembelajaran *transfer learning*. Arsitektur yang dibandingkan yaitu *VGG16* (16 lapisan), *GoogLeNetBN* (34 lapisan) dan *InceptionV3* (48 lapisan) dengan menggunakan dataset penyakit daun tomat dari dataset *PlantVillage*. Hasil yang didapatkan yakni arsitektur jaringan *VGG16* memiliki kinerja lebih baik dari yang lainnya.

Penelitian dengan tujuan identifikasi penyakit tanaman sudah banyak dilakukan oleh para peneliti, namun perkembangan teknologi dari tahun ke tahun selalu menghadirkan metode dan algoritma-algoritma terbaru dengan kemampuan yang lebih baik. Dari semua penelitian yang ada, algoritma CNN dengan pendekatan *Transfer learning* mampu memberikan tingkat akurasi yang sangat baik. Pendekatan *transfer learning* memungkinkan untuk melakukan pelatihan model dengan lebih mudah dan cepat dan bahkan mengurangi kemungkinan *overfitting*. Namun dalam pelaksanaanya ada banyak sekali variasi model dari algoritma CNN. Model *VGG*, *ResNet* dan *DenseNet* terbukti dari beberapa penelitian mampu memberikan tingkat akurasi yang tinggi, sehingga dirasa perlu untuk mengkomparasikan ketiga model tersebut untuk melakukan identifikasi penyakit pada daun tomat.

2.2. *Computer Vision*

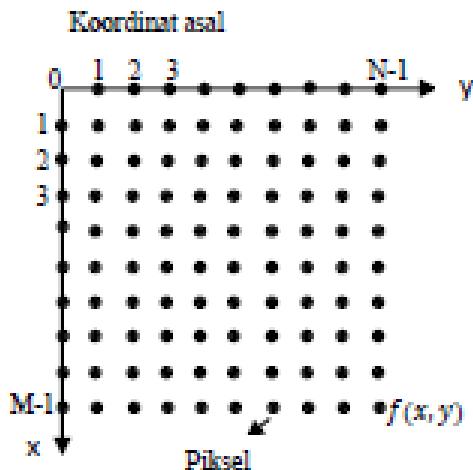
Computer Vision merupakan bidang analisis data dengan menggunakan gambar visual (Hameed, Chai and Rassau, 2018). *Computer vision* bertanggung jawab untuk studi dan penerapan metode yang memungkinkan komputer untuk memahami konten gambar. Interpretasi ini melibatkan ekstraksi karakteristik tertentu yang penting untuk tujuan tertentu (Gomes and Leta, 2012). Dalam industri pertanian, salah satu persyaratan yang penting adalah mengembangkan suatu sistem digital yang tidak merusak dalam melakukan penilaian otomatis. *Computer vision* merupakan metode *non-destructif* yang dapat digunakan untuk identifikasi dan telah terbukti dapat diterapkan di industri pertanian (Wan *et al.*, 2018) termasuk identifikasi dalam pengenalan dan penilaian penyakit yang menyerang tanaman.

Pada dasarnya, sistem *computer vision* terdiri dari tahapan akuisisi citra dan pemrosesan citra. Akuisisi citra merupakan tahapan paling awal yang perlu dilakukan. Akuisisi citra bertujuan untuk mendapatkan citra yang siap untuk di proses. Ada beberapa metode untuk mendapatkan gambar yang diperlukan, salah satunya adalah dengan menggunakan kamera digital. Lensa pada kamera digital menerima cahaya yang dipantulkan oleh objek lalu menyalurkannya pada sensor kamera untuk ditangkap menjadi sebuah gambar digital. Metode lain untuk melakukan akuisisi citra adalah dengan metode pencitraan *hyperspectral*, metode ini akan menghasilkan citra digital berbentuk 3 dimensi dari objek yang dituju.

2.3. Citra Digital

Citra adalah kombinasi antara titik, garis, bidang dan warna untuk menciptakan suatu imitasi dari suatu objek. Ditinjau dari sudut pandang matematis, citra merupakan fungsi *continue* dari intensitas cahaya pada bidang dwimatra. Sumber cahaya memancarkan cahaya pada objek, dan objek memantulkan sebagian cahaya tersebut sesuai dengan kapasitasnya. Cahaya yang dipantulkan ini akan ditangkap oleh alat optik seperti mata, kamera, dan lain-lain sehingga bayangan dari objek tersebut terekam. Citra terbagi dua yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Citra analog harus dikonversi menjadi citra digital terlebih dahulu agar dapat diproses di komputer (Pulung Nurtantio Andono, T.Sutojo, 2017)

Sebuah citra dapat didefinisikan sebagai fungsi $f(x, y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y dan nilai amplitudo f secara keseluruhan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar 2.1 menunjukkan posisi koordinat citra digital.



Gambar 2.1 Koordinat Citra Digital

Nilai suatu irisan antara baris dan kolom (pada posisi x, y) disebut dengan piksel. Sebuah citra digital adalah sebuah matriks (array dua dimensi) dari kumpulan piksel sebagai balok-balok bangunan dasar. Nilai setiap piksel sebanding dengan kecerahan titik yang sesuai dengan lokasinya. Matriks piksel dari sebuah gambar biasanya berbentuk persegi dan digambarkan sebagai piksel $N \times M$ dimana terdiri atas N kolom dan M baris. Secara sistematis, citra digital berbentuk matriks dinyatakan dalam persamaan 2.1.

$$f(x, y) = \begin{bmatrix} f(0, 0) & \cdots & f(0, N-1) \\ \vdots & \ddots & \vdots \\ f(M-1, 0) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (2.1)$$

Besar intensitas yang diterima oleh sensor di setiap titik (x, y) disimbolkan oleh $f(x, y)$. nilai dari $f(x, y)$ ditentukan oleh besar kecilnya intensitas cahaya yang diterima, yang artinya sebanding dengan besarnya energi yang dipancarkan oleh sumber pantulan cahaya, sehingga dalam bentuk matematisnya besar $f(x, y)$ dapat dilihat dalam persamaan 2.2.

$$0 < f(x, y) < \infty \quad (2.2)$$

Fungsi $f(x,y)$ dapat dipisahkan menjadi dua komponen, yaitu *illumination* dan *reflection*. *Illumination* merupakan besarnya cahaya yang dipancarkan sumber cahaya, disimbolkan $i(x,y)$. sedangkan *reflection* merupakan derajat kemampuan objek dalam memantulkan cahaya, disimbolkan $r(x,y)$. sehingga dalam bentuk matematisnya dapat dilihat pada persamaan 2.3.

$$f(x, y) = i(x, y) + r(x, y) \quad (2.3)$$

dan jika disubstitusikan dengan rumus sebelumnya akan menjadi persamaan 2.4.

$$0 < i(x, y) < \infty \text{ dan } 0 < r(x, y) < \infty \quad (2.4)$$

Sebuah citra sama seperti sebuah grid dengan masing-masing kotak persegi di dalam grid berisi satu warna atau piksel. Sebuah citra dengan resolusi 1024×768 merupakan sebuah grid yang berisi 1024 kolom dan 768 baris, yang mana berarti berisi $1024 \times 768 = 786432$ piksel.

Dalam pengolahan citra dikenal suatu metode *Pre-Processing* yang bertujuan untuk meningkatkan data citra dan menyoroti fitur atau wilayah yang akan dikaji, sehingga menghilangkan distorsi dan kebisingan tanpa menambahkan infomasi yang tidak perlu kedalam gambar (Gomes and Leta, 2012) Teknik *Pre-Processing* melibatkan Teknik untuk menyorot wilayah dan detail serta menghilangkan kebisingan yang dapat mengganggu analisis objek dan/atau wilayah yang dikaji. Dalam konteks ini ada beberapa macam teknik yang bisa digunakan, seperti penggunaan *Scale*, *Grayscale*, transformasi warna, serta *Thresholding* dan *Filtering*.

Proses segmentasi bisa didasarkan pada kesamaan warna setiap piksel dan piksel tetangganya, terkadang piksel yang serupa seperti dalam hal warna, bukan merupakan bagian dari objek atau fitur yang sama (Gomes and Leta, 2012). Setelah tahap ini, gambar harus siap untuk dilakukan proses *Feature Extraction*.

2.4. Identifikasi Penyakit Tomat

Ada beberapa jenis penyakit yang umum menyerang tanaman tomat, diantaranya *Bacterial Spot*, *Early Blight*, *Late Blight*, *Leaf Mold*, *Septoria Leaf Spot*, *Spider Mites*, *Two Spider Mite*, *Target Spot*, *Tomato Mosaic Virus*, *Tomato Yellow Leaf Curl*. Dari penyakit-penyakit ini, diantaranya disebabkan oleh jamur, virus, dan serangga, yang didukung oleh kondisi cuaca di lingkungan penanaman.

2.4.1. Bacterial Spot

Bacterial spot disebabkan oleh bakteri *Xanthomonas* yang tersebar luas dan merupakan penyakit bakteri yang merusak pada tanaman tomat (Osdaghi *et al.*, 2017). Infeksi biasanya menghasilkan lesi dan defoliasi pada daun dan buah, ini dapat menurunkan kualitas buah. Inveksi *bacterial spot* banyak terjadi di daerah yang hangat dan lembab dalam satu musim. Sayangnya, ketika suatu lahan pertanian terkena penyakit ini ini, akan sulit sekali untuk dilakukan pengendalian penyakit (Strayer-Scherer *et al.*, 2018).

2.4.2. Early Blight (Pembusukan Dini)

Early blight (EB) atau pembusukan daun dini disebabkan oleh jamur *Alaternaria Solani*. EB ditandai dengan cincin konsentris berwarna coklat hingga

hitam seperti bintik-bintik daun, yang menyatu secara lateral yang mengakibatkan busuk daun, *defoliasi* dan membuat gugur buah-buahan yang belum matang. Penyakit ini juga ditandai dengan adanya lesi yang cekung dan bintik-bintik hitam di dekat pangkal batang yang mengakibatkan pengerdilan pada batang (Shoaib, Awan and Khan, 2019). Daun yang terinfeksi menunjukkan bercak coklat hingga hitam (lesi) hingga diameter 1-2 cm dengan tepi gelap dan memiliki pola cincin konsentris (Sanoubar and Barbanti, 2017).

2.4.3. *Late Blight* (Busuk Daun)

Late Blight (LB) atau penyakit busuk daun disebabkan oleh ‘jamur’ *Phytophthora Infestans* yang mana penyakit ini akan sangat merusak bagi tanaman tomat yang ditanam di lingkungan lembab dan dapat menyebabkan gagal panen total hanya dalam beberapa hari (Lage, Marouelli and Café-Filho, 2019). Semua bagian tanaman bisa terpengaruh oleh penyakit ini dan juga bisa mengakibatkan pembusukan pada buah. Penyakit busuk daun dapat menginfeksi daun muda (atas) atau tua (bawah). Penyakit ini ditandai dengan munculnya sebagai bercak air hijau pucat mulai dari ujung daun yang membesar dengan cepat, membentuk bercak hitam kehijauan yang tidak beraturan. Ini dapat berkembang dengan cepat ketika daun basah atau kelembaban udara tinggi (Sanoubar and Barbanti, 2017).

2.4.4. *Leaf Mold* (Bercak Daun karena Jamur)

Bercak daun disebabkan oleh jamur *Cladosporium fulfum*, merupakan salah satu penyakit daun tomat yang paling merusak tanaman dan tumbuh dalam kondisi lingkungan yang lembab. Penyakit ini merupakan masalah yang umum terjadi pada

tomat yang dibudidayakan di rumah kaca, namun kadang dapat muncul pada tomat yang ditanam di ladang dengan lingkungan lembab (Veloukas *et al.*, 2007).

2.4.5. Septoria Leaf Spot

Bintik daun Septoria disebabkan oleh jamur *Septoria lycopersici*. Bercak daun Septoria menyebar dengan cepat dan dapat membuat defoliasi dan melemahkan tanaman, itu mengakibatkan tanaman tidak dapat menghasilkan buah hingga matang. Penyakit ini dapat dikenali dengan munculnya bintik-bintik kecil yang direndam air pada bagian bawah daun yang lebih tua dan bagian bawah tanaman yang segera menjadi bintik melingkar. Lesi secara bertahap mengembangkan pusat-pusat putih keabu-abuan dengan tepi gelap. Pusat-pusat berwarna terang dari bintik-bintik ini adalah gejala paling khas dari bercak daun Septoria (Sanoubar and Barbanti, 2017).

2.4.6. Spider Mites Two Spider Mite

Spider Mites Two Spider Mite atau tungau laba laba berbintik dua (*Tetranychus urticae* Koch) adalah hama pertanian yang menyerang lebih dari 140 famili tanaman yang berbeda dan 1.100 spesies tanaman termasuk tomat. Tungau laba-laba ini berkembang dengan baik ketika kondisi hangat dan kering. Tungau laba-laba menyerang tanaman tomat dengan memasukkan *stylet* mereka ke dalam jaringan daun dan menghilangkan isi sel. Hilangnya klorofil daun berpengaruh terhadap fotosintesis yang dilakukan daun, ini menyebabkan perubahan warna daun, sering disebut *bronzing* juga mengakibatkan penurunan kesehatan secara keseluruhan atau bahkan mengakibatkan kematian (Rakha *et al.*, 2017).

2.4.7. Target Spot

Target spot banyak tersebar di negara beriklim tropis. Infeksi dari penyakit ini ditandai dengan bintik-bintik berbentuk tidak beraturan dengan garis tepi berwarna kuning. Bintik-bintik membesar hingga menyebar ke seluruh permukaan daun. infeksi ini disebabkan oleh jamur, mengakibatkan tanaman kehilangan banyak daun dan penurunan hasil panen. Penyakit ini bisa saja terjadi baik di rumah kasa maupun di lapangan.

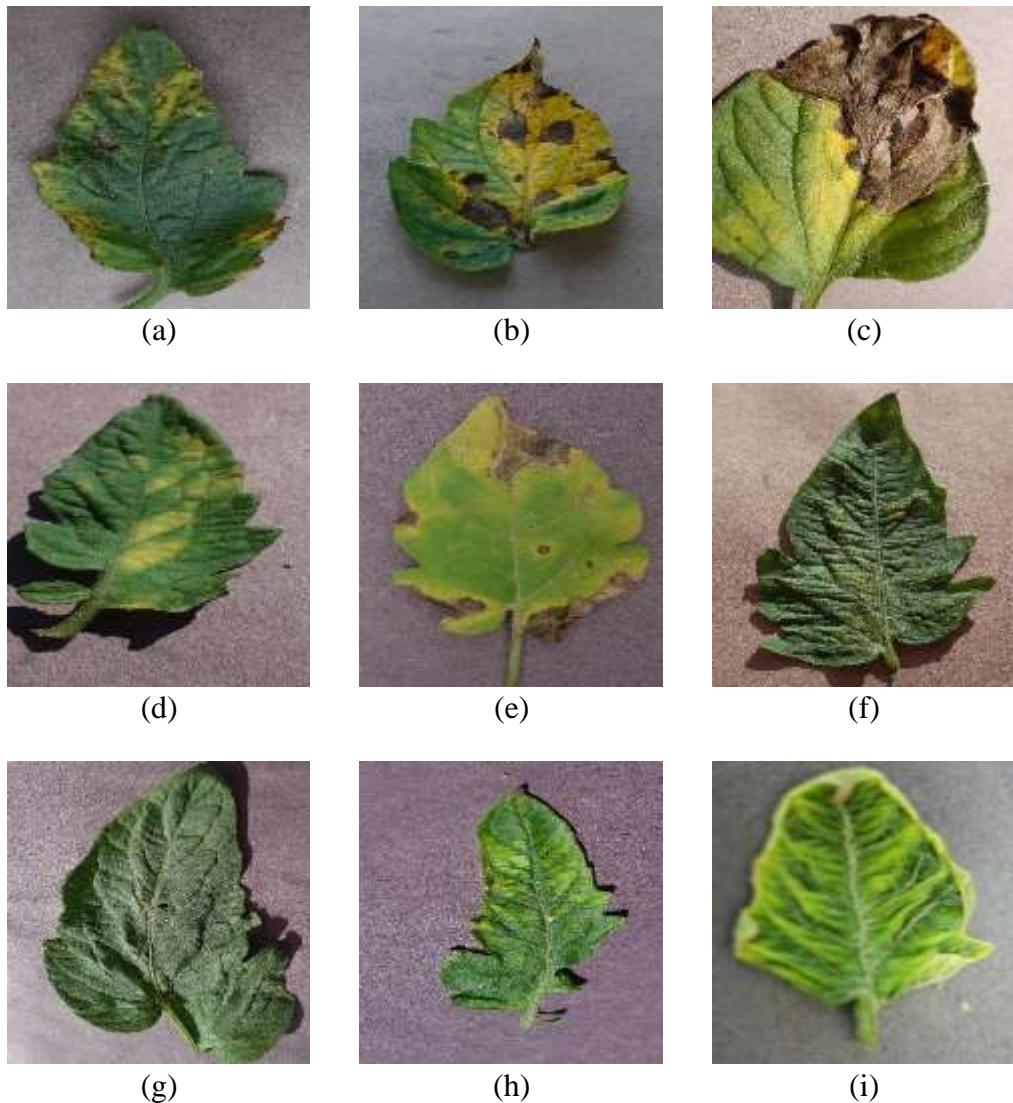
2.4.8. Tomato Mosaic Virus

Mosaic virus merupakan salah satu virus yang menyerang tomat, bahkan bukan hanya tomat tapi juga lada, kentang, apel, pir, ceri, dan lainnya. virus ini memiliki gejala berupa bercak hijau muda dan tua pada daun, daun mungkin melengkung, berbentuk aneh dan ukurannya mengecil. Jika tanaman terinfeksi lebih awal, tanaman akan nampak lebih kecil dan menguning. akibat kondisi yang memburuk juga berdampak pada buah yang dihasilkan yakni biasanya terjadi matang yang tidak rata, bahkan jumlah buah akhirnya berkurang begitu juga dengan ukurannya.

2.4.9. Tomato Yellow Leaf Curl Virus

Tomato yellow leaf curl virus merupakan penyakit tomat yang disebabkan sebuah virus. Tanaman tomat yang terinfeksi akan sulit untuk tumbuh atau mengerdil. Daun dari tanaman yang terinfeksi berukuran kecil dan melengkung keatas serta menunjukkan warna kekuningan. Bahkan dalam beberapa kasus, tanaman terinfeksi yang sejak dini pertumbuhan tanaman menyerupai bonsai karena tidak tumbuh namun berdaun agak lebat. meskipun berdaun lebat virus ini

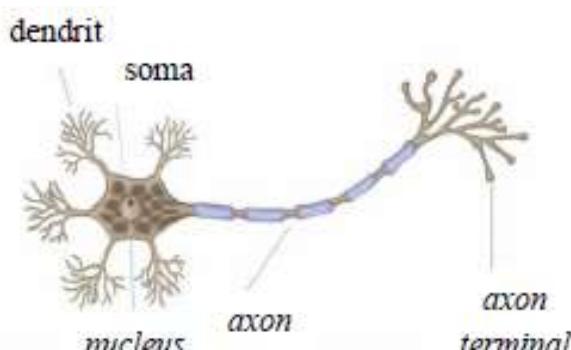
mengakibatkan gugurnya setiap bunga yang tumbuh. Akibat pengerdilan ini mengakibatkan penurunan jumlah panen.



Gambar 2.3 Penyakit Tanaman Tomat (a) *Bacterial Spot*, (b) *Early Blight*, (c) *Late Blight*, (d) *Leaf Mold*, (e) *Septoria Leaf Spot*, (f) *Spider Mites Two Spider Mite*, (g) *Target Spot*, (h) *Tomato Mosaic Virus*, (i) *Tomato Yellow Leaf Curl Virus*

2.5. Artificial Neural Network

Kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan sebuah studi tentang bagaimana membuat komputer dapat melakukan pekerjaan manusia. Salah satu bentuk dari AI adalah *Artificial Neural Network* (ANN) atau Jaringan Syaraf Tiruan (JST) yang menggunakan cara kerja dari otak manusia dengan tujuan meniru kecerdasan yang dimiliki oleh manusia. Otak manusia memproses informasi yang didapat dari alat indera manusia. Pemrosesan dilakukan oleh neuron yang saling terhubungan antar jaringan syaraf dan bekerja pada sinyal listrik yang melewatkannya. Jaringan syaraf menerapkan logika membuka dan menutup gerbang untuk mengarahkan sinyal listrik sesuai dengan objek yang diterima dari alat indera (Giuseppe Ciaburro, 2017).

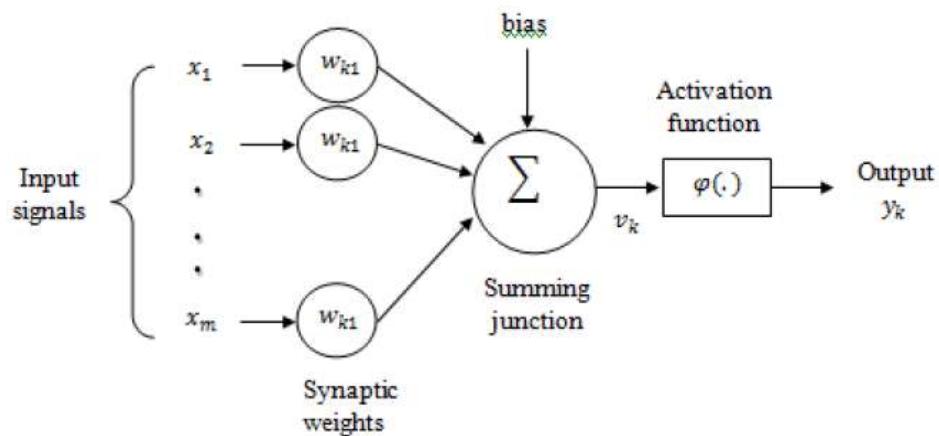


Gambar 2.4 Jaringan Syaraf Manusia

ANN memiliki cara kerja seperti jaringan syaraf biologi, sehingga bisa dikatakan bahwa ANN merupakan suatu generalisasi matematis dari cara kerja jaringan syaraf manusia. (Giuseppe Ciaburro, 2017) memberikan asumsi cara kerja dari sebuah *Neural Networks* sebagai berikut:

1. Pemrosesan informasi terjadi pada neuron.
2. Sinyal mengalir diantara sel syaraf melalui suatu node.

3. Setiap node memiliki bobot yang bersesuaian. Bobot ini digunakan untuk menggandakan sinyal yang dikirim oleh neuron sebelumnya
4. Setiap neuron menerapkan fungsi aktivasi terhadap hasil jumlah sinyal dengan bobot untuk menentukan sinyal output dari neuron tersebut dan disalurkan ke node dan neuron berikutnya



Gambar 2.5 Jaringan Syaraf Tiruan (JST)

Bobot (*weight*) dan fungsi aktivasi menentukan sifat neuron pada ANN. Dalam satu layer memiliki fungsi aktivasi yang sama, sedangkan antar layer bisa saja menggunakan fungsi aktivasi yang berbeda.

Ada berbagai macam *Activation Function* yang biasa dipakai pada CNN, setidaknya ada dua jenis *Activation Function* yaitu *Linear* dan *Non-Linear*, yang umum digunakan diantaranya:

1. *Linear Function*

Linear bisa diartikan sebagai *Actifation Layer Default* dari sebuah neuron, jika suatu neuron menggunakan fungsi aktifasi *Linear* maka output yang dihasilkan

dari meuron tersebut adalah *weighted sum* dari input ditambah dengan nilai bias.

Linear function bisa dirumuskan dalam persamaan 2.5.

$$f(x) = x \quad (2.5)$$

2. *Sigmoid Function (Non-Linear)*

Sigmoid Function merupakan salah satu dari *Activation Function Non-Linear* yang mempunyai rentang nilai antara 0 dan 1. Biasanya digunakan untuk klasifikasi dari dua kelas atau kelompok data. *Sigmoid Function* dirumuskan dalam persamaan 2.6.

$$\sigma(a) = \frac{1}{1 + e^{-a}} \quad (2.6)$$

3. *Tanh Function (Non-Linear)*

Sama halnya dengan *Sigmoid Function*, *Tanh Function* juga digunakan untuk melakukan klasifikasi dari dua kelas data, bedanya *Tanh Function* memiliki rentang antara -1 dan 1. *Tanh Function* dirumuskan dalam persamaan 2.7.

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (2.7)$$

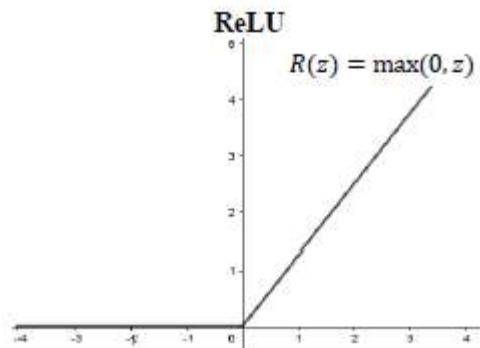
4. *ReLU Function (Non-Linear)*

Berbeda dengan *Sigmoid* dan *Tanh*, *ReLU (Rectified Linear Unit)* memiliki nilai antara nol dan tak terhingga. Bisa dikatakan fungsi aktifasi ini menutupi kekurangan dari fungsi aktifasi lainnya. fungsi ini melakukan *Thresholding* dengan nilai nol terhadap nilai piksel pada input citra. Fungsi aktivasi *ReLU* membuat seluruh nilai piksel yang negatif pada suatu citra menjadi nol dan mempertahankan nilai yang positif. Diantaranya kelabihan dari fungsi aktivasi *ReLU*

- Fungsi Aktivasi ReLU merupakan fungsi aktivasi default ketika mengembangkan *Multi-Layer Perceptron* dan CNN.
- ReLU mengatasi masalah *Gradient Dencent* yang hilang, yang memungkinkan model jaringan dapat belajar lebih cepat dan memiliki kinerja lebih baik.
- Melatih jaringan lebih cepat sehingga mengurangi kemungkinan *Overfitting*.

Representasi fungsi aktifasi ReLU dituliskan dalam persamaan 2.8.

$$ReLU(a) = \max(0, a) \quad (2.8)$$



Gambar 2.6 Fungsi Aktivasi ReLU

5. Softmax Function (Non-Linear)

Softmax Function akan mentransformasikan input menjadi *probability distribution*. *Softmax* memiliki nilai antara 0 dan 1. *Softmax* dapat digunakan untuk menentukan seberapa yakin model mengenali image dan melakukan klasifikasi. Fungsi aktivasi *softmax* berada pada lapisan yang menghubungkan antara *Fully-*

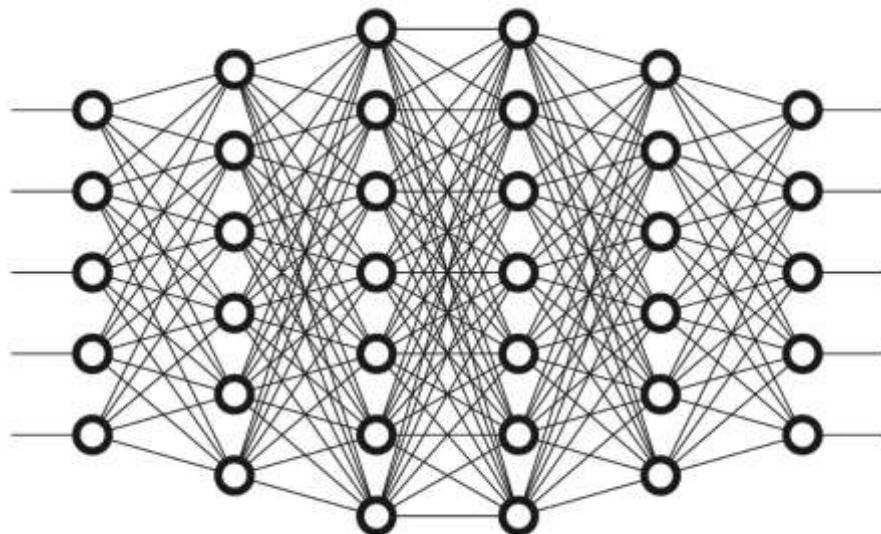
Connected Layer dengan *Dense Connection*. *Softmax* didefinisikan dalam persamaan 2.9.

$$\sigma_i(a)_i = \frac{e^{a_i}}{\sum_{j=1}^k e^{a_j}} \quad (2.9)$$

Layer penyusun ANN dapat dibagi tiga yaitu *Input layer*, *Hidden layer*, dan *Output layer*. *Input layer* merupakan representasi bentuk satu dimensi dari objek atau permasalahan yang diinputkan. Jumlah neuron pada input layer disesuaikan dengan jumlah inputan yang akan diterima. Setelah *Input layer*, data yang diolah dihadapkan dengan *Hidden layer*. Pada lapisan ini dilakukan fungsi aktivasi untuk melakukan penyaringan berdasarkan bobot masing masing node dari neuron pada *Input layer*. *Output layer* menerima hasil aktivasi *Hidden layer* lalu dilakukan fungsi aktivasi lagi pada neuronnya, dan hasil dari aktivasi pada *Output layer* ini lah yang menjadi hasil dari seluruh proses pada ANN. Proses ini disebut dengan *Forward Pass* yang mana aliran data masih searah dari input menuju output. Tahapan selanjutnya pada ANN adalah *Back Propagation*. *Back Propagation* bertujuan untuk mengupdate nilai *weight* pada masing masing node dan neuron. Seperti namanya, *Back Propagation* mengalirkan kembali data pada *output layer* kembali menuju *input layer* untuk untuk mengupdate *weight* dari masing masing neuron. Proses antara *Forward Pass* dan *Back Propagation* akan diulangi sebanyak *epoch* yang diperlukan, semakin banyak perulangan akan menghasilkan nilai toleransi atau loss yang semakin kecil.

Dalam setiap lapisan atau *layer* pada *artificial naural network*, bobot pada lapisan tersebut diubah atau diupdate dengan menggunakan fungsi aktivasi. Selain

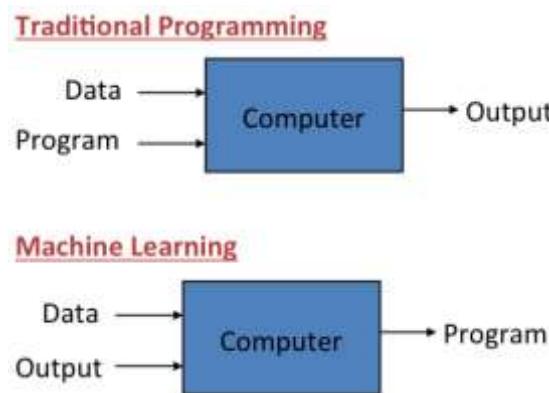
jumlah epoch, jumlah lapisan dan neuron juga sangat berpengaruh terhadap hasil yang akan didapatkan. Semakin kompleks suatu Neural Network, akan memiliki hasil akurasi yang semakin tinggi dengan nilai toleransi yang semakin kecil, oleh karenanya dalam Neural Network mengenal intilah yang membedakan jenis Neural Network berdasarkan jumlah layernya. Diantaranya Single Perceptron, Multi-Layer Perceptron, dan Competitive Layer.



Gambar 2.7 *Multi-Layer Perceptron (MLP)*

Machine Learning atau pembelajaran mesin adalah pendekatan dalam *Artificial Intelligence (AI)* yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, *Machine Learning* mencoba menirukan bagaimana manusia belajar dan menggeneralisasi. *Machine Learning* membantu menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. *Machine Learning* dapat membantu

komputer memprogram diri mereka sendiri. Jika pemrograman adalah pekerjaan untuk membuat otomatis, maka *Machine Learning* mengotomatisasi proses otomatis.



Gambar.2.8 Konsep *Machine Learning*

Ciri khas dari *Machine Learning* adalah adanya proses pelatihan, pembelajaran atau *training*. Oleh karena itu, agar bisa mengaplikasikan teknik-teknik *Machine Learning* maka dibutuhkan data latih. Tanpa data yang memadai *Machine Learning* tidak dapat bekerja. Data yang digunakan dibagi menjadi dua, yaitu data untuk pelatihan dan untuk pengujian. Data pelatihan digunakan untuk melatih algoritma, sedangkan data uji digunakan untuk mengetahui performa algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat. Setidaknya ada dua pemanfaatan yang umum digunakan dengan *Machine Learning* yaitu, klasifikasi dan prediksi.

Klasifikasi adalah metode dalam *Machine Learning* yang digunakan oleh mesin untuk memilah atau mengklasifikasikan obyek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari

suatu data masukan berdasarkan data yang sudah dipelajari dalam *training*. Ada empat jenis cara pembelajaran pada machine learning, yaitu *Supervised Learning* yang mana data pembelajaran mencakup keluaran yang sudah ditentukan. *Unsupervised Learning* yang mana data pembelajaran tidak mencakup keluaran yang ditentukan. *Semi-supervised Learning* yang mana data pembelajaran mencakup beberapa keluaran yang ditentukan. Dan yang terakhir *Reinforcement Learning* yang mana pemberian hadiah dari setiap serangkaian tindakan yang dilakukan.

Pengembangan dari *machine learning* dengan penerapan jaringan saraf tiruan disebut dengan *Deep Learning*. Secara deskriptif *Deep Learning* merupakan cabang ilmu dari *Machine Learning* berbasis jaringan saraf tiruan yang mengajarkan komputer untuk melakukan suatu tindakan yang dianggap alami oleh manusia, seperti belajar dari pengalaman. Lapisan atau *Layer* pada *Deep Learning* terdiri atas tiga bagian yaitu *Input Layer*, *Hidden Layer* dan *Output Layer*. *Input Layer* berisi node-node yang masing-masing menyimpan sebuah nilai masukan yang tidak berubah pada fase latih dan hanya bisa berubah jika diberikan nilai masukan baru. Pada *Hidden Layer* dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar meminimalisir error pada output. Kemudian output layer berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada lapisan *Hidden Layer* berdasarkan input yang diterima. Penambahan lebih banyak lapisan menjadikan model pembelajaran yang bisa mewakili citra berlabel dengan lebih baik. Aplikasi konsep *Artificial Neural Network* (ANN) yang lebih dalam atau yang memiliki banyak lapisan dapat ditangguhkan pada algoritma

Machine Learning yang sudah ada sehingga komputer bisa belajar dengan skala yang besar, lebih kecepatan, dan lebih akurat. Prinsip tersebut semakin berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas skala industri dan riset dalam menyelesaikan masalah data besar seperti pada *Computer Vision*, *Speech Recognition*, dan *Neural Languange Processing*.

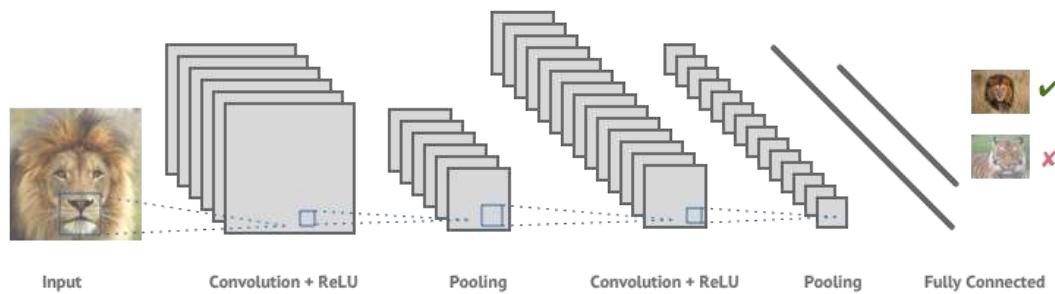
Deep learning yang merupakan pengembangan dari *Artificial Neural Network* (ANN) juga biasa disebut sebagai *Deep Neural Network* (DNN). *Layer* Arsitektur didalam DNN biasa disebut sebagai *Multi-Layer Perceptron* (MLP). Arsitektur pertama mempunyai 3 buah neuron pada *Input layer* dan 2 buah node *Output layer*. Diantara input dan output, terdapat 1 *Hidden layer* dengan 4 buah neuron. Neuron-neuron tersebut akan terhubung langsung dengan neuron lain pada layer selanjutnya.

Deep Learning memungkinkan model komputasi yang terdiri dari beberapa *Processing layer* untuk mempelajari representasi data dengan berbagai tingkat abstraksi, metode ini membuat perbaikan dalam pengenalan suara (*Speech Recognition*), pengenalan objek visual (*Visual Object Recognition*), deteksi objek (*Object Detection*) dan banyak lainnya.

2.6. *Convolutional Neural Network (CNN)*

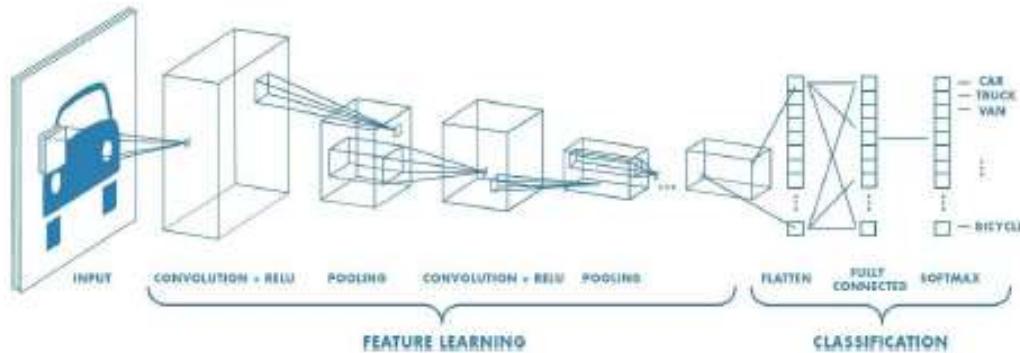
Convolutional Neural Network (CNN) merupakan pengembangan dari *Multi-Layer Perceptron* (MPL) yang dirancang untuk mengolah data dalam bentuk grid, salah satunya adalah citra dua dimensi. CNN digunakan untuk

mengklasifikasikan data yang terlabel dengan menggunakan metode *Supervised Learning*, yang mana di dalam *Supervised Learning* terdapat data yang dilatih dan variable yang ditargetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atau mendetaksi suatu objek.



Gambar 2.9 Illustrasi *Convolutional Neural Network* (CNN)

Convolutional Neural Network menggabungkan tiga pokok arsitektur, yaitu *Local Connection*, *Shared Weight* berupa *filter*, *Spatial Subsampling* yang berupa *Pooling* (Altunta, Cömert and Fatih, 2019). Berbagi bobot antar unit di lokasi berbeda cenderung mencari pola yang sama di berbagai bagian data input (Altunta, Cömert and Fatih, 2019). *Pooling Layer* menggabungkan fitur yang hampir sama menjadi satu, mengurangi dimensi representasi dan mencegah *overfitting* (Yu *et al.*, 2017). Konvolusi atau *Convolution* merupakan matris yang berfungsi untuk melakukan *filtering*.

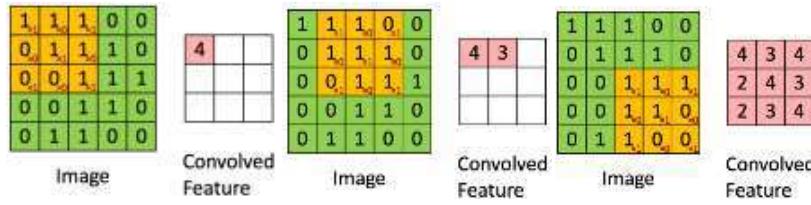


Gambar 2.10 Arsitektur *Convolutional Neural Network* (CNN)

Tahapan dalam arsitektur CNN diantaranya tahap konvolusi dengan menggunakan *Convolutional Layer* dan lapisan *filter*, lalu dilakukan aktivasi fungsi dari hasil konvolusi, dan selanjutnya dilakukan penggabungan dengan *Pooling Layer*. Proses ini diulang terus menerus sampai didapatkan fitur yang cukup untuk dilanjutkan ke *Fully-Connected Layer* untuk tahap akhir dari algoritma yakni pengklasifikasian data.

2.6.1. *Convolutional Layer (Conv. Layer)*

Proses dalam *Convolutional Layer* merupakan proses utama yang mendasari algoritma CNN. *Convolution layer* melakukan operasi konvolusi pada nilai output dari lapisan sebelumnya atau *input layer*. Konvolusi merupakan suatu istilah matematis yang dalam pengolahan citra berarti mengaplikasikan sebuah kernel pada citra disemua *offset* yang memungkinkan. Dalam Gambar 2.11, kotak kuning merupakan kernel yang melakukan operasi konvolusi sedangkan kotak berwarna hijau secara keseluruhan merupakan citra yang akan dikonvolusi. Kernel (kotak kuning) bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya.



Gambar 2.11 Ilustrasi Proses Konvolusi Citra

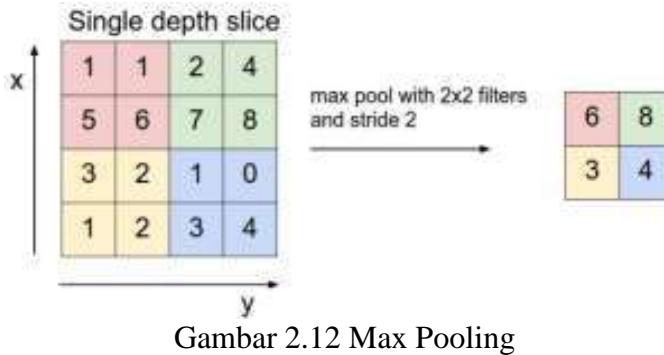
Tujuan konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data yang berkenaan pada posisi ruang atau informasi lokal. Bobot pada lapisan konvolusi menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.

2.6.2. *Pooling layer*

Pooling Layer merupakan lapisan yang menggunakan fungsi dengan *feature map* sebagai masukan dan mengolahnya dengan berbagai macam operasi statistik berdasarkan nilai piksel terdekat. *Pooling layer* pada model CNN biasanya disisipkan secara teratur setelah beberapa *Convolution layer*. *Pooling layer* yang dimasukkan di antara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *feature map*, sehingga jumlah parameter dan perhitungan di jaringan berkurang, serta untuk mengurangi kemungkinan *Overfitting*.

Pooling layer digunakan untuk mengambil nilai maksimal (*max-pooling*) atau nilai rata-rata (*average pooling*) dari bagian-bagian piksel pada citra dalam satu kernel. Metode yang lebih sering digunakan dalam adalah metode *max-pooling*. *Max pooling* membagi output dari *convolution layer* menjadi beberapa grid

kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi



Gambar 2.12 Max Pooling

Pada Gambar 2.19 kotak yang berwarna merah, hijau, kuning dan biru pada sisi kiri merupakan kelompok kotak yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan kotak disebelah kanan. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran). Penggunaan *pooling layer* pada CNN bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan. *Stride* merupakan parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah satu, maka filter akan bergeser sebanyak satu piksel secara horizontal lalu vertikal. Semakin kecil *stride* yang digunakan, maka semakin detail informasi yang didapatkan dari sebuah input, namun membutuhkan komputasi lebih jika dibandingkan dengan *stride* yang besar.

2.6.3. Fully-Connected Layer

Fully-Connected layer merupakan lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya.

Fully-Connected layer biasanya digunakan dalam penerapan *Multi-Layer Perceptron* (MLP) dan bertujuan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linear.

Perbedaan antara *Fully-Connected layer* dan *Convolution layer* biasanya adalah neuron di *Convolution layer* terhubung hanya ke daerah tertentu pada input, sedangkan *Fully-Connected layer* mempunyai neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoprasikan produk dot, sehingga fungsinya tidak begitu berbeda.

Hyperparameter merupakan parameter yang perlu untuk diatur sebelum melakukan training dalam CNN. Variable dalam *hyperparameter* menentukan bagaimana sebuah jaringan dilatih. Pengaturan *hyperparameter* bertujuan untuk mengoptimalkan bobot dan bias. Diantaranya hyperparameter adalah *Learning rate* yang menentukan seberapa cepat jaringan memperbarui bobot dan biasnya. *Epoch* merupakan proses iterasi pada dataset yang berulang setelah proses training. *Stride* adalah nilai ukuran pergeseran yang digunakan untuk menggeser filter. *Padding* merupakan penambahan piksel bernilai 0 pada bingkai luar citra.

2.7. *Transfer Learning*

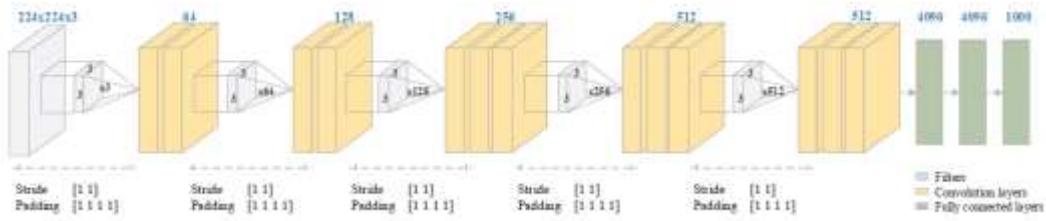
Ada dua cara pendekatan dasar untuk melatih model CNN, yakni dengan membentuk jaringan dari awal (*Scratch*) atau dengan pembelajaran transfer (*Transfer Learning*) (Altunta, Cömert and Fatih, 2019). Melatih model CNN dari awal (*scratch*) membutuhkan banyak sample objek untuk mengungkap fitur

deskriptif yang ada dalam citra. Meskipun pendekatan ini memberikan kendali penuh terhadap jaringan yang dibuat, waktu yang dibutuhkan untuk melakukan pelatihan model akan sangat lama, selain itu juga masalah *overfitting* dan konvergensi adalah permasalahan yang potensial dihadapi dalam proses pelatihan.

Untuk mengatasi masalah ini, model CNN dilatih dengan menggunakan *fine-tuning* atau pendekatan ini disebut *transfer learning*. Dalam pendekatan pembelajaran transfer yang umum, lapisan konvolusi digunakan sebagai fitur ekstraktor yang tetap, dan hanya *fully-connected layer* yang disesuaikan dengan tugas klasifikasi yang baru. *Fine-tunning* dimulai dengan mentransfer bobot dari jaringan yang telah dilatih (*pre-trained network*) ke jaringan yang baru yang akan dilatih. Lapisan-lapisan terakhir termasuk didalamnya *fully-connected layer*, *softmax*, dan layer klasifikasi dipisahkan dari jaringan, kemudian konfigurasi baru direalisasikan untuk tugas spesifik baru yang akan dilakukan.

2.7.1. VGG

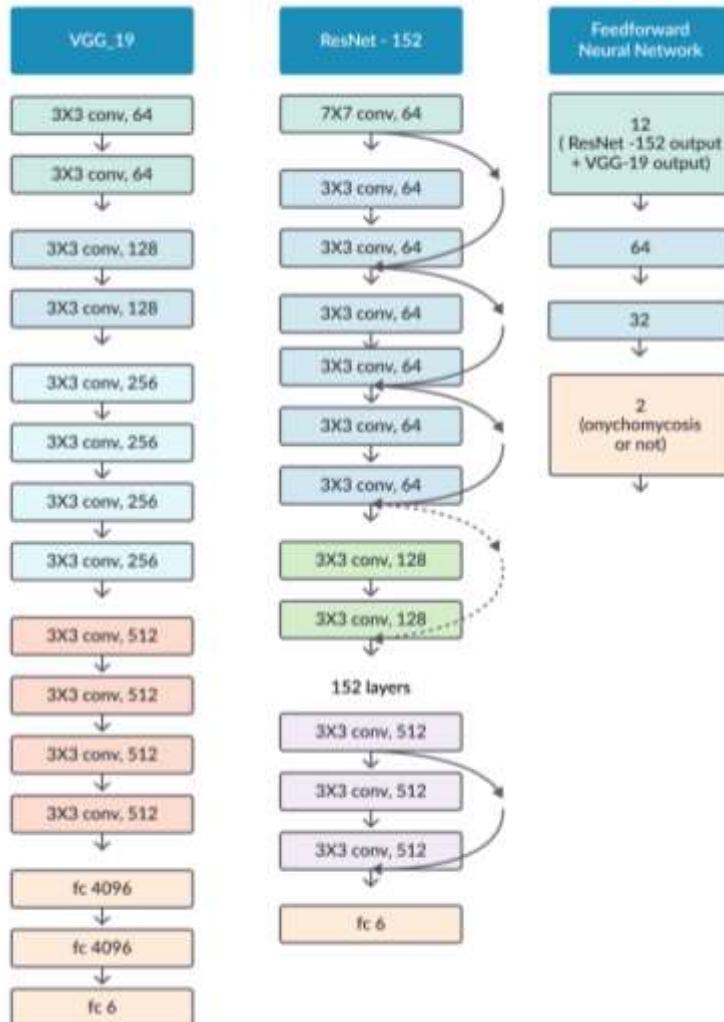
Latar belakang terbentuknya arsitektur jaringan VGG adalah untuk menyelidiki efek kedalaman jaringan konvolusional atas akurasinya dalam pengenalan gambar skala besar. Model VGG memiliki dua kunci utama, yakni arsitekturnya dan kedalaman lapisan konvolusinya. Model ini memberikan peningkatan yang signifikan dengan memberikan kedalaman 16 sampai dengan 19 lapisan konvolusi



Gambar 2.13 Arsitektur jaringan VGG

2.7.2. ResNet

Model arsitektur jaringan *ResNet* mengedepankan kerangka belajar residual untuk mempermudah pelatihan jaringan. *ResNet* adalah arsitektur jaringan CNN yang memungkinkan untuk menggunakan ratusan bahkan ribuan lapisan konvolusi. Semakin dalam suatu jaringan neural network, muncul suatu masalah yaitu hilangnya nilai gradient atau yang dinamakan *vanishing gradient*. *ResNet* menumpuk pemetaan identitas, lapisan yang awalnya tidak melakukan apa-apa, dan melompati mereka, menggunakan kembali aktivasi dari lapisan sebelumnya. Melewati awalnya memampatkan jaringan menjadi hanya beberapa lapisan, yang memungkinkan pembelajaran lebih cepat. Kemudian, ketika jaringan melatih lagi, semua lapisan diperluas dan bagian residual dari jaringan mengeksplorasi semakin banyak fitur dari gambar.



Gambar 2.14 Arsitektur ResNet Dibandingkan dengan VGG dan *Feed Forward*

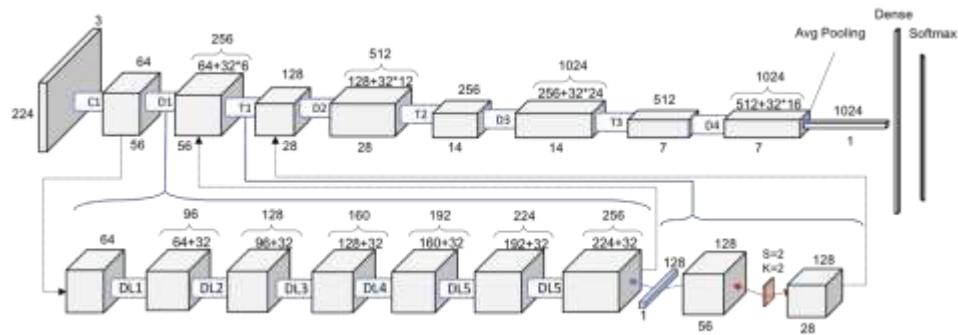
Neural Network

2.7.3. DenseNet

DenseNet merupakan Jaringan Konvolusional yang terkoneksi Secara padat.

DenseNets mengeksplorasi potensi jaringan melalui penggunaan kembali fitur, cara ini membutuhkan parameter yang lebih sedikit dibandingkan dengan model CNN lain yang setara. *DenseNets* terdiri dari blok-blok bernama *DenseBlocks*, di mana dimensi dari peta fitur tetap konstan di dalam blok, tetapi jumlah filter

berubah di antara mereka. Dan diantara masing-masing blok terdapat *transition layer* yang mengurus *downsampling*.



Gambar 2.15 Arsitektur *DenseNet* dengan Model *DenseBlocks*

BAB V

SIMPULAN DAN SARAN

5.1. Simpulan

Dengan semua proses penelitian dan hasil dari analisis yang dilakukan, diperoleh kesimpulan sebagai berikut.

1. Penelitian ini melakukan identifikasi penyakit tanaman tomat berdasarkan citra daun tomat dengan menggunakan metode *computer vision* dan algoritma *Convolutional Neural Network*.
2. Data yang digunakan merupakan data sekunder dari dataset *PlantVillage* dengan dimensi input yakni 244, 244 piksel. Data yang digunakan memiliki 10 kelas diantaranya 9 kelas penyakit dan 1 kelas sehat.
3. Penggunaan algoritma CNN menerapkan pendekatan *transfer learning*. Proses pelatihan dilakukan dengan membekukan *base model* dan menambahkan beberapa *layer* tambahan. *Learning rate* dalam pelatihan yang digunakan adalah 0,001 dan dropout 0,5.
4. Dalam penelitian ini menggunakan 3 jenis model CNN dengan masing masing model terdapat 2 variasi model, sehingga totalnya menggunakan 6 buah model berbeda untuk melakukan identifikasi penyakit tanaman tomat. Keenam model tersebut adalah VGG-16, VGG-19, ResNet-50, ResNet-101, DenseNet-121, dan DenseNet-169.
5. Dengan menggunakan pendekatan transfer learning mampu meminimalisir terjadinya overfitting dalam proses pelatihan model.

6. Metode evaluasi dilakukan dengan memprediksi kelas dari data uji lalu dilakukan evaluasi dengan menggunakan *confusion matrix* untuk mendapatkan perhitungan penilaian model yakni *accuracy*, *precision*, *recall*, *f1-score*, dan AUC
7. Model dengan performa terbaik dalam melakukan klasifikasi dicapai oleh model ResNet-50 dengan nilai *accuracy* 96,16%, *precision* 97%, *recall* 96%, *F1-score* 97% dan AUC 0,9792.

5.2. Saran

Saran yang diberikan untuk penelitian ini adalah sebagai berikut.

1. Penelitian selanjutnya diharapkan dapat menggunakan dataset dengan jumlah yang lebih banyak, sehingga dapat menghasilkan performa model yang lebih baik, atau bisa dengan menggunakan data primer untuk pengujinya.
2. Penelitian selanjutnya diharapkan bisa menggunakan metode penilaian yang lain seperti *K-Fold Cross Validation*, sehingga bisa digunakan sebagai acuan penilaian performa model dari sudut pandang yang lain.
3. Penelitian selanjutnya diharapkan bisa memperbanyak kalas data atau menerapkan metode CNN kedalam objek penelitian yang lain.
4. Diharapkan aplikasi bisa dibuat dengan lebih *Advance*, akan lebih baik jika bisa *Machine Learning* yang dikembangkan bisa menerapkan *Self Learning*.

DAFTAR PUSTAKA

- Adhikari, P., Oh, Y. and Panthee, D. R. (2017) ‘Current status of early blight resistance in tomato: An update’, *International Journal of Molecular Sciences*, 18(10). doi: 10.3390/ijms18102019.
- Alex Krizhevsky, Ilya Sutskever, G. E. H. (2007) ‘ImageNet Classification with Deep Convolutional Neural Networks’, *Handbook of Approximation Algorithms and Metaheuristics*, pp. 1–1432. doi: 10.1201/9781420010749.
- Ali, M. M. *et al.* (2019a) ‘Non-destructive techniques of detecting plant diseases: A review’, *Physiological and Molecular Plant Pathology*. Elsevier Ltd, 108(July), p. 101426. doi: 10.1016/j.pmpp.2019.101426.
- Ali, M. M. *et al.* (2019b) ‘Non-destructive techniques of detecting plant diseases: A review’, *Physiological and Molecular Plant Pathology*. Elsevier Ltd, 108(June), p. 101426. doi: 10.1016/j.pmpp.2019.101426.
- Altunta, Y., Cömert, Z. and Fatih, A. (2019) ‘Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach’, *Computers and Electronics in Agriculture*, 163(40), pp. 1–11. doi: 10.1016/j.compag.2019.104874.
- Brahimi, M., Boukhalfa, K. and Moussaoui, A. (2017) ‘Deep Learning for Tomato Diseases: Classification and Symptoms Visualization’, *Applied Artificial Intelligence*. Taylor & Francis, 31(4), pp. 299–315. doi:

- 10.1080/08839514.2017.1315516.
- Chebet, E. *et al.* (2018) ‘A comparative study of fine-tuning deep learning models for plant disease identification’, *Computers and Electronics in Agriculture*. Elsevier, (March), pp. 0–1. doi: 10.1016/j.compag.2018.03.032.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, S. R. and Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, A. R. (2015) ‘Going Deeper with Convolutions Christian’, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi: <https://doi.org/10.1109/CVPR.2015.7298594>.
- Cömert, Zafer, A. F. K. (2019) ‘Fetal Hypoxia Detection Based on Deep Convolutional Neural Network with Transfer Learning Approach’, *Csoc 2018, Aisc* 763, 763, pp. 66–74. doi: 10.1007/978-3-319-91186-1.
- Ferentinos, K. P. (2018) ‘Deep learning models for plant disease detection and diagnosis’, *Computers and Electronics in Agriculture*. Elsevier, 145(January), pp. 311–318. doi: 10.1016/j.compag.2018.01.009.
- Garcia-Garcia, A. *et al.* (2018) ‘A survey on deep learning techniques for image and video semantic segmentation’, *Applied Soft Computing Journal*. Elsevier B.V., 70, pp. 41–65. doi: 10.1016/j.asoc.2018.05.018.
- Giuseppe Ciaburro, B. V. (2017) *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. Packt Publishing Ltd.

- Gomes, J. F. S. and Leta, F. R. (2012) 'Applications of computer vision techniques in the agriculture and food industry: A review', *European Food Research and Technology*, 235(6), pp. 989–1000. doi: 10.1007/s00217-012-1844-2.
- Hameed, K., Chai, D. and Rassau, A. (2018) 'A comprehensive review of fruit and vegetable classification techniques', *Image and Vision Computing*. Elsevier B.V, 80, pp. 24–44. doi: 10.1016/j.imavis.2018.09.016.
- He, K. *et al.* (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- Kc, K. *et al.* (2019) 'Depthwise separable convolution architectures for plant disease classification', *Computers and Electronics in Agriculture*. Elsevier, 165(July), p. 104948. doi: 10.1016/j.compag.2019.104948.
- Kiani, E. and Mamedov, T. (2018) 'ScienceDirect ScienceDirect Identification of plant disease infection using soft-computing: Application to modern botany', *Procedia Computer Science*. Elsevier B.V., 120, pp. 893–900. doi: 10.1016/j.procs.2017.11.323.
- Kumlachew Alemu (2015) 'Detection of Diseases, Identification and Diversity of Viruses: A Review', *Journal of Biology, Agriculture and Healthcare*, 5, pp. 204–213. Available at: <https://www.semanticscholar.org/paper/Detection-of-Diseases%2C-Identification-and-Diversity-Alemu/e0ba4c8a63340f62bf9425e523b184cc611e1908>.

- Lage, D. A. C., Marouelli, W. A. and Café-Filho, A. C. (2019) ‘Management of powdery mildew and behaviour of late blight under different irrigation configurations in organic tomato’, *Crop Protection*, 125(February). doi: 10.1016/j.cropro.2019.104886.
- Lee, S. H. *et al.* (2020) ‘New perspectives on plant disease characterization based on deep learning’, *Computers and Electronics in Agriculture*. Elsevier, 170(January 2019), p. 105220. doi: 10.1016/j.compag.2020.105220.
- Lin, M., Chen, Q. and Yan, S. (2014) ‘Network in network’, *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pp. 1–10.
- Lu, Y. *et al.* (2017) ‘Identification of rice diseases using deep convolutional neural networks’, *Neurocomputing*. Elsevier B.V., 267, pp. 378–384. doi: 10.1016/j.neucom.2017.06.023.
- Martinelli, F. *et al.* (2015) ‘Advanced methods of plant disease detection. A review’, *Agronomy for Sustainable Development*, 35(1), pp. 1–25. doi: 10.1007/s13593-014-0246-1.
- Mohanty, S. P., Hughes, D. P. and Salathé, M. (2016) ‘Using deep learning for image-based plant disease detection’, *Frontiers in Plant Science*, 7(September), pp. 1–10. doi: 10.3389/fpls.2016.01419.
- Oppenheim, D. and Shani, G. (2017) ‘Potato Disease Classification Using Convolution Neural Networks’, *Advances in Animal Biosciences*, 8(2), pp. 244–249. doi: 10.1017/s2040470017001376.

- Osdaghi, E. *et al.* (2017) ‘Monitoring the occurrence of tomato bacterial spot and range of the causal agent *Xanthomonas perforans* in Iran’, *Plant Pathology*, 66(6), pp. 990–1002. doi: 10.1111/ppa.12642.
- Pulung Nurtantio Andono, T.Sutojo, M. (2017) *Pengolahan Citra Digital*. Edited by Arie Premesta. Penerbit Andi.
- Rakha, M. *et al.* (2017) ‘Evaluation of wild tomato accessions (*Solanum* spp.) for resistance to two-spotted spider mite (*Tetranychus urticae* Koch) based on trichome type and acylsugar content’, *Genetic Resources and Crop Evolution*. Springer Netherlands, 64(5), pp. 1011–1022. doi: 10.1007/s10722-016-0421-0.
- Rangarajan, A. K. and Purushothaman, R. (2018) ‘ScienceDirect ScienceDirect Tomato crop disease classification using pre-trained deep learning algorithm’, *Procedia Computer Science*. Elsevier B.V., 133, pp. 1040–1047. doi: 10.1016/j.procs.2018.07.070.
- Rumpf, T. *et al.* (2010) ‘Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance’, *Computers and Electronics in Agriculture*. Elsevier B.V., 74(1), pp. 91–99. doi: 10.1016/j.compag.2010.06.009.
- Sanoubar, R. and Barbanti, L. (2017) ‘Fungal diseases on tomato plant under greenhouse condition’, *European Journal of Biological Research*, 7(4), pp. 299–308. doi: 10.5281/zenodo.1011161.
- Shoaib, A., Awan, Z. A. and Khan, K. A. (2019) ‘Intervention of antagonistic

- bacteria as a potential inducer of disease resistance in tomato to mitigate early blight’, *Scientia Horticulturae*. Elsevier, 252(October 2018), pp. 20–28. doi: 10.1016/j.scienta.2019.02.073.
- Simonyan, K. and Zisserman, A. (2015) ‘Very deep convolutional networks for large-scale image recognition’, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14.
- Strayer-Scherer, A. et al. (2018) ‘Advanced copper composites against copper-tolerant xanthomonas perforans and tomato bacterial spot’, *Phytopathology*, 108(2), pp. 196–205. doi: 10.1094/PHYTO-06-17-0221-R.
- Tian, H. et al. (2019) ‘Computer vision technology in agricultural automation —A review’, *Information Processing in Agriculture*. China Agricultural University, 7(1), pp. 1–19. doi: 10.1016/j.inpa.2019.09.006.
- Veloukas, T. et al. (2007) ‘Management of tomato leaf mould caused by Cladosporium fulvum with trifloxystrobin’, *Crop Protection*, 26(6), pp. 845–851. doi: 10.1016/j.cropro.2006.08.005.
- Wan, P. et al. (2018) ‘A methodology for fresh tomato maturity detection using computer vision’, *Computers and Electronics in Agriculture*. Elsevier, 146(January), pp. 43–50. doi: 10.1016/j.compag.2018.01.011.
- Yu, Y. et al. (2017) ‘Deep Transfer Learning for Modality Classification of Medical Images’, *information*. doi: 10.3390/info8030091.