

**IMPLEMENTASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* PADA KLASIFIKASI PENYAKIT PADI MELALUI  
CITRA DAUN**

(Studi Kasus: Penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*)

**TUGAS AKHIR**



Siti Rahmah Danur Amiril

16611043

**PROGRAM STUDI STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2020**

## HALAMAN PERSETUJUAN PEMBIMBING

### TUGAS AKHIR

Judul : Implementasi Algoritma *Convolutional Neural Network* pada Klasifikasi Penyakit Padi melalui Citra Daun (Studi Kasus: Penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*)

Nama Mahasiswa : Siti Rahmah Danur Amiril

NIM : 16611043

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK  
DIUJIKAN**

Yogyakarta, 14 Oktober 2020

Pembimbing

  
(Dr. techn. Rohmatul Fajriyah, S.Si., M.Si.)

**HALAMAN PENGESAHAN**

**TUGAS AKHIR**

**IMPLEMENTASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* PADA KLASIFIKASI PENYAKIT PADI MELALUI CITRA DAUN**

(Studi Kasus: Penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*)

**Nama Mahasiswa : Siti Rahmah Danur Amiril**

**NIM : 16611043**

**TUGAS AKHIR INI TELAH DIUJIKAN  
PADA TANGGAL 29 Oktober 2020**

**Nama Penguji:**

**Tanda Tangan**

1. Ayundyah Kesumawati, S.Si., M.Si. .... 
2. Arum Handini Primandari, S.Pd.Si., M.Si ..... 
3. Dr. techn. Rohmatul Fajriyah, S.Si., M.Si. .... 

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Islam Indonesia



**Prof. Riyanto, S.Pd., M.Si., Ph.D**

## KATA PENGANTAR



*Assalamu'alaikum Wr.Wb*

Alhamdulillah rabbil 'alamin, segala puji dan syukur penulis panjatkan kepada Allah SWT Tuhan semesta alam yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyusun dan menyelesaikan Tugas Akhir yang berjudul “Implementasi Algoritma *Convolutional Neural Network* Pada Klasifikasi Penyakit Padi Melalui Citra Daun” dengan baik dan lancar sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang strata satu di Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia. Shalawat serta salam tak lupa penulis haturkan kepada Baginda Nabi Muhammad SAW beserta keluarga dan para sahabat, yang telah membimbing umatnya kepada jalan yang penuh berkah.

Penyusunan dan penulisan tugas akhir ini tidak akan terselesaikan tanpa adanya dukungan, bantuan dan bimbingan serta kemurahan hati dari banyak pihak. Oleh sebab itu, penulis ingin menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Kedua Orang Tua yang selalu menjadi semangat saya yaitu Mama Yersi Faradilla Taun dan Bapak Darmawani yang selalu mendoakan dan memberi nasihat serta yang telah banyak berjuang dan berkorban materi, tenaga dan pikiran agar anaknya ini dapat menyelesaikan perkuliahan dan Tugas Akhir ini.
2. Kedua adik yang saya sayangi yaitu Siti Aisyah Danur Amiril dan Siti Nur Badriyyah Danur Amiril yang menjadi semangat saya untuk menyelesaikan Tugas Akhir.
3. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.
4. Bapak Dr. Edy Widodo, S.Si., M.Si, selaku Ketua Program Studi Statistika, Universitas Islam Indonesia.

5. Ibu Dr. techn. Rohmatul Fajriyah, S.Si., M.Si, selaku Dosen Pembimbing tugas akhir yang telah banyak membantu, memberikan arahan dan membimbing dalam penyusunan tugas akhir ini serta selalu berbagi nikmat rezeki kepada kami para bimbingannya.
6. Bapak dan Ibu dosen Program Studi Statistika, Universitas Islam Indonesia yang telah mengajari dan membagikan banyak ilmu kepada saya dari awal kuliah hingga selesai, sehingga saya memperoleh banyak pengalaman serta ilmu terbaik yang dapat saya tuangkan dalam penyusunan Tugas Akhir ini.
7. Sahabat saya selama masa perkuliahan yaitu Nadya Puji Sarawati, Yolla Torina, Annisa Ayunda Permatasari, Havidzah Asri Isnandar dan Widya Sri Mulyaningsih, yang selalu memberikan semangat dan membantu saya ketika dalam kesulitan baik dalam hal akademik maupun dalam kehidupan sehari-hari, semoga persahabatan ini tidak pernah terputus sampai kapanpun.
8. Sahabat SMA yang sudah saya anggap sebagai saudari yaitu Yanti Tri Utami, Dayu Anggreani Utama, Erviyani, Dwita Safitri dan Reffi Shopia Melati yang selalu memberikan semangat serta selalu ada untuk saya di masa sulit maupun bahagia serta selalu saling mendoakan, semoga persahabatan ini selalu terjalin hingga akhir hayat.
9. Teman seperjuangan bimbingan tugas akhir Ibu Ema yang selalu memberikan motivasi dan semangat serta bantuan sehingga saya dapat menyelesaikan Tugas Akhir ini.
10. Teman KKN unit 148 tahun 2019 yang telah memberikan kenangan dan pengalaman baru selama masa KKN maupun sesudah KKN dan masih menjalin silaturahmi hingga saat ini.
11. Sepupu saya yang sedang berjuang bersama di perantauan yaitu Nelly Nurhaliza Danur Amiril dan Muhammad Akbar Nur Amiril yang selalu memberikan motivasi, semangat serta selalu mendoakan dan saling membantu baik dalam masa sulit dan bahagia.
12. Seluruh keluarga besar yang selalu memberikan semangat, nasihat, motivasi serta selalu mendoakan saya.

13. Teman-teman angkatan 2016 Program Studi Statistika yang menjadi tempat bertukar ilmu dan bertukar informasi terkait akademik maupun non akademik serta berjuang bersama untuk menyelesaikan kuliah dan menyanggah gelar sarjana.
14. Serta semua pihak-pihak lain yang tidak dapat saya sebutkan satu persatu yang telah banyak mendoakan, memberi motivasi serta membantu saya dalam penyusunan tugas akhir ini.

Walaupun penyusunan tugas akhir ini dilakukan dengan sebaik-baiknya, namun penulis menyadari bahwa tugas akhir ini masih kurang sempurna, sehingga sangat membutuhkan bantuan dari berbagai pihak, baik berupa saran dan kritik serta pembaharuan dan penyempurnaan. Penulis mengucapkan terimakasih dan mohon maaf atas segala kekurangan yang ada, karena kesempurnaan hanyalah milik Allah SWT. Semoga tugas akhir ini tidak hanya bermanfaat bagi penulis tetapi juga bermanfaat bagi para pembaca dimasa mendatang.

*Wassalamualaikum Wr.Wb*

Yogyakarta, 14 Oktober 2020



(Siti Rahmah Danur Amiril)

## DAFTAR ISI

HALAMAN Sampul.....	i
HALAMAN PERSETUJUAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
PERNYATAAN.....	xiii
ABSTRAK.....	xiv
<i>ABSTRACT</i> .....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 Penelitian Menggunakan Metode <i>Convolutional Neural Network</i> .....	6
2.2 Penelitian Tentang Klasifikasi Penyakit Tanaman Padi.....	9
2.3 Perbedaan Dengan Penelitian Sebelumnya.....	12
BAB 3 LANDASAN TEORI.....	13
3.1 Padi.....	13
3.1.1 Penyakit Padi <i>Bacterial Leaf Blight</i> .....	14
3.1.2 Penyakit Padi <i>Brown Spot</i> .....	15
3.1.3 Penyakit Padi <i>Leaf Spot</i> .....	16
3.2 Kecerdasan Buatan ( <i>Artificial Intelligence</i> ).....	17
3.3 <i>Deep Learning</i> .....	19
3.4 <i>Convolutional Neural Network (CNN)</i> .....	21

3.4.1	<i>Convolutional Operation</i> .....	22
3.4.2	Fungsi Aktivasi .....	23
3.4.3	<i>Pooling Layer</i> .....	25
3.4.4	<i>Fully Connected Layer</i> .....	26
3.4.5	<i>Backpropagation</i> .....	27
3.4.6	<i>Learning Rate</i> .....	27
3.4.7	<i>Crossentropy Loss Function</i> .....	28
3.4.8	<i>Softmax Classifier</i> .....	28
3.4.9	<i>Optimizer</i> .....	29
3.4.10	<i>Stochastic Gradient Descent (SGD)</i> .....	30
3.4.11	RMSprop.....	30
3.4.12	Adam .....	31
3.5	<i>EBImage</i> .....	33
3.6	Keras .....	33
3.7	Citra .....	33
3.8	Citra Warna <i>Red, Green, Blue (RGD)</i> .....	34
3.9	Pengolahan Citra ( <i>Image Processing</i> ) .....	35
3.10	Piksel ( <i>Pixel</i> ) .....	36
3.11	Pengenalan Pola ( <i>Pattern Recognition</i> ).....	36
3.12	Akurasi .....	37
3.13	R Studio.....	37
BAB 4	METODOLOGI PENELITIAN .....	38
4.1	Populasi Penelitian .....	38
4.2	Jenis dan Sumber Data.....	38
4.3	Variabel Penelitian .....	38
4.4	Metode Analisis Data.....	40
4.5	Tahapan Penelitian .....	40
BAB 5	HASIL DAN PEMBAHASAN .....	41
5.1	Pengambilan Data.....	41
5.2	<i>Pre-processing</i> Data Citra.....	42
5.3	Pelabelan Kategori.....	42

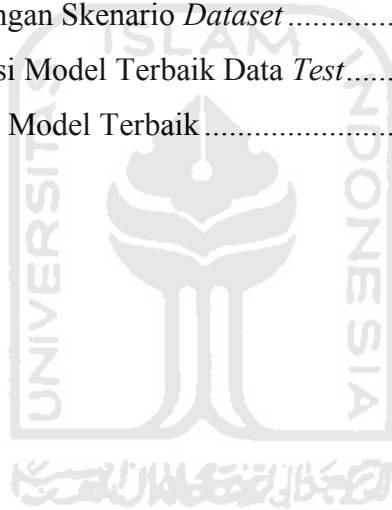


5.4	Skenario Data <i>Training</i> Dan <i>Testing</i> .....	43
5.5	Pembuatan Model CNN .....	44
5.5.1	Membandingkan Nila <i>Epoch</i> .....	45
5.5.2	Membandingkan Jenis <i>Optimizer</i> .....	45
5.5.3	Membandingkan Skenario <i>Dataset</i> .....	46
5.6	Hasil Klasifikasi Model Terbaik .....	50
BAB 6	PENUTUP .....	52
6.1	Kesimpulan .....	52
6.2	Saran .....	52
	DAFTAR PUSTAKA .....	53
	LAMPIRAN .....	57



## DAFTAR TABEL

<b>Tabel 2.1.</b> Penelitian Menggunakan Metode CNN.....	6
<b>Tabel 2.2.</b> Penelitian Klasifikasi Penyakit Tanaman Padi .....	9
<b>Tabel 2.3.</b> Perbedaan Penelitian Sebelumnya Dengan Penelitian Penulis .....	12
<b>Tabel 4.1.</b> Definisi Operasional Variabel.....	38
<b>Tabel 5.1.</b> Skenario Pembagian Data <i>Training</i> Dan <i>Testing</i> .....	43
<b>Tabel 5.2.</b> Hasil Perbandingan Nilai <i>Epoch</i> .....	45
<b>Tabel 5.3.</b> Hasil Perbandingan Jenis <i>Optimizer</i> .....	46
<b>Tabel 5.3.</b> Hasil Perbandingan Skenario <i>Dataset</i> .....	47
<b>Tabel 5.5.</b> Hasil Klasifikasi Model Terbaik Data <i>Test</i> .....	50
<b>Tabel 5.6.</b> <i>Output</i> Akurasi Model Terbaik.....	50



## DAFTAR GAMBAR

<b>Gambar 3.1.</b> Citra Jenis Penyakit <i>Bacterial Leaf Blight</i> .....	14
<b>Gambar 3.2.</b> Citra Jenis Penyakit <i>Brown Spot</i> .....	15
<b>Gambar 3.3.</b> Citra Jenis Penyakit <i>Leaf Spot</i> .....	16
<b>Gambar 3.4.</b> <i>Layer-Layer</i> Yang Terdapat Di Dalam <i>Deep Learning</i> .....	20
<b>Gambar 3.5.</b> Ilustrasi Arsitektur CNN .....	21
<b>Gambar 3.6.</b> Proses <i>Convolution Operation</i> .....	22
<b>Gambar 3.7.</b> Fungsi Aktivasi Pada ReLu .....	24
<b>Gambar 3.8.</b> Fungsi Aktivasi Sigmoid .....	24
<b>Gambar 3.9.</b> Fungsi Aktivasi Tanh .....	25
<b>Gambar 3.10.</b> Contoh Operasi <i>Max Pooling</i> .....	26
<b>Gambar 3.11.</b> Citra Warna Dengan Komponen RGB.....	35
<b>Gambar 4.1.</b> Diagram Tahapan Penelitian .....	40
<b>Gambar 5.1.</b> Penyimpanan Citra Dalam Satu Folder .....	41
<b>Gambar 5.2.</b> Citra Sebelum dan Sesudah <i>Resize</i> .....	42
<b>Gambar 5.3.</b> <i>Output</i> Pelabelan Data Dengan Skenario Data <i>Training</i> 80% .....	43
<b>Gambar 5.4.</b> Skema Arsitektur CNN .....	47
<b>Gambar 5.5.</b> <i>Output Summary</i> Model Terbaik .....	49

## DAFTAR LAMPIRAN

<b>Lampiran 1</b> Data Citra Penyakit Daun Padi.....	57
<b>Lampiran 2</b> <i>Syntax</i> Program R Studio.....	59
<b>Lampiran 3</b> Hasil Klasifikasi.....	64



## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 14 Oktober 2020



(Siti Rahmah Danur Amiril)



## ABSTRAK

### IMPLEMENTASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* PADA KLASIFIKASI PENYAKIT PADI MELALUI CITRA DAUN (Studi Kasus: Penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*)

Siti Rahmah Danur Amiril

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Padi atau nama lain dalam bahasa latin nya adalah *Oryza Sativa* merupakan salah satu tanaman budi daya terpenting didalam peradaban khususnya di negara Indonesia. Di negara ini padi merupakan komoditas utama bagi kehidupan dalam pemenuhan kebutuhan bahan pokok pangan. Menurut data dari Badan Pusat Statistik (BPS) pada tahun 2020 terkait jumlah penduduk Indonesia saat ini yang mencapai 269.600.000 jiwa, secara tidak langsung membuat kebutuhan akan tanaman ini juga semakin meningkat. Proses penanaman pada padi biasanya dihadapkan dengan berbagai kendala baik dari luar (hama) maupun kendala dari dalam (penyakit). Penyakit yang paling sering menyerang tanaman padi di wilayah Indonesia ini adalah *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*. Oleh sebab itu, penelitian ini bertujuan untuk mengklasifikasi penyakit padi berdasarkan citra daunnya yang telah terserang penyakit menggunakan teknik *Deep Learning* dengan metode *Convolutional Neural Network* (CNN), untuk memudahkan dalam melakukan pendeteksian dini terhadap penyakit padi agar tidak menular dan menjadi semakin parah. Metode CNN mampu menghasilkan tingkat akurasi yang signifikan karena memiliki kedalaman jaringan dan telah banyak di aplikasikan pada data gambar. Berdasarkan dari hasil klasifikasi yang dilakukan, didapatkan tingkat akurasi sebesar 91,7% dengan arsitektur terbaik menggunakan parameter skenario perbandingan dataset 90%, size 100x100 piksel, kernel 3x3, *learning rate* 0,01, *optimizer* Adam, *epoch* 150 dan *batch size* 30.

**Kata Kunci:** Padi, *Bacterial Leaf Blight*, *Brown Spot*, *Leaf Spot*, Klasifikasi, CNN

## **ABSTRACT**

### **IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK ALGORITHM ON RICE DISEASE CLASSIFICATION THROUGH LEAF IMAGE**

*(Case Study: Disease Bacterial Leaf Blight, Brown Spot and Leaf Spot)*

Siti Rahmah Danur Amiril

*Department of Statistics, Faculty of Mathematics and Natural Sciences*

*Islamic University of Indonesia*

*Rice or another name in Latin is *Oryza Sativa* which is one of the most important cultivated plants in civilization, especially in Indonesia. In this country, rice is the main commodity for life in meeting the needs of basic foodstuffs. According to data from the Badan Pusat Statistik (BPS) in 2020 regarding the current population of Indonesia, which reaches 269,600,000 people, this indirectly makes the need for this plant also increasing. The process of planting in rice is usually faced with various constraints both from outside (pests) and obstacles from within (disease). The diseases that most often attack rice plants in Indonesia are Bacterial Leaf Blight, Brown Spot and Leaf Spot. Therefore, this study aims to classify rice disease based on the image of its leaves that have been attacked by the disease using the technique Deep Learning with the method Convolutional Neural Network (CNN), to facilitate early detection of rice disease so that it is not contagious and gets worse. The CNN method is able to produce a significant level of accuracy because it has network depth and has been widely applied to image data. Based on the results of the classification carried out, an accuracy rate of 91.7% was obtained with the best architecture using a parameter comparison scenario of 90% dataset, size 100x100 pixels, 3x3 kernel, learning rate 0,01, optimizer Adam, epoch 150 and batch size 30.*

**Keywords:** *Rice, Bacterial Leaf Blight, Brown Spot, Leaf Spot, Classification, CNN*

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang Masalah

Padi atau dengan nama lain dalam bahasa latin nya adalah *Oryza Sativa* merupakan salah satu tanaman budi daya terpenting didalam peradaban khususnya di negara Indonesia. Di negara ini padi merupakan komoditas utama bagi kehidupan dalam pemenuhan kebutuhan bahan pokok pangan. Budidaya tanaman padi ini biasanya dilakukan oleh para petani. Padi mengandung karbohidrat sebesar 78.9%, protein 6.8%, lemak 0.7% dan lain-lain 0.6%. Menurut data dari Badan Pusat Statistik (BPS) pada tahun 2020 terkait jumlah penduduk Indonesia saat ini yang mencapai 269.600.000 jiwa, secara tidak langsung membuat kebutuhan akan tanaman ini juga semakin meningkat (Pratiwi, 2016).

Proses penanaman pada padi biasanya dihadapkan dengan berbagai kendala baik dari luar (hama) maupun kendala dari dalam (penyakit). Kendala dari luar ini biasanya disebabkan oleh hama seperti tikus, wareng coklat, penggerek batang, ganjur, keong mas, dsb. Sedangkan kendala dari dalam biasanya disebabkan oleh penyakit pada tanaman padi itu sendiri seperti, hawar daun bakteri, bercak coklat, noda palsu, blas, hawar pelepah, busuk batang, bakteri daun bergaris, dan beberapa penyakit virus seperti kerdil rumput, kerdil hampa, tungro, dsb. Kedua kendala di atas sangat penting untuk diketahui dan di amati karena penyebarannya yang sangat luas diberbagai wilayah dan sangat umum terjadi di ekosistem tropika seperti di Indonesia ini yang mengakibatkan penurunan hasil produksi yang sangat besar (Suparyono, 2007).

Negara dengan tingkat jumlah konsumsi tanaman padi tertinggi di dunia adalah Indonesia. bahkan di tingkat Asia, Indonesia menduduki peringkat pertama dan mengalahkan negara Korea, Jepang, Malaysia dan Thailand yang juga tingkat konsumsi tanaman padi nya cukup tinggi. Eksistensi tanaman padi sebagai bahan makanan pokok bagi hampir seluruh bangsa Indonesia harus tetap terjaga kestabilan nya. Di prediksi oleh Badan Pusat Statistik (BPS) bahwa pada tahun 2015 angka produksi padi adalah sebanyak 74,99 juta ton gabah kering giling (GKG) yang berarti mengalami peningkatan sebesar 4,15 juta ton atau sebesar 5,85%



dibandingkan dengan tahun sebelumnya yaitu tahun 2014. Jika prediksi BPS tersebut benar maka dapat dipastikan jumlah produksi padi pada tahun 2015 merupakan yang tertinggi dalam periode 10 tahun terakhir. Berdasarkan dari publikasi BPS mengenai produksi tanaman pangan diketahui bahwa produksi padi pada tahun 2014 di Indonesia adalah sebesar 70,85 juta ton GKG dan mengalami penurunan sebesar 433,24 ribu ton atau sebesar 0,61% dibanding dengan tahun 2013. Penurunan tersebut diketahui terjadi di wilayah pulau Jawa yaitu sebesar 829,97 ribu ton sedangkan di luar wilayah pulau Jawa mengalami peningkatan produksi tanaman padi sebesar 396,73 ribu ton. Berkurangnya tingkat produksi terjadi karena adanya penurunan luas panen sebesar 37.95 ribu hektar atau sebesar 0.27% dan terjadi penurunan produktivitas sebesar 0.17 kuintal/hektar atau sebesar 0.33%. Adapun wilayah dengan jumlah penurunan produksi padi yang cukup besar pada tahun 2014 adalah Provinsi Jawa Barat, Jawa Tengah, Sumatera Utara, Aceh dan Nusa Tenggara Barat. Indonesia bahkan pernah tercatat sebagai salah satu penghasil produksi padi terbesar di antara negara-negara lain, namun pada saat ini seperti yang kita ketahui bahwa Indonesia setiap tahunnya mengimpor beras dari luar negeri untuk memenuhi kebutuhan domestik. Berdasarkan data BPS impor beras Indonesia dari negara Thailand pada periode Januari-Oktober 2018 mencapai 780 ribu ton setara dengan 36.45% dari total impor beras yang mencapai 2.14 juta ton sepanjang tahun 2018 dan tercatat sebagai volume impor terbesar ketiga sejak tahun 1999. Provinsi Jawa Timur adalah penyumbang terbesar bagi produksi padi di negara Indonesia. Pada tahun 2014 diketahui bahwa produksi padi di Jawa Timur mencapai nilai 12.397.049 juta ton atau sebesar 17.5% dari jumlah total produksi tanaman padi di Indonesia dan mengalami kenaikan sebesar 347.707 ribu ton dibandingkan dengan tahun 2013 (Ishaq, Rumiati, & Permatasari, 2017).

Penyakit tanaman terutama pada padi dapat memberikan dampak yang sangat besar bagi kehidupan umat manusia bahkan dapat menyebabkan kematian. Tercatat pada tahun 1940an kurang lebih sekitar dua juta penduduk di Bangladesh mati akibat kelaparan yang disebabkan oleh tanaman padi yang merupakan panganan pokok terjangkit jamur *Helminthosporium oryzae* (*Brown Spot*) dan hingga saat ini masih banyak kasus kelaparan di berbagai negara yang disebabkan karena tanaman

penghasil pangan di negara tersebut mengalami gagal panen akibat penyakit dengan frekuensi yang cukup tinggi.

Permasalahan dan kendala yang dihadapi dalam budidaya penanaman padi ini semakin beragam, dimana meliputi permasalahan konversi lahan sawah dan perubahan iklim secara global yang juga memberikan dampak anomali kepada pertumbuhan dan perkembangan hama serta penyakit yang sangat mengancam jumlah produksi padi. Hal tersebut tentunya mengakibatkan keuntungan dari segi pertanian juga ikut menurun. Menurut Semangun (2008), bagian organ tertentu atau bahkan seluruh organ tanaman juga dapat rusak disebabkan oleh penyakit. Oleh karena itu, dalam pengelolaan penyakit tanaman hal yang paling penting adalah menjaga kestabilan pangan karena penyakit pada tanaman dapat terus berkembang secara terus-menerus dan mengancam pertumbuhan bahkan mengakibatkan gagal panen.

Penyakit tanaman padi yang cukup penting dan sering terjadi di wilayah Indonesia ini adalah hawar daun bakteri (*Bacterial Leaf Blight*) yang disebabkan oleh bakteri *Xanthomonas campestris* pv. *Oryzae*, bercak coklat (*Brown Spot*) yang disebabkan oleh jamur *Helminthosporium oryzae*, penyakit bercak bergaris/ sempit (*Narrow Brown Leaf Spot/Leaf Spot*) yang disebabkan oleh jamur *Cercospora oryzae*, tungro yang disebabkan oleh virus tungro, bercak daun *pyricularia* yang disebabkan oleh jamur *Pyricularia grisea*, hawar pelepah daun yang disebabkan oleh jamur *Rhizoctania solani*, kerdil hampa yang disebabkan oleh virus *Ragged stunt* dan kerdil rumput yang disebabkan oleh virus *Grassy stunt*. Kehilangan hasil padi akibat gangguan hawar daun bakteri adalah sekitar 15-24% (Sutarman, 2017).

Oleh sebab itu, untuk memudahkan dalam pendeteksian dini terhadap penyakit padi khususnya bagi para petani dan bagi kepentingan dalam industri pertanian agar penyakit tidak menular dan semakin parah apalagi dewasa ini *artificial intelligence* atau kecerdasan buatan sedang jadi topik yang sering diperbincangkan di beberapa tahun belakangan ini. Sektor industri pertanian adalah salah satu sektor yang dapat mengimplementasikan kecerdasan buatan. Sehingga dengan menggunakan teknologi ini dapat mengetahui penyakit padi melalui citra daunnya dengan membaginya ke dalam beberapa kategori. Sehingga dibutuhkan

kemampuan dalam pengolahan klasifikasi citra, yang memiliki kemampuan dalam pengolahan citra ini salah satu nya adalah teknik *deep learning* dengan metode *Convolutional Neural Network* (CNN). Metode CNN mampu menghasilkan tingkat akurasi yang signifikan karena memiliki kedalaman jaringan dan telah banyak di aplikasikan pada data gambar.

Oleh karena itu peneliti menggunakan metode CNN dalam penelitian ini untuk mengklasifikasi gambar daun padi berdasarkan dari jenis penyakitnya. Adapun jenis penyakit daun padi yang akan diklasifikasikan adalah sebanyak tiga kategori yaitu *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut permasalahan yang di angkat dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana implementasi algoritma CNN terbaik yang mampu mengklasifikasikan jenis penyakit daun padi berdasarkan citra daunnya?
- b. Bagaimana tingkat akurasi yang diperoleh dari proses pengklasifikasian citra penyakit daun padi dengan menggunakan CNN?

## 1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

- a. Data yang digunakan adalah data citra penyakit dari genus *Oryza Sativa* sebanyak 40 citra dari setiap jenis penyakit pada daun padi.
- b. Citra yang digunakan memiliki ukuran *pixel*.
- c. Klasifikasi yang dilakukan berdasarkan 3 jenis penyakit daun padi yaitu *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*.
- d. *Software* yang digunakan dalam pengimplementasian *deep learning* dalam penelitian ini adalah R Studio versi 3.6.2.
- e. *Packages* yang digunakan dalam analisis dengan metode CNN adalah Keras, *EImage* dan *tensorflow* pada *software* R Studio.

#### **1.4 Tujuan Penelitian**

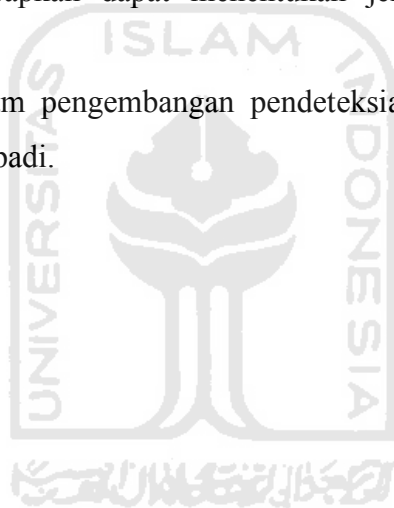
Tujuan dari penelitian ini adalah sebagai berikut:

- a. Mengetahui implementasi dan arsitektur terbaik yang mampu mengklasifikasikan jenis penyakit daun padi berdasarkan citra daunnya dengan metode *deep learning* menggunakan CNN.
- b. Mengetahui hasil akurasi yang diperoleh dari proses pengklasifikasian citra penyakit daun padi menggunakan CNN.

#### **1.5 Manfaat Penelitian**

Adapun manfaat yang diharapkan dari penelitian ini adalah:

- a. Penelitian ini diharapkan dapat menentukan jenis penyakit daun padi berdasarkan citra.
- b. Menjadi acuan dalam pengembangan pendeteksian secara otomatis jenis penyakit pada daun padi.



## BAB 2 TINJAUAN PUSTAKA

### 2.1 Penelitian Menggunakan Metode *Convolutional Neural Network*

Referensi dari penelitian-penelitian terdahulu dengan menggunakan metode CNN untuk mengklasifikasi dengan menggunakan citra gambar pada objek penelitiannya dirangkum dalam tabel dibawah ini:

**Tabel 2.1.** Penelitian Menggunakan Metode CNN

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
1.	Kusumaningrum, 2018	Implementasi <i>Convolutional Neural Network</i> (CNN) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras	2018	Penelitian ini melakukan klasifikasi terhadap gambar jamur konsumsi dengan mengklasifikasikan 3 jenis jamur menggunakan metode CNN dan diperoleh akurasi sebesar 100%. Peneliti menggunakan skenario data <i>train</i> dan <i>testing</i> sebesar 80% : 20%, dan didapatkan hasil prediksi yaitu model mampu dengan tepat mengklasifikasi semua citra dengan tepat.
2.	Shafira, 2018	Implementasi <i>Convolutional Neural Networks</i> Untuk Klasifikasi Citra Tomat Menggunakan Keras	2018	Penelitian ini melakukan pengklasifikasian citra tomat menjadi dua kelas yaitu citra tomat bagus (layak) dan busuk (tidak layak) dengan skenario data <i>training</i> dan <i>testing</i> yaitu 80% dan 20%. Tingkat akurasi yang didapatkan dari data <i>testing</i> pada model yang terbentuk yaitu sebesar 90%. Hasil klasifikasi dari sepuluh

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
				data baru citra tomat dalam pengujian model yang terbentuk adalah semua citra berhasil diklasifikasi secara tepat.
3.	Hidayat & Hermawan, 2018	Deteksi Hama Pada Daun Teh Dengan Metode <i>Convolutional Neural Network</i> (CNN)	2018	Pada penelitian ini dilakukan klasifikasi berdasarkan dua jenis hama yaitu <i>helloweltis</i> dan <i>blister blight</i> pada citra daun teh. Dari hasil pengujian dengan menggunakan metode CNN didapatkan tingkat akurasi rata-rata sebesar 95% dengan epoch = 20. Berdasarkan dari hasil yang didapatkan tersebut dapat disimpulkan bahwa metode <i>convolutional neural network</i> (CNN) dapat diimplementasikan dalam pendeteksian pengenalan jenis hama pada daun the.
4.	Wicaksono, Andrayana, & Benrahman, 2020	Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode <i>Convolutional Neural Network</i>	2020	Penelitian ini mengimplementasikan teknik <i>deep learning</i> dengan menggunakan metode CNN untuk mendeteksi jenis penyakit pada daun tanaman apel. Pengklasifikasian dilakukan berdasarkan kelas daun tanaman apel yang sehat dan daun yang terkena penyakit sebanyak tiga kelas yaitu <i>apple scab</i> , <i>black rot</i> dan

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
				<p><i>cedar-apple rust</i> yang mana ketiga penyakit pada tanaman daun apel tersebut disebabkan oleh jamur. Skenario yang digunakan dalam pembagian data <i>training</i> dan <i>testing</i> adalah 80% dan 20%. Berdasarkan dari pengujian yang dilakukan pada data <i>training</i> untuk epoch 50, 75 dan 100 didapatkan rata-rata tingkat akurasi pada model sebesar 99,2% dan rata-rata tingkat akurasi pada data validasi sebesar 94,9%, maka dapat dikatakan metode ini bekerja dengan baik dalam mendeteksi penyakit apel dengan menggunakan citra daun apel.</p>
5.	Felix, Faisal, Butarbutar, & Sirait, 2019	Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun	2019	<p>Penelitian ini mengimplementasikan metode CNN dan SVM dalam melakukan klasifikasi terhadap indentifikasi penyakit tomat melalui citra daun. Citra daun tomat dibagi kedalam empat kelas berdasarkan jenis penyakitnya yaitu <i>septorial leaf spot</i>, <i>tomato mosaic</i>, <i>tomato yellow leaf curl</i> dan daun sehat. Data <i>training</i> dan <i>testing</i> dibagi menjadi 80% dan 20%.</p>

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
				Berdasarkan dari hasil pengujian didapatkan rata-rata tingkat akurasi dengan menggunakan metode SVM adalah sebesar 95% sedangkan pada metode CNN didapatkan rata-rata tingkat akurasi sebesar 97,5%. Perbandingan tersebut menunjukkan bahwa metode CNN merupakan <i>classifier</i> yang lebih baik digunakan dalam pendeteksian penyakit pada citra daun tomat.

## 2.2 Penelitian Tentang Klasifikasi Penyakit Tanaman Padi

Berikut dirangkum penelitian terdahulu mengenai pengklasifikasian penyakit pada daun padi dengan menggunakan beberapa teknik dan metode yang berbeda sebagai referensi dari objek penelitian:

**Tabel 2.2.** Penelitian Klasifikasi Penyakit Tanaman Padi

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
1.	Sudir, Nuryanto, & Kadir, 2012	Epidemiologi, Patotipe, dan Strategi Pengendalian Penyakit Hawar Daun Bakteri pada Tanaman Padi	2012	Penyakit hawar daun bakteri (HDB) merupakan salah satu penyakit penting tanaman padi di negara- negara penghasil padi, termasuk di Indonesia. Penyakit ini disebabkan oleh bakteri <i>Xanthomonas oryzae</i> pv. <i>oryzae</i> ( <i>Xoo</i> ). Patogen ini menginfeksi daun padi pada semua fase pertumbuhan tanaman, mulai dari pesemaian



No	Nama Peneliti	Judul Penelitian	Tahun	Isi
				sampai menjelang panen. Gejala yang timbul pada tanaman fase vegetatif disebut kresak dan pada fase generatif disebut hawar. Apabila infeksi terjadi pada fase generatif mengakibatkan proses pengisian gabah menjadi kurang sempurna. Kehilangan hasil karena penyakit HDB bervariasi antara 15– 80%, bergantung pada stadia tanaman saat penyakit timbul
2.	Kusanti & Haris, 2018	Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut	2018	Hasil pengujian menunjukkan akurasi untuk mengidentifikasi penyakit pada citra daun tanaman padi mencapai 80%. Metode <i>backpropagation</i> , dapat digunakan untuk melakukan klasifikasi fitur penyakit pada citra daun tanaman padi.
3.	Sahbana, 2017	Identifikasi Jenis Penyakit Daun Padi Menggunakan <i>Adaptif Neuro Fuzzy Inferene System</i> (ANFIS) Berdasarkan Tekstur	2019	Kemampuan pengenalan (akurasi) ANFIS terhadap penyakit daun padi jenis Bakterial leaf blight dan brown spot sebesar 100%, sedangkan kemampuan pengenalan (akurasi) terhadap penyakit daun padi jenis leaf smut sebesar 87,5%, sehingga kemampuan pengenalan (akurasi) rata-rata adalah 98,5%. Parameter penduga

No	Nama Peneliti	Judul Penelitian	Tahun	Isi
				yang digunakan adalah energi, kontras, homogenitas dan entropi sebagai penciri tekstur tanpa dilakukan pengolahan citra berupa perbaikan kualitas citra maupun segmentasi citra.
4.	Sahbana, 2017	Implementasi Sistem Pakar Dalam Mengidentifikasi Penyakit Tanaman Padi Menggunakan Metode <i>Certainty Factor</i>	2017	Penelitian ini mengembangkan sebuah aplikasi yang dapat mendiagnosis penyakit pada tanaman padi dengan menggunakan metode <i>certainty factor</i> . Fitur yang terdapat pada aplikasi tersebut yakni diagnosis, riwayat diagnosis, daftar penyakit, tambah dan edit data, rekap diagnosis serta panduan.
5.	Ihsan, Agus, & Khairina, 2017	Penerapan Metode <i>Dempster Shafer</i> Untuk Sistem Deteksi Penyakit Tanaman Padi	2017	Penelitian ini mengklasifikasikan penyakit pada tanaman padi dengan menggunakan metode <i>dempster shafer</i> . Metode ini juga termasuk kedalam teknik sistem pakar. Metode <i>dempster shafer</i> berhasil di implemmentasikan untuk mendiagnosa jenis penyakit pada tanaman padi yaitu dengan mendiagnosa jenis-jenis penyakit dengan mendeteksi gejala-gejala yang muncul pada tanaman.

### 2.3 Perbedaan Dengan Penelitian Sebelumnya

Kemudian setelah memahami dan mempelajari penelitian-penelitian terdahulu yang berkaitan dengan metode dan objek pada penelitian ini seperti pada tabel di atas, berikut dapat diketahui perbedaan antara penelitian ini dengan penelitian terdahulunya yang dirangkum dan disajikan pada tabel dibawah ini:

**Tabel 2.3.** Perbedaan Penelitian Sebelumnya Dengan Penelitian Penulis

Nama	Judul Penelitian	Tahun	Isi
Siti Rahmah Danur Amiril	Implementasi Algoritma <i>Convolutional Neural Network</i> Dalam Klasifikasi Penyakit Padi Melalui Citra Daun	2020	Penelitian ini menggunakan dataset citra daun padi yang terserang penyakit, masing-masing tiap jenis penyakit memiliki dataset sebanyak empat puluh citra dan terbagi menjadi tiga jenis penyakit daun padi yaitu <i>Bacterial Leaf Blight</i> , <i>Brown Spot</i> dan <i>Leaf Spot</i> . Teknik yang digunakan dalam penelitian ini adalah <i>deep learning</i> dengan metode <i>Convolutional Neural Network</i> (CNN) dan dilakukan perbandingan terhadap beberapa parameter untuk mengetahui model yang memberikan tingkat akurasi yang terbaik dalam melakukan pengklasifikasian jenis penyakit dengan tepat.

## BAB 3 LANDASAN TEORI

### 3.1 Padi

Padi merupakan tanaman pangan dunia yang dibudidayakan di lebih dari 100 negara, tanaman padi menghasilkan produksi sebanyak 600 juta ton/tahun dengan luas lahan 153 juta hektar. Sebagian besar tanaman padi dunia dibudidayakan di Asia Tenggara, Asia Selatan, Asia Timur dan sebagian di Asia dan Sub Sahara Afrika dengan persentase sekitar 90% yang berarti hampir keseluruhan budidaya tanaman padi dilakukan di Asia dengan kisaran jumlah produksi adalah antara 1 ton/hektar pada daerah kering hingga 10 ton/hektar di daerah dengan sistem irigasi intensif. Tanaman padi menghasilkan biji yang disebut sebagai gabah yang mana setelah digiling kulit biji terkelupas dan meninggalkan beras (*cotyledon*). Beras inilah yang merupakan bahan pangan makanan pokok bagi bangsa Indonesia (Agus, 2019).

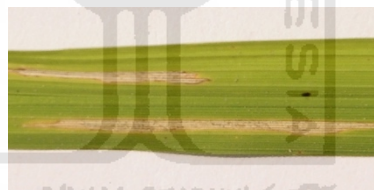
Padi adalah tanaman yang kaya akan karbohidrat dan merupakan tanaman pangan utama di dunia, oleh karena itu sebagian besar masyarakat di dunia menjadikan padi sebagai makanan pokok. Berdasarkan dari klasifikasinya padi termasuk kedalam golongan familia rumput-rumputan (*poaceae*) dengan nama genus *oryza* dan juga termasuk jenis tanaman biji-bijian (*serealia*). Di negara Indonesia padi dapat tumbuh sepanjang musim. China dan India adalah negara yang menjadi sentra produksi padi dengan persentase sebesar 35% dan 20% dari total produksi padi di dunia (Afni, 2012).

Tanaman padi memiliki kemampuan beradaptasi yang baik sehingga membuat tanaman ini tersebar dari dataran rendah hingga ke dataran tinggi yaitu mulai dari 0 hingga 1.500 mdpl pada suhu 19°C sampai 27°C di tanah subur maupun tanah kritis, rawa-rawa hingga lahan kering serta pada daerah tropis sampai subtropis pada garis lintang 45° LU sampai 45° LS dengan intensitas penyinaran yang penuh tanpa adanya perlindungan atau naungan. Pada masa pertumbuhannya, tanaman padi terkhusus padi sawah membutuhkan air yang banyak. Curah hujan

yang dibutuhkan untuk pertumbuhan tanaman padi adalah sekitar 1.500 sampai 2.000 mm/tahun atau sekitar 200 mm/bulannya. Di daerah dengan irigasi yang cukup baik, tanaman padi ditanam sebanyak tiga kali dalam waktu satu tahun, sedangkan di daerah yang tidak memiliki irigasi maka perlu memperhatikan kondisi dari curah hujan tahunan (Utama M. H., 2019)

### 3.1.1 Penyakit Padi *Bacterial Leaf Blight*

Penyakit *Bacterial Leaf Blight* atau dalam Bahasa Indonesia disebut sebagai Hawar Daun Bakteri adalah jenis penyakit utama dan penting pada tanaman padi sawah pada negara-negara penghasil padi di dunia, termasuk di Negara Indonesia. Penyakit yang menyerang bagian daun pada padi ini disebabkan oleh bakteri *Xanthomonas campestris pv oryzae*. Penyebaran serta perkembangan dari patogen ini tergolong cepat dan juga sangat sulit untuk dikendalikan. Pada kondisi dengan kelembaban yang tinggi serangan akan meningkat sehingga sel bakteri akan bebas tersebar dengan larutnya embun di permukaan daun (Suryanto, 2010).



**Gambar 3.1.** Citra Jenis Penyakit *Bacterial Leaf Blight*  
Sumber: (Prajapati, 2019)

Ciri-ciri dan gejala dari penyakit ini menurut (Suryanto, 2010) antara lain adalah sebagai berikut:

- Ciri-ciri dari serangan penyakit dapat terlihat sejak tanaman mulai di semai hingga tanaman dewasa
- Terdapat bercak dengan warna kuning hingga putih ke abuan, bercak dapat ditemukan mulai dari salah satu atau kedua tepi daun, menjalar sampai ke bawah daun atau seluruh daun menjadi kering (layu)
- Daun terlihat layu seperti disiram air panas
- Pertumbuhan pada tanaman menjadi terhambat
- Bulir-bulir padi menjadi kurang bernas

- Serangan yang terjadi pada saat pembibitan menyebabkan bibit menjadi kering
- Kualitas beras menjadi rendah dan persentase beras yang pecah cukup tinggi

### 3.1.2 Penyakit Padi *Brown Spot*

Penyakit *Brown Spot* atau Bercak Coklat adalah jenis penyakit yang berbahaya bagi tanaman padi. Penyakit ini tersebar luas di negara-negara penghasil padi di benua Asia dan Afrika. Serangan dari penyakit ini dapat mengakibatkan kerugian mencapai hingga 40%. Penyakit ini selain dapat menyerang tanaman padi dewasa, juga dapat menyerang bibit tanaman padi dari benih yang telah terinfeksi. Permasalahan yang diakibatkan dari serangan penyakit ini adalah kematian bibit mencapai lebih dari 50%. Penyakit *Brown Spot* ini disebabkan oleh jamur *Helminthosporium oryzae* Breda de Haan yang dapat menyerang seluruh bagian tanaman padi seperti bibit, akar, batang, daun hingga bulir gabah. Penyebaran penyakit ini terjadi melalui proses spora yang terbang disebabkan oleh angin. Pada benih yang telah terinfeksi, jamur patogen ini memiliki kemampuan bertahan dalam kurun waktu yang cukup lama yaitu sekitar empat tahun. Pada kondisi tanah yang kekurangan unsur hara penyakit ini mampu berkembang dengan cukup baik terutama pada kelembaban nisbi yang tinggi (lebih dari 80%) dan perkembangannya pada suhu optimum yaitu 16-36°C (Utama, 2015).



**Gambar 3.2.** Citra Jenis Penyakit *Brown Spot*  
Sumber: (Prajapati, 2019)

Gejala penyakit ini muncul pada kulit gabah dan daun. Selain itu, juga tampak pada pelepah daun, cabang malai dan koleoptil. Pada tanaman padi yang terserang penyakit ini terlihat bercak dengan bentuk oval pada daun dengan persebaran yang merata, bercak yang tampak berwarna coklat dengan

titik tengah berwarna coklat atau hitam dimana bercak-bercak ini biasanya menutupi permukaan kulit. Pada tanaman yang kuat dari serangan penyakit ini, bercak-bercak menyebar dengan warna hijau keabu-abuan sedikit menggulung, kemudian menjadi kering berwarna abu keputih-putihan. Sedangkan pada tanaman yang rentan, gejala tersebut berkembang sampai permukaan daun menjadi kering seluruhnya dan sering sampai ke pelepah daun. Pada pagi hari dengan cuaca lembab akan tampak bakteri di permukaan bercak berbentuk cairan dengan warna kuning. Kelembaban yang tinggi dan hujan juga disertai dengan angin adalah faktor pendukung untuk berkembangnya penyakit ini (Prasetyo, 2012).

### 3.1.3 Penyakit Padi *Leaf Spot*

Penyakit Bercak Daun Coklat Sempit atau *Narrow Brown Leaf Spot (Leaf Spot)* adalah jenis penyakit pada tanaman padi yang disebabkan oleh jamur *Cercospora oryzae* Miyake. Penyakit ini tersebar luas di negara penanam padi. Di negara Indonesia, penyakit ini berkembang dengan baik terutama di daerah dengan lahan yang sangat minim unsur kalium dan nitrogen. Penyakit ini menyerang bagian daun pada tanaman padi yang mengakibatkan fungsi fotosintesis menjadi terganggu. Jika penyakit ini menyerang pada fase generatif akan mengakibatkan pengisian gabah menjadi hampa atau kurang sempurna sehingga kualitas dan bobot gabah menjadi rendah.



**Gambar 3.3.** Citra Jenis Penyakit *Leaf Spot*

Sumber: (Prajapati, 2019)

Gejala penyakit ini tampak pada daun yaitu terdapat bercak-bercak sempit memanjang, sejajar dengan ibu tulang dan berwarna coklat kemerahan. Jumlah bercak akan meningkat pada saat tanaman membentuk anakan. Ketika serangan dari penyakit ini cukup berat, bercak-bercak akan

tampak pada upih daun, bunga dan juga batang. Pada saat tanaman padi mulai siap panen, gejala yang berat mulai tampak pada daun bendera. Gejala mulai terlihat dua sampai empat minggu setelah padi di pindah. Gejala paling berat akan tampak kurang lebih dalam waktu satu bulan sebelum panen (Bulelengkab.go.id, 2018).

### **3.2 Kecerdasan Buatan (*Artificial Intelligence*)**

*Artificial Intelligence* atau kecerdasan buatan adalah suatu rancangan kecerdasan yang ditambahkan kedalam suatu sistem yang dapat di atur dalam konteks ilmiah atau Intelegensi Artifisial yang didefinisikan sebagai kecerdasan ilmiah. Kecerdasan buatan dirancang dan diciptakan serta dimasukkan kedalam mesin (komputer) agar dapat melakukan pekerjaan dan bertindak seperti manusia dengan dibekali pengetahuan serta kemampuan menalar yang baik. AI ditujukan untuk mengetahui atau memodelkan proses berpikir seperti manusia dan mendesain mesin sehingga dapat menirukan perilaku manusia. Bagian utama yang paling penting dan dibutuhkan dalam AI adalah basis pengetahuan (*knowledge base*) serta motor inferensi (*inference engine*) dimana basis pengetahuan berisikan pemikiran, fakta beserta teori-teori sedangkan motor inferensi adalah sebuah kemampuan dalam melakukan penarikan kesimpulan dengan berdasarkan pengetahuan.

Para ahli setuju bahwa *Artificial Intelligence* atau biasa disingkat dengan AI merupakan sebuah proses berfikir manusia dan mempresentasikan proses tersebut kedalam mesin yang digunakan. AI memiliki kemampuan menyelesaikan sebuah masalah yang merupakan tolak ukur sebuah kecerdasan. mesin yang dapat menyelesaikan masalah secara rutin adalah sebuah mesin yang cerdas. Berikut beberapa alasan untuk memodelkan performa manusia dalam hal ini yaitu:

- Untuk melakukan pengujian terhadap teori psikologis dari performa manusia
- Untuk membuat sebuah komputer dapat memahami dan melakukan penalaran seperti manusia
- Untuk membuat manusia memahami penalaran sebuah mesin (komputer)
- Untuk mengeksplorasi pengetahuan apa saja yang dapat diambil dari manusia



Meskipun AI memiliki sebuah konotasi fiksi ilmiah yang kuat, AI membentuk cabang yang sangat penting dalam ilmu komputer yang berhubungan dengan perilaku, pembelajaran serta adaptasi yang cerdas pada sebuah mesin. Penelitian dengan menggunakan sistem AI biasanya menyangkut pembuatan sebuah mesin untuk mengotomatisasikan sebuah tugas yang dibutuhkan perilaku cerdas dalam pemecahan masalahnya. Contohnya adalah pengendalian, perencanaan, penjadwalan, kemampuan dalam menjawab sebuah diagnosa dan pertanyaan serta melakukan pengenalan tulisan tangan, suara dan wajah. Hal-hal tersebut telah menjadi sebuah disiplin ilmu sendiri yang berfokus pada penyediaan solusi dalam masalah kehidupan secara real atau nyata. Sistem AI untuk sekarang ini sangat banyak digunakan dalam bidang teknik, militer, obat-obatan dan ekonomi seperti yang telah dibangun pada beberapa aplikasi perangkat lunak komputer dan video *game*.

AI dibagi kedalam dua garis besar yaitu AI konvensional dan kecerdasan komputasional (*Computational Intelligence* atau disingkat CI). AI konvensional lebih banyak melibatkan metode yang diklasifikasikan sebagai pembelajaran mesin yang ditandai dengan formalisme dan analisis statistik serta dikenal sebagai AI simbolis, AI murni, AI logis dan AI cara lama (*Good Old Fashioned Artificial Intelligence* atau disingkat GOFAI). Berikut metode-metode nya:

1. *Bayesian network*
2. Pertimbangan berdasar kasus
3. AI berdasarkan tingkah laku: metode modular dalam pembentukan secara manual sistem AI
4. Sistem pakar: sebuah sistem pakar mampu memproses dalam jumlah besar informasi yang diketahui dan memberikan kesimpulan berdasar pada informasi tersebut.

Kecerdasan komputasional melibatkan pembelajaran atau pengembangan iteratif. Pembelajaran ini berdasar pada data empiris dan di asosiasikan dengan AI nonsymbolis, AI yang tidak teratur dan perhitungan lunak. Berikut metode pokoknya yaitu:

1. Sistem *Fuzzy*: Teknik pertimbangan dalam ketidakpastian, telah digunakan secara luas pada industri modern dan sistem kendali produk konsumen.
2. Jaringan saraf: sistem yang memiliki kemampuan melakukan pengenalan pola yang sangat kuat.
3. Komputasi evolusioner: menerapkan konsep yang terinspirasi secara biologis seperti populasi, mutasi dan *survival of the fittest* dalam menghasilkan pemecahan sebuah masalah yang lebih baik.

Metode di atas dibagi menjadi algoritme evolusioner dan kecerdasan berkelompok. Dengan sistem cerdas hibrid percobaan dibuat untuk menggabungkan dua kelompok tersebut. Aturan inferensi pakar dapat dibangkitkan dengan jaringan saraf atau aturan produksi dari pembelajaran statistik seperti pada ACT-R (Pradana, 2018).

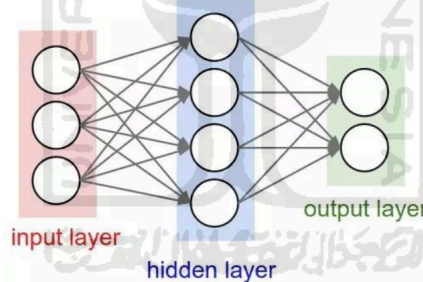
### 3.3 *Deep Learning*

*Deep Learning* adalah salah satu dari cabang ilmu *machine Learning* yang memiliki basis jaringan saraf tiruan yang mengajarkan sebuah mesin atau komputer dalam melakukan tindakan layaknya seperti manusia, seperti belajar dari sebuah contoh yang kemudian pembelajaran tersebut akan diterapkan dalam pengambilan sebuah tindakan. Didalam *deep learning* sebuah komputer belajar untuk mengklasifikasi secara langsung dari teks, suara atau gambar.

*Deep Learning* merupakan sebuah teknik di dalam *Neural Network* yang menggunakan teknik tertentu seperti *Restricted Boltzman Machine* (RBM) untuk mempercepat proses dalam pembelajaran pada *Neural Network* yang menggunakan lapisan yang banyak. Lapisan yang terdapat pada *Deep Learning* terdiri dari tiga bagian yaitu *input layer*, *hidden layer* dan *output layer*. Pada *input layer* berisi node-node yang dimana masing-masing node menyimpan nilai masukan yang tidak berubah di fase latihan dan hanya berubah ketika diberikan nilai masukan baru. Kemudian pada *hidden layer* dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat dalam meminimalisir *error* pada *output*. Pada *output layer* yang berfungsi menampilkan hasil dari perhitungan sistem oleh fungsi aktivasi di lapisan *hidden layer* yang berdasar pada *input* yang diterima. Semakin

banyak lapisan yang ditambahkan maka dapat menjadikan model pembelajaran mewakili citra berlabel dengan lebih baik. Aplikasi konsep jaringan saraf tiruan yang lebih dalam atau yang memiliki banyak lapisan dapat ditanggung pada algoritma *Machine Learning* yang telah ada sehingga komputer dapat belajar dengan skala besar, akurasi dan kecepatan. Prinsip tersebut saat ini semakin berkembang sehingga *deep learning* semakin sering digunakan oleh komunitas industri dan riset dalam penyelesaian masalah data yang besar seperti *Speech Recognition*, *Computer Vision* dan *Neural Language Processing*.

Dapat dilihat pada **Gambar 3.4** di bawah ini yang merupakan arsitektur yang biasa disebut sebagai *Multi Layer Perceptron* (MLP). Pada arsitektur pertama memiliki tiga buah neuron pada *input layer* dan memiliki dua buah node pada *output layer*. Di antara *input* dan *output* terdapat satu *hidden layer* yang memiliki empat buah neuron. Neuron-neuron tersebut akan langsung terhubung dengan neuron yang lain pada *layer* berikutnya.



**Gambar 3.4.** *Layer-Layer* Yang Terdapat Di Dalam *Deep Learning*  
 Sumber: (Pham, 2018)

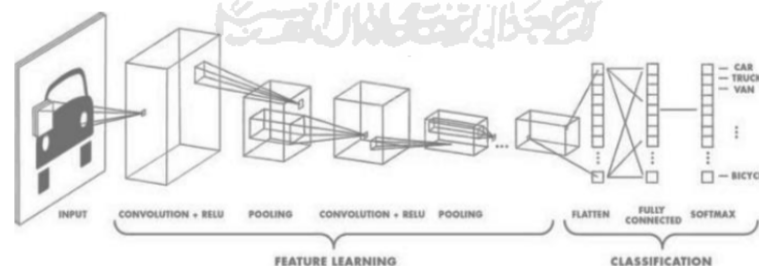
*Deep Learning* memberikan kemungkinan model komputasi yang terdiri atas beberapa *processing layer* yang digunakan untuk mempelajari representasi dari data terhadap berbagai tingkat abstraksi. Metode ini telah memperbaiki *state-of-the-art* pada pengenalan objek visual (*visual object recognition*), suara (*speech recognition*) dan deteksi objek (*object detection*) serta banyak lainnya. Berikut adalah beberapa algoritma yang menerapkan konsep dari *Deep Learning* yaitu antara lain *Deep Belief Network – Deep Neural Network* (DBN-DNN) untuk pengenalan suara, *Convolutional Neural Network* (CNN) untuk pengklasifikasian gambar, *Recurrent Neural Network* (RNN) untuk penerjemahan Bahasa, *Query-Oriented Deep Extraction* (QODE) dengan basis *Restricted Boltzman Machine*

(RBM) untuk mendeteksi *Drug Target Interaction* (DTI) dan *Deep Belief Network* (DBN) untuk prediksi data sesuai waktu (Kusumaningrum, 2018).

### 3.4 *Convolutional Neural Network* (CNN)

*Convolutional Neural Network* atau CNN adalah sebuah metode jaringan syaraf yang sangat terkenal dan cukup diminati. CNN digunakan untuk mengolah data dengan struktur grid yaitu salah satunya berupa citra dua dimensi dan juga mampu memproses data dengan dimensi tinggi seperti video. CNN bekerja mirip dengan *neural network standart*, hanya saja perbedaannya yaitu menggunakan kernel dua dimensi atau dimensi tinggi di setiap unit pada lapisan CNN yang akan dikonvolusi. Untuk melakukan penggabungan fitur dari spasial dengan bentuk spasial yang serupa media *input* maka digunakan kernel yang ada didalam CNN. Kemudian pada CNN untuk mengurangi jumlah dari variabel agar lebih mudah untuk dilakukan pembelajaran yaitu dengan menggunakan beberapa parameter.

Berdasarkan dari Namanya yaitu *convolutional neural network* yang menunjukkan bahwa metode jaringan tersebut menggunakan operasi dari matematika yang disebut sebagai konvolusi. CNN juga dilatih untuk mempelajari fitur dari objek untuk dapat memprediksinya. Berikut ilustrasi secara umum dari CNN pada gambaran arsitektur dibawah ini:



**Gambar 3.5.** Ilustrasi Arsitektur CNN  
Sumber : (Saha, 2018)

Dari **Gambar 3.5** di atas diketahui bahwa terdapat dua bagian lapisan arsitektur, yaitu:

a. *Feature Learning (Feature Extraction Layer)*

Pada *feature learning* terdapat lapisan yang digunakan untuk menerima masukkan (*input*) gambar secara langsung di awal dan mengolahnya hingga menghasilkan keluaran (*output*) data *multidimension array*.

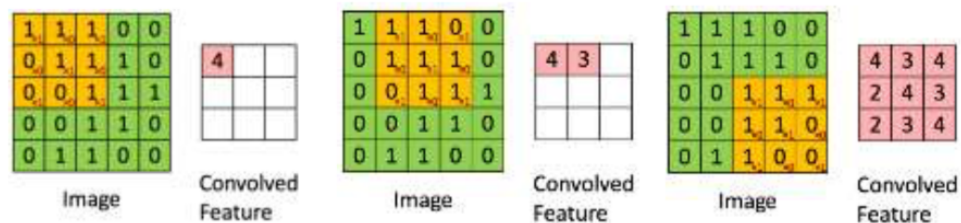
Lapisan yang terdapat dalam proses ini terdiri dari lapisan konvolusi dan lapisan *pooling*, yang mana pada masing-masing proses lapisan akan menghasilkan *feature maps* dalam bentuk angka-angka yang merepresentasikan gambar yang kemudian akan diteruskan ke bagian lapisan klasifikasi.

b. *Classification Layer* (Lapisan Klasifikasi)

Pada lapisan ini terdiri dari lapisan-lapisan yang memiliki neuron yang tersambung seluruhnya (*fully connected*) dengan lapisan yang lainnya. Lapisan ini menerima masukan (*input*) yang berasal dari *output layer* dari bagian *feature learning* yang kemudian akan di proses di *flatten* dengan menambahkan beberapa *hidden layer* pada *fullt connected* sehingga menghasilkan keluaran berupa nilai akurasi dari klasifikasi setiap kelas.

**3.4.1 Convolutional Operation**

Lapisan konvolusi atau *convolution layer* merupakan jenis operasi linier khusus dan melakukan operasi konvolusi pada *output* dari lapisan sebelumnya. Lapisan tersebut adalah proses yang mendasari sebuah CNN. *Convolution layer* adalah lapisan utama yang sangat penting untuk digunakan. Konvolusi adalah sebuah istilah matematis yang pada proses pengolahan citra berarti mengaplikasikan sebuah kernel (kotak kuning) pada sebuah citra dan disemua *offset* yang memungkinkan seperti pada **Gambar 3.6** dibawah ini, sedangkan kotak dengan warna hijau secara keseluruhan adalah citra yang akan dilakukan konvolusi. Kernel bergerak dari sudut kiri atas ke arah kanan bawah sehingga hasil dari konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanan nya pada **Gambar 3.6**.



**Gambar 3.6.** Proses *Convolution Operation*

Tujuan dari sebuah konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input* dan kemudian akan menghasilkan transformasi linear dari data *input* sesuai dengan informasi spasial yang terdapat pada data. Bobot yang terdapat pada *layer* tersebut menspesifikasikan kernel konvolusi yang digunakan sehingga kernel konvolusi bisa dilatih berdasar dari *input* pada CNN (Kusumaningrum, 2018).

### 3.4.2 Fungsi Aktivasi

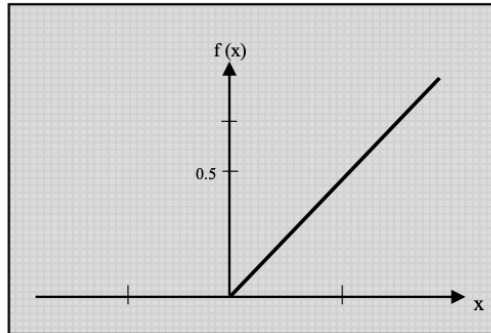
Fungsi aktivasi merupakan fungsi yang digunakan untuk menggambarkan keterkaitan atau hubungan tingkat aktivitas internal yang mungkin berbentuk linear maupun non-linear. Tahapan fungsi ini berada pada sebelum melakukan *pooling layer* dan setelah proses konvolusi. Di tahap ini nilai dari hasil konvolusi dikenakan fungsi aktivasi (*activation function*). Beberapa fungsi aktivasi yang paling sering digunakan dalam *convolutional network* yaitu diantaranya adalah ReLu (*Rectified Linear Unit*), Sigmoid dan Tanh (*Hyperbolic tangent function*). Fungsi aktivasi ReLu bagi beberapa peneliti sering menjadi pilihan dikarenakan sifatnya yang berfungsi dengan lebih baik (Pradana, 2018).

#### Fungsi Aktivasi ReLU (*Rectified Linier Unit*)

Fungsi aktivasi ReLu secara umum dinyatakan dalam persamaan (3.1) dibawah ini.

$$f(x) = \max(0, x) \quad (3.1)$$

Berdasarkan dari persamaan tersebut, nilai *output* dari neuron dapat dinyatakan sebagai 0 jika *inputnya* bernilai negatif. Jika nilai *input* dari fungsi aktivasi adalah bernilai positif maka *output* dari *neuron* adalah nilai *input* aktivasi itu sendiri. Fungsi aktivasi yang digunakan pada ReLu dapat dilihat pada **Gambar 3.7** dibawah ini.



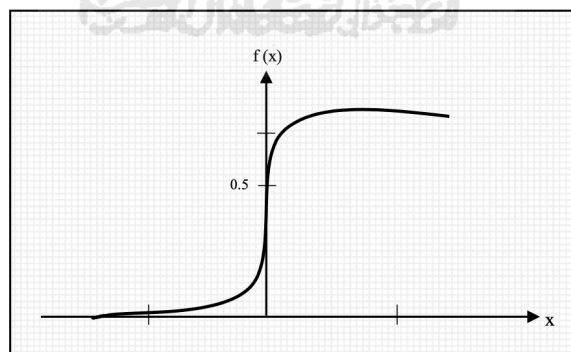
**Gambar 3.7.** Fungsi Aktivasi Pada ReLu

### Fungsi Sigmoid

Fungsi aktivasi sigmoid merupakan fungsi non-linear yang memiliki persamaan matematika sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.2)$$

Berdasarkan dari persamaan (3.2) tersebut, *input* untuk fungsi aktivasi tersebut adalah nilai *real* dan *output* dari fungsi tersebut bernilai antara 0 sampai 1. Jika *input*nya bernilai sangat negatif maka *output* yang didapatkan bernilai 0, jika *input*nya bernilai positif maka nilai *output* yang didapatkan bernilai 1. Nilai *input* dan *output* dari fungsi aktivasi sigmoid dapat dilihat dalam grafik pada **Gambar 3.8**.

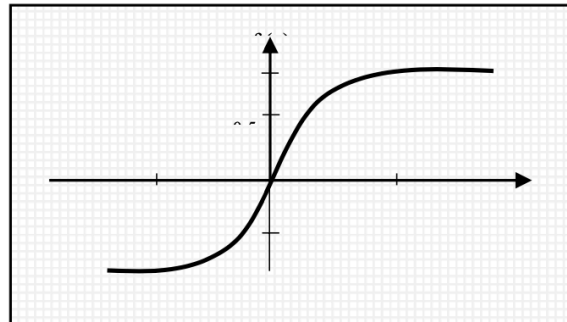


**Gambar 3.8.** Fungsi Aktivasi Sigmoid

Fungsi aktivasi sigmoid memiliki dua kelemahan utama yaitu yang pertama fungsi ini memiliki gradien mendekati nilai 0 (nol) jika nilai *input* sangat positif atau sangat negatif. hal ini tidak diinginkan karena nilai gradien digunakan pada proses pelatihan. Kelemahan yang kedua adalah bahwa fungsi ini tidak terpusat pada nilai 0 (*zero centered*).

### Fungsi Aktivasi Tanh

Fungsi aktivasi Tanh mengubah *input* dengan nilai *real* menjadi nilai dari -1 sampai 1 seperti yang ditunjukkan pada **Gambar 3.9** dibawah ini.



**Gambar 3.9.** Fungsi Aktivasi Tanh

**Gambar 3.9** menunjukkan bahwa nilai yang sangat negative diubah menjadi -1 dan nilai yang sangat positif diubah menjadi 1. Fungsi aktivasi Tanh merupakan fungsi dengan nilai yang terpusat pada nilai 0 sehingga sering digunakan dibanding dengan fungsi aktivasi Sigmoid. Nilai dari fungsi Tanh merupakan dua kali dari nilai fungsi aktivasi Sigmoid dikurangi satu dengan persamaan sebagai berikut:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

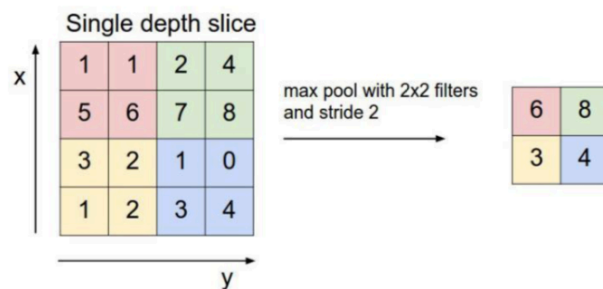
#### 3.4.3 Pooling Layer

*Pooling layer* adalah lapisan yang menggunakan fungsi dengan *feature map* untuk masukannya dan mengolahnya dengan berbagai jenis operasi statistik berdasar pada nilai piksel yang terdekat. Biasanya pada model CNN lapisan *pooling* disisipkan secara teratur setelah beberapa lapisan konvolusi. Dalam arsitektur model CNN lapisan *pooling* yang dimasukkan secara berturut-turut di antara lapisan konvolusi dapat secara progresif mengurangi ukuran volume *output* pada *feature map* sehingga dapat mengurangi jumlah perhitungan di jaringan dan parameter serta untuk pengendalian *overfitting*.

Hal yang sangat penting pada pembuatan model CNN adalah dengan memilih banyak jenis dari lapisan *pooling* yang dapat menguntungkan kinerja dari model. Pada setiap tumpukan *feature map* lapisan *pooling* bekerja serta



mengurangi ukurannya. Bentuk yang paling umum dari lapisan *pooling* adalah dengan menggunakan filter 2x2 yang diaplikasikan dengan langkah sebanyak dua dan kemudian beroperasi di setiap irisan dari *input*. Bentuk ini dapat mengurangi *feature map* hingga 75% dari ukuran aslinya.



**Gambar 3.10.** Contoh Operasi *Max Pooling*  
 Sumber : (Harjoseputro, Handarkho, & Adie, 2019)

Lapisan *pooling* akan berjalan di setiap irisan kedalaman volume *input* dengan cara bergantian. Pada **Gambar 3.10** di atas dapat dilihat bahwa lapisan *pooling* menggunakan salah satu operasi maksimal yang merupakan operasi yang paling umum. **Gambar 3.10** menampilkan operasi dengan langkah 2 dan dengan ukuran *filter* 2x2. Dari ukuran *input* 4x4, masing-masing empat angka pada *input* operasi mengambil nilai maksimalnya dan membentuk ukuran *output* baru menjadi 2x2. Pada CNN penggunaan *pooling layer* bertujuan untuk mereduksi ukuran gambar sehingga dapat secara mudah diganti dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan (Sumardi, 2019).

### 3.4.4 *Fully Connected Layer*

*Fully connected layer* merupakan suatu lapisan yang terbuat dari kumpulan hasil dari proses konvolusi. Setelah melewati proses dari konvolusi, maka kemudian sebuah citra akan menjadi nilai *input* pada *fully connected layer*. Pada lapisan ini terdapat *input* dan proses sebelumnya untuk menentukan fitur yang manakah yang berhubungan atau memiliki korelasi dengan kelas tertentu. Selain itu fungsi dari *fully connected layer* adalah untuk menyatukan semua node menjadi kedalam satu dimensi. Secara umum proses dari *fully connected layer* terletak di akhir arsitektur yang mana prosesnya direpresentasikan sebagai perkalian dari matriks sederhana yang

diikuti dengan penambahan vektor bias dan menerapkan fungsi non-linear berikut:

$$y = f(W^T x + b) \quad (3.4)$$

Dimana  $y$  adalah nilai vektor dari *output* yang berisi fungsi dari  $W$  yaitu matriks sederhana dengan bobot koneksi antar unit yang kemudian dikalikan dengan  $x$  sebagai sebuah vektor *input* lalu ditambah dengan  $b$  sebagai vektor bias (Khan, Rahmani, Shah, & Bennamoun, 2018).

### 3.4.5 *Backpropagation*

*Backpropagation* adalah algoritma pembelajaran yang diawasi dan biasanya digunakan pada *perceptron* yang memiliki banyak lapisan untuk mengubah bobot yang terhubung dengan *neuron* yang terdapat pada lapisan yang tersembunyi. Algoritma ini menggunakan. Algoritma *backpropagation* menggunakan *error* dari *output* untuk mengubah nilai bobot dalam arah mundur (*backward*). Untuk mendapatkan *error* tersebut, terdapat 45 tahap perambatan maju (*forward propagation*) yang harus dikerjakan terlebih dahulu. Ketika perambatan maju maka *neuron-neuron* diaktifkan menggunakan fungsi aktivasi yang dapat diturunkan seperti fungsi sigmoid. *Backpropagation* juga melakukan penghitungan ulang yang berguna untuk mengoreksi bias dan *error* agar dapat digunakan untuk memperbarui bobot dan meminimalkan adanya *error* pada *output* jaringan yang dihasilkan (Pradana, 2018).

### 3.4.6 *Learning Rate*

Untuk memberikan perubahan pada bobot secara cepat dengan memberikan penambahan atau mengurangi mengambil langkah yang terlalu besar yang dapat menghambat kemampuan untuk meminimalkan fungsi kerugian dikarenakan tidak ingin melakukan lompatan yang cukup besar sehingga melewati nilai optimal pada bobot yang diberikan.

Untuk memastikan hal itu tidak terjadi maka digunakan sebuah variable yang disebut sebagai *learning rate* atau tingkat pembelajaran. Biasanya hanya angka yang sangat kecil yaitu sekitar 0,001, yang kemudian digandakan oleh gradien kedalam skalanya. Hal tersebut memastikan bahwa

setiap perubahan yang diberikan pada bobot yang diberikan cukup kecil. Secara matematis jika mengambil langkah-langkah yang sangat besar bisa menyebabkan algoritma tidak akan bertemu secara optimal. Namun, disaat yang sama, kita tidak ingin untuk mengambil langkah yang terlalu kecil dikarenakan dapat memungkinkan tidak akan berakhir dengan nilai yang sesuai untuk bobot. Langkah-langkah yang sangat kecil bisa mengakibatkan pengoptimal mengonversi minimum local untuk fungsi kerugian, namun bukan minimum absolut. Sederhananya, harus diingat *learning rate* memastikan bahwa dengan memberikan perubahan bobot pada kecepatan yang tepat, maka tidak akan membuat perubahan yang terlalu besar ataupun terlalu kecil (Sumardi, 2019).

#### **3.4.7 Crossentropy Loss Function**

*Loss function* atau *cost function* adalah sebuah fungsi yang menampilkan kerugian yang berkaitan dengan seluruh kemungkinan yang dihasilkan dari model. *Loss function* bekerja saat model pembelajaran memberi kesalahan yang harus diperhatikan. *Loss function* dikatakan baik apabila fungsi yang menghasilkan *error* yang diharapkan paling rendah. Pada saat sebuah model memiliki kelas yang cukup banyak, maka diperlukan adanya sebuah cara untuk mengukur perbedaan antara probabilitas dari hasil hipotesis dan probabilitas kebenaran yang asli, dan pada saat pelatihan banyak algoritma yang mampu menyesuaikan parameter sehingga perbedaan tersebut diminimalkan. Gambaran secara umum dari algoritma ini adalah dengan meminimalkan kemungkinan log negatif dari dataset yang merupakan ukuran langsung dari performa prediksi model (Shafira, 2018).

#### **3.4.8 Softmax Classifier**

*Softmax classifier* merupakan suatu fungsi aktivasi yang digunakan dalam permasalahan klasifikasi yang biasanya fungsi ini digunakan pada *output layer*. Sebenarnya fungsi ini adalah sebuah probabilitas eksponensial yang dinormalisasi dari pengamatan kelas yang di wakili sebagai aktivasi neuron. Eksponensial berfungsi meningkatkan probabilitas nilai maksimum lapisan sebelumnya dibanding dengan nilai yang lainnya. Tujuan dari lapisan

ini adalah untuk memprediksi *output* dari klasifikasi ke dalam sebuah bentuk nilai probabilitas yang mana nilai probabilitas dari kelas terbesar adalah *output* prediksi kelas yang diperoleh. *Softmax* memberikan *result* yang lebih intuitif serta memiliki hasil interpretasi probabilistik yang lebih baik dibanding dengan algoritma klasifikasi yang lainnya.

*Softmax function* merupakan perhitungan kemungkinan dari setiap kelas target atau semua kelas target yang mungkin dan membantu untuk menentukan kelas target atau *input* yang diberikan. *Softmax* memiliki keuntungan yaitu seperti nilai rentang probabilitas yang dihasilkan dari 0 hingga 1 dan jumlah dari semua kemungkinan adalah sama dengan satu. Pada saat *softmax* digunakan untuk model klasifikasi multi, maka akan mengembalikan peluang dari setiap kelas sehingga kelas target akan memiliki probabilitas yang lebih tinggi dari kelas yang lainnya. Perhitungan *softmax* dengan menggunakan eksponensial dari nilai masukkan yang diberikan dan jumlah dari nilai eksponensial dari seluruh nilai dalam masukkan (Khaeriyah, 2019).

#### **3.4.9 Optimizer**

Algoritma optimisasi memiliki tujuan untuk meminimalkan kesalahan, menemukan bobot optimal dan memaksimalkan tingkat akurasi. Dalam proses pelatihan, parameter (bobot) model diganti untuk mencoba serta meminimalkan fungsi kerugian agar dapat memprediksi dengan sangat akurat. Namun permasalahannya adalah kapan, berapa banyaknya perubahan dan bagaimana cara yang tepat itu masih belum dapat dipastikan, maka pada saat inilah pengoptimal masuk. Mereka menyatukan parameter model dan fungsi kerugian dengan memperbaharui model dalam merespon *output* dari fungsi kerugian. Dapat dikatakan dalam istilah sederhana nya adalah, pengoptimalisasi membentuk model yang kita miliki menjadi bentuk yang paling akurat dengan memanfaatkan bobotnya (Sumardi, 2019).

### 3.4.10 *Stochastic Gradient Descent (SGD)*

*Gradient Descent* merupakan salah satu algoritma yang sangat populer dalam melakukan optimasi pada model jaringan syaraf tiruan dan juga algoritma ini adalah cara yang sering digunakan pada berbagai macam model pembelajaran. Pada dasarnya saat melatih sebuah model dibutuhkan sebuah *Loss Function* yang dapat memungkinkan kita untuk mengukur kualitas setiap bobot atau parameter tertentu. Tujuan dari melakukan pengoptimalan adalah untuk menemukan parameter mana yang mampu meminimalkan *Loss Function*.

Cara kerja dari *Gradient Descent* adalah dengan cara meminimalkan fungsi dari  $J(\theta)$  yang memiliki parameter  $\theta$  dengan memperbaharui parameter ke suatu arah menurun. Untuk menentukan langkah-langkah yang akan kita ambil dalam mencapai titik minimum *Gradient Descent* memiliki *Learning Rate* ( $\eta$ ). Hal dapat digambarkan dimana sebuah objek akan menuruni sebuah bukit dengan langkah tersebut sehingga mencapai lembah atau titik minimumnya.

*Stochastic Gradient Descent* atau biasa disingkat sebagai SGD adalah metode dari *Gradient Descent* yang melakukan pengupdate-an parameter untuk setiap data pelatihan  $x(i)$  serta label  $y(i)$  dan memiliki persamaan sebagai berikut:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (3.5)$$

Fungsi objektif meningkat secara tidak beraturan karena SGD sering melakukan update dengan varians yang tinggi. Hal tersebut dapat membuat *Loss Function* melompat ke titik minimal yang baru dan menimbulkan potensi melompat ke minimum yang tidak pasti. Tetapi hal itu dapat dicegah dengan cara mengurangi *Learning Rate* dan SGD akan menuruni *Loss Function* ke titik minimum dengan optimal (Shafira, 2018).

### 3.4.11 **RMSprop**

RMSProp merupakan pengoptimal yang menggunakan besarnya gradien terbaru untuk menormalkan gradien, memantau rata-rata bergerak di atas gradien *root mean square* sehingga disebut sebagai RMS. Yang mana

$f'(\theta_t)$  menjadi turunan dari *loss* berkaitan dengan parameter pada langkah waktu  $t$ . Pada bentuk dasarnya, dengan laju langkah  $\alpha$  dan peluruhan  $\gamma$ , dilakukan pembaruan sebagai berikut:

$$r_t = (1 - \gamma) f'(\theta_t)^2 + \gamma r_{t-1} \quad (3.6)$$

$$v_{t+1} = \frac{\alpha}{\sqrt{r_t}} f'(\theta_t) \quad (3.7)$$

$$\theta_{t+1} = \theta_t - v_{t+1} \quad (3.8)$$

Keterangan:

- $\alpha$  : tingkat pembelajaran awal (*learning rate*)
- $r_t$  : rata-rata eksponensial dari kotak gradien
- $\theta_t$  : gradien pada waktu  $t$  sepanjang  $v^j$

RMSProp mempunyai keunggulan yaitu merupakan pengoptimal yang sangat kuat yang mempunyai informasi kelengkungan semu. Selain itu juga dapat mengatasi tujuan stokastik dengan sangat baik sehingga berlaku juga untuk pembelajaran pada *batch* mini (RMSProp, 2013).

### 3.4.12 Adam

Adam (*Adaptive moment estimation*) merupakan algoritma optimisasi yang sering digunakan dan populer dalam *Deep Learning*. Algoritma optimisasi Adam merupakan pengganti untuk keturunan gradien stokastik dalam melatih model *Deep Learning*. Algoritma Adam diperkenalkan pertama kali oleh Kingma dan Ba. Optimisasi Adam menggabungkan sifat terbaik dari algoritma RMSProp dan AdaGrad untuk menyediakan algoritma pengoptimalan yang mampu menangani gradien yang menyebar. Algoritma Adam bekerja atas:

1. Komponen gradien dengan menggunakan  $m$ , rata-rata bergerak eksponensial gradien (seperti pada momentum), dan
2. Komponen *learning rate* dengan membagi kecepatan pembelajaran  $\alpha$  dengan akar kuadrat  $v$ , rata-rata bergerak eksponensial dari gradien kuadrat (seperti pada RMSProp).

Adapun tahap yang dilakukan dalam *optimizer* Adam yaitu:

Memperbaiki bias *first moment* dan *second moment*

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3.9)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (3.10)$$

Menghitung bias *first moment* dan *second moment*

$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) \cdot g \quad (3.11)$$

$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) \cdot g^2 \quad (3.12)$$

Memperbaiki parameter

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (3.13)$$

Keterangan:

g : gradien

m : *first moment*

v : *second moment*

$\beta_1, \beta_2$  : *Exponential decay rates*

$\alpha$  : *Step size* atau *learning rate*

$\theta$  : parameter yang akan diperbaiki (pada kasus ini adalah bobot)

Nilai yang diinisialisasikan yaitu:

m = 0

v = 0

$\epsilon = 10^{-8}$

t = 0

$\alpha = 0.001$

$\beta_1 = 0.9$

$\beta_2 = 0.999$

Tahap pada *optimizer* Adam diulang sebanyak jumlah *dataset* yang diambil secara *random* sampai seluruh *epoch* telah selesai. Perbedaan antara RMSProp dengan Adam terdapat pada perubahan *step size* di awal perubahan

parameter, karena pada Adam terdapat *bias correction* dalam tahap perhitungannya (Karim, 2018).

### 3.5 *EImage*

Salah satu *package* yang terdapat didalam *software* R yang menyediakan fungsi tujuan umum untuk *pre-processing* dan analisis gambar adalah *package* *EImage*. *EImage* merupakan sebuah alat yang mampu mensegmentasi serta mengekstraksi sel atau gambar mikroskopis secara kuantitatif, bahasa pemrograman R digunakan untuk dapat mengefisiensikan tugas-tugas tersebut serta memfasilitasi penggunaan alat-alat lain dalam lingkungan R sebagai signal *pre-processing*, *Machine Learning*, pemodelan statistik serta visualisasi dengan data gambar (Sumardi, 2019).

### 3.6 Keras

Keras adalah salah satu *package* yang digunakan dalam penyelesaian masalah tentang jaringan syaraf. Keras mendukung hampir semua dari model jaringan saraf dengan fokus mempercepat eksperimen dalam proses konvolusi dan *reccurent* pada *neural networks* maupun antara kombinasi keduanya. Untuk mengekspresikan perhitungan matematisnya dalam pembuatan model jaringan syaraf dengan menggunakan *keras* tidak perlu untuk menuliskan kode nya satu persatu. Hal tersebut dikarenakan *keras* telah menyediakan beberapa model dasar untuk CNN yang telah di optimasi untuk mempermudah penelitian mengenai *deep learning*. Proses komputasi dengan menggunakan *package* *keras* dapat berjalan dengan baik menggunakan CPU ataupun GPU (Shafira, 2018).

### 3.7 Citra

Citra yang secara luas disebut atau dikenal sebagai suatu gambar, merupakan representasi (gambaran), memiliki kemiripan atau imitasi dari suatu objek atau benda. Citra merupakan suatu keluaran sistem perekaman data yang bersifat optik, analog maupun digital. Citra juga dapat di definisikan sebagai suatu fungsi intensitas cahaya dua dimensi  $f(x,y)$ , dengan  $x$  dan  $y$  adalah merupakan koordinat spasial, dengan memiliki ukuran  $M$  baris dan  $N$  kolom. Amplitudo  $f$  pada titik

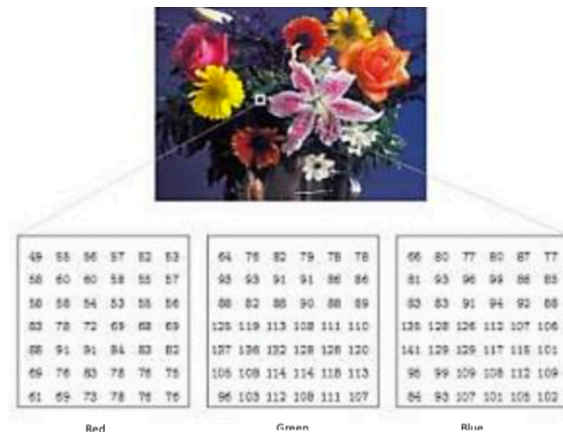


koordinat  $(x,y)$  disebut tingkat keabuan atau intensitas (kecerahan) dari citra pada titik tersebut.

Citra dibagi menjadi dua yaitu citra analog dan citra digital. Citra analog merupakan citra yang bersifat kontinu seperti foto sinar X, plat nomor kendaraan, gambar pada monitor televisi, foto yang dicetak pada kertas foto, sebuah lukisan, hasil dari CT *scan* dan lain sebagainya. Sedangkan citra digital merupakan citra yang dapat diolah oleh komputer. Secara umum citra digital memiliki bentuk persegi panjang atau bujur sangkar bahkan ada yang berbentuk segi enam pada beberapa sistem pencitraan dan memiliki lebar serta tinggi tertentu. Ukuran tersebut biasanya dinyatakan oleh jumlah titik atau piksel sehingga ukuran citra selalu bernilai bulat. Masing-masing titik memiliki koordinatnya sesuai pada posisinya dalam sebuah citra. Biasanya koordinat ini dinyatakan dalam bentuk bilangan bulat positif, yang dimulai dari angka 0 atau 1 bergantung pada sistem yang sedang digunakan. Setiap titiknya memiliki nilai angka digital yang merepresentasikan informasi yang dimiliki oleh titik tersebut (Andono, Sutojo, & Muljono, 2017).

### **3.8 Citra Warna *Red, Green, Blue* (RGD)**

Citra warna merupakan citra yang dimana setiap piksel memiliki tiga komponen warna yang spesifik, yaitu komponen warna merah (*red*), warna hijau (*green*), dan warna biru (*blue*). Warna dari setiap piksel ditentukan oleh kombinasi intensitas dari warna merah, hijau dan biru yang disimpan dalam bidang warna pada lokasi piksel. Citra warna disimpan oleh format file grafis sebagai citra 24 bit yang mana berasal dari masing-masing warna yaitu merah, hijau dan biru sebesar 8 bit. Hal ini dapat membuat citra warna memiliki 24 juta kemungkinan warna. Dapat dilihat pada **Gambar 3.11** dibawah ini yang menunjukkan citra warna jika dilihat dari dekat dengan beberapa nilai intensitas piksel (Andono, Sutojo, & Muljono, 2017).



**Gambar 3.11.** Citra Warna Dengan Komponen RGB  
 Sumber : (Kusumanto & Tompunu, 2011)

### 3.9 Pengolahan Citra (*Image Processing*)

Pengolahan citra atau *image processing* adalah sebuah proses mengolah piksel dalam citra digital untuk tujuan tertentu. Karena beberapa alasan tertentu pengolahan citra dibutuhkan agar mendapatkan citra asli dari sebuah citra yang mengalami penurunan kualitas dikarenakan pengaruh derau atau untuk mendapatkan citra dengan karakteristik dan cocok secara visual yang dibutuhkan untuk tahap lebih lanjut dalam proses analisis citra. Pengolahan citra juga dilakukan untuk memperbaiki kualitas suatu citra agar dapat diinterpretasi oleh manusia atau mesin (komputer). Input dan output nya adalah sebuah citra tetapi dengan kualitas yang lebih baik daripada citra inputan nya, contohnya sebuah citra dengan warna yang kurang tajam, kabur atau blur serta mengandung *noise* (missal terdapat bitnik-bintik putih), sehingga diperlukan proses untuk memperbaiki citra tersebut karena akan sulit di interpretasikan.

Terdapat dua kelompok dalam pengolahan citra yaitu citra diperbaiki sesuai dengan kebutuhan dan mengolah informasi yang terdapat didalam citra. Umumnya mengolah informasi didalam citra adalah untuk mengolah citra dengan cara mengekstraksi informasi penting di dalamnya. Dengan melakukan pengolahan citra diharapkan citra yang diproses dapat di ambil cirinya (Ahmad, 2005).

### 3.10 Piksel (*Pixel*)

Piksel merupakan sebuah kata serapan dari akronim berbahasa Inggris yaitu *Picture Element* yang disingkat menjadi sebuah kata yaitu *Pixel*. Piksel merupakan unsur gambar atau representasi dari titik terkecil dalam sebuah gambar grafis yang dihitung per *inci*. Sebuah objek digital terbentuk dari titik-titik cahaya yang kemudian disebut sebagai piksel.

Semakin banyak jumlah piksel yang terdapat pada sebuah citra digital maka semakin besar pula resolusi spasial dari citra tersebut sehingga citra akan terlihat semakin tajam. Secara umum sebuah gambar terbentuk dari kotak-kotak persegi yang mewakili jarak vertikal dan horizontal yang sama dalam seluruh bagian gambar. Setiap bagian dari piksel diwakili oleh dua bilangan bulat (*integer*) yang menunjukkan lokasinya didalam bidang gambar dan sebuah nilai bilangan bulat guna menunjukkan intensitas cahaya gambar pada piksel tersebut. Lokasi suatu piksel ditunjukkan dengan koordinat  $(0,0)$  pada posisi kiri atas pada bidang gambar dan koordinat  $(m-1,n-1)$  untuk tingkat pencahayaan sebuah piksel, yang dimana biasanya digunakan bilangan bulat sebesar 8-bit dan panjang selisih dengan nilai 0-225 dimana semakin gelap warna dari sebuah gambar maka dilambangkan dengan 0 dan jika semakin terang sebuah gambar maka dilambangkan dengan 255. Dapat diartikan secara keseluruhan bahwa piksel adalah sampel dari pemandangan yang mengandung intensitas sebuah gambar yang dinyatakan dalam bilangan bulat (Ahmad, 2005).

### 3.11 Pengenalan Pola (*Pattern Recognition*)

Pengenalan pola atau *Pattern Recognition* adalah cabang ilmu komputer yang di artikan sebagai pengumpulan data-data mentah untuk diklasifikasikan dengan maksud dan tujuan tertentu. Pengenalan pola ini bersifat *conceptually driven processing*, dimana proses ini dimulai dari pembentukan konsep pada objek yang dijumpai.

Pola merupakan entitas yang dapat diidentifikasi dan terdefinisi melalui ciri-cirinya (*features*), ciri-ciri tersebutlah yang digunakan untuk membedakan suatu pola dengan pola lainnya (Kusumadewi, 2003).

### 3.12 Akurasi

Akurasi dapat didefinisikan sebagai sebuah metrik atau persentase dari data uji yang di klasifikasikan terhadap kelas atau kategori yang benar untuk mengevaluasi hasil dari model klasifikasi. Nilai akurasi didapatkan dari pembagian prediksi model yang di anggap benar terhadap jumlah total yang di prediksi, perhitungan tersebut dapat dinyatakan dalam persamaan berikut:

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Prediksi} \quad (3.14)$$

### 3.13 R Studio

R Studio adalah sebuah aplikasi *Open Sources* dengan bahasa pemrograman dan sistem perangkat lunak dengan beberapa fasilitas untuk manipulasi, perhitungan dan penampilan grafik yang handal yang dirancang khusus untuk keperluan komputasi statistika. Bahasa pemrograman R ini pertama kali dikembangkan oleh dua pakar statistik yaitu Ross Ihaka dan Robert Gentleman pada tahun 1993 di Auckland University, New Zealand. Hingga saat ini Bahasa pemrograman R terus berkembang seiring dengan semakin populernya *Big Data* dan meningkatnya kebutuhan *data scientist* bagi perusahaan untuk mengolah serta menganalisis data di perusahaan tersebut yang digunakan sebagai dasar pengambilan kebijakan (Verzani, 2011).

## BAB 4 METODOLOGI PENELITIAN

### 4.1 Populasi Penelitian

Populasi dalam penelitian ini adalah daun padi yang terserang penyakit dan sampel dalam penelitian ini adalah 120 citra daun padi yang terserang penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*. Sebanyak 120 data citra daun padi tersebut terbagi kedalam tiga kategori berdasarkan jenis penyakitnya yaitu masing-masing sebanyak 40 citra perkategori.


### 4.2 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah merupakan data sekunder dalam bentuk citra yang diperoleh melalui *website UCI Machine Learning Repository archive.ics.uci.edu*, yang merupakan sebuah *website* penyedia data-data untuk proyek *Machine Learning* dan *Intelligent System*. *Keyword* yang digunakan untuk mendapatkan data citra dari penyakit daun padi adalah *Rice Leaf Diseases Data Set*. Dari dataset tersebut terdapat tiga kategori jenis penyakit daun padi yaitu *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot* dengan masing-masing sebanyak 40 citra dari setiap kategori dengan total citra adalah sebanyak 120 dengan *format* JPG. Citra daun padi tersebut di ambil dengan latar belakang berwarna putih dan dibawah sinar matahari langsung. Data ini berasal dari Dharmsinh Desai *University* dari *Department Of Information Technology*, Gujarat, India.

### 4.3 Variabel Penelitian

Adapun variabel-variabel yang digunakan dalam penelitian ini adalah sebagai berikut:

**Tabel 4.1.** Definisi Operasional Variabel

Citra Penyakit	Nama Variabel	Definisi Operasional Variabel
	<i>Bacterial Leaf Blight</i>	<i>Bacterial Leaf Blight</i> atau biasa dikenal dengan penyakit Kresek atau Hawar Daun Bakteri adalah jenis penyakit pada daun padi yang disebabkan oleh bakteri <i>Xanthomonas</i>

Citra Penyakit	Nama Variabel	Definisi Operasional Variabel
		<p><i>campestris</i> pv. <i>Oryzae</i>. Ciri-ciri dari daun yang terserang oleh penyakit ini adalah daun menguning, menggulung, mengering dan menjadi layu, terdapat warna luka bercak yang menjadi jingga kekuningan dari ujung daun ke pangkal daun, ada bulatan kecil berwarna kuning pada pelepah daun di pagi hari dan bibit menjadi layu tetapi sulit dicabut.</p>
	<p><i>Brown Spot</i></p>	<p><i>Brown Spot</i> atau dalam Bahasa Indonesia nya dikenal sebagai penyakit Bercak Daun Coklat adalah penyakit pada daun padi yang disebabkan oleh jamur <i>Helminthosporium oryzae</i>. Penyakit ini ditandai dengan adanya bercak-bercak coklat kecil berbentuk bulat atau oval sejajar dengan tulang daun yang kemudian berkembang menjadi berwarna abu-abu ditengah dikelilingi batas coklat kemerahan pada daun maupun pelepah daun.</p>
	<p><i>Leaf Spot</i></p>	<p><i>Leaf Spot</i> atau <i>Narrow Brown Leaf Spot</i> atau dalam Bahasa Indonesia nya disebut dengan penyakit Bercak Coklat Sempit yang disebabkan oleh jamur <i>Cercospora oryzae</i> Miyake. Gejala yang terjadi apabila terserang penyakit ini adalah terdapat bercak-bercak sempit memanjang pada daun dengan warna coklat kemerahan dan sejajar dengan ibu tulang daun dan</p>

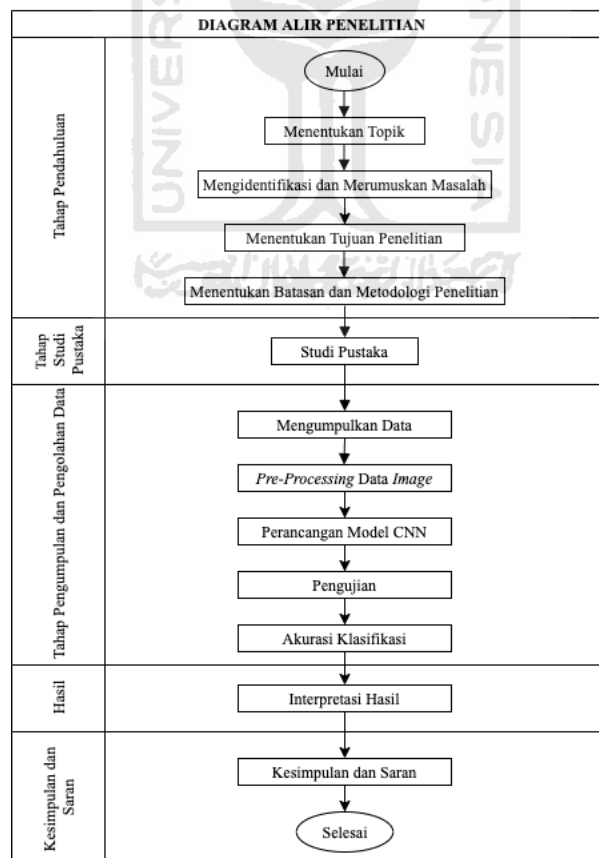
Citra Penyakit	Nama Variabel	Definisi Operasional Variabel
		berukuran panjang kurang lebih 5 mm dengan lebar 1-1,5 mm.

#### 4.4 Metode Analisis Data

Penelitian ini menggunakan Teknik *Deep Learning* dengan metode *Convolutional Neural Network* (CNN) untuk melakukan pengklasifikasian terhadap data gambar. Untuk membantu proses algoritma dari metode tersebut dibantu dengan *software* R Studio Versi 3.6.2 dengan bantuan *package* *EImage*, *Keras* dan *Tensorflow*.

#### 4.5 Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini disajikan dalam diagram alir pada **Gambar 4.1** berikut:



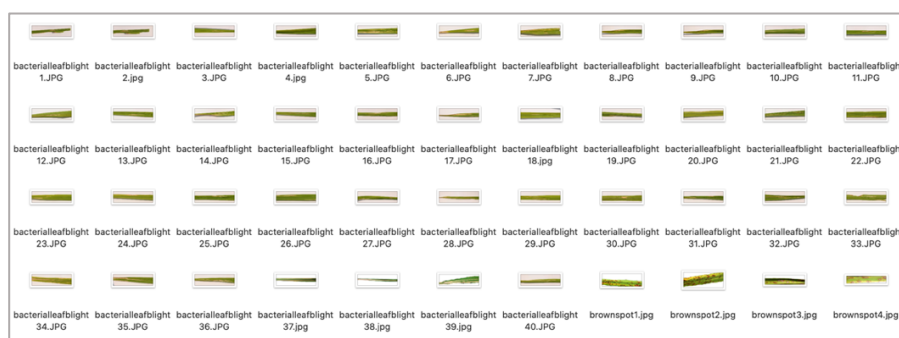
**Gambar 4.1.** Diagram Tahapan Penelitian

## BAB 5 HASIL DAN PEMBAHASAN

Pada tahapan ini akan membahas rangkaian proses dari tahapan-tahapan analisis serta hasil yang didapatkan dari pengklasifikasian tiga kategori penyakit padi berdasarkan citra daun, yaitu citra penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot*. Analisis pada data dilakukan dengan menggunakan metode *Deep Learning* yaitu *Convolutional Neural Network* (CNN). Adapun tahapan yang akan dilakukan yaitu meliputi pengambilan data, *pre-processing*, perancangan model CNN, pengujian model dan hasil klasifikasi.

### 5.1 Pengambilan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang didapatkan melalui sebuah *website* UCI *Machine Learning Repository* dengan alamat *website* yaitu [archive.ics.uci.edu](http://archive.ics.uci.edu), yang merupakan sebuah *website* yang menyediakan data-data untuk proyek *Machine Learning* dan *Intelligent System*, *keyword* yang digunakan untuk mendapatkan data adalah *Rice Leaf Disease Data Set*. Citra dari penyakit berdasarkan daun yang diambil yaitu adalah citra berwarna dan memiliki *size* yang berbeda-beda. Data yang di ambil adalah citra daun padi yang terserang penyakit yaitu sebanyak 120 citra yang terdiri dari tiga kategori. Masing-masing kategori berjumlah 40 citra jenis penyakit *Bacterial Leaf Blight*, 40 citra jenis penyakit *Brown Spot* dan 40 citra jenis penyakit *Leaf Spot*. Seluruh data yang telah di unduh dijadikan satu kedalam sebuah folder dan diberi label sesuai dengan kategori nya serta diberi nomor urut.



Gambar 5.1. Penyimpanan Citra Dalam Satu Folder



## 5.2 Pre-processing Data Citra

Setelah data citra yang akan digunakan dalam penelitian telah didapatkan maka langkah selanjutnya adalah melakukan *preprocessing*. Hal ini dilakukan untuk menyeragamkan unsur citra yang akan dilakukan analisis karena unsur dari masing-masing citra yang sangat bervariasi dan untuk memperjelas fitur dari citra. *Package* yang digunakan untuk melakukan *preprocessing* ini adalah *EBImage* dan *Keras*. Untuk memproses dan menganalisis data dari citra digunakan *package* *EBImage* sedangkan untuk melakukan pengklasifikasian digunakan *package* *Keras* yang merupakan *library Deep Learning*. *Package* tersebut sebelumnya telah terinstal didalam R Studio. Perintah yang digunakan untuk mengaktifkan *package* tersebut adalah `library (EBImage)` dan `library(Keras)`. Dalam penelitian ini semua *size* citra di seragamkan menjadi 100x100 piksel dengan kualitas diubah menjadi 80% karena sebelum dilakukan *preprocessing* ini masing-masing citra memiliki ukuran yang berbeda-beda dan dengan *size* yang cukup besar. Hal ini dilakukan untuk memudahkan pada saat dilakukan proses *training* agar tidak menghabiskan banyak waktu.



**Gambar 5.2.** Citra Sebelum dan Sesudah *Resize*

## 5.3 Pelabelan Kategori

Setelah data disatukan, selanjutnya adalah melakukan pemberian label pada data *training* dan data *testing*. Pemberian label pada citra sesuai dengan kategorinya dilakukan agar *machine* dapat melakukan prediksi. Adapun perintah yang digunakan untuk memberikan pelabelan adalah sebagai berikut:

```
trainy <- c(rep(0,28),rep(1,28),rep(2,28))
testy <- c(rep(0,12),rep(1,12),rep(2,12))
```

Pelabelan dilakukan sesuai dengan jumlah data dengan menggunakan empat perulangan dari fungsi `rep`, dimana untuk data *train* sebanyak 32 data dan untuk

data *test* sebanyak 8 data yang mana angka 0 mewakili citra dari jenis penyakit *Bacterial Leaf Blight*, angka 1 mewakili citra dari jenis penyakit *Brown Spot* dan angka 2 mewakili citra dari jenis penyakit *Leaf Spot*. Berikut adalah *output* dari hasil pola data yang telah diberikan pelabelan:

```

> #TARGET/ PELABELAN DATA
> #80%
> trainy<-c(rep(0,32),rep(1,32),rep(2,32))
> testy<-c(rep(0,8),rep(1,8),rep(2,8))
> trainy
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
 [2] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [64] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> testy
 [1] 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2

```

**Gambar 5.3.** Output Pelabelan Data Dengan Skenario Data Training 80%

#### 5.4 Skenario Data Training Dan Testing

Kemudian *dataset* yang akan dilakukan proses analisis akan dibagi menjadi data *training* dan data *testing*. Untuk penelitian ini pembagian skenario data dilakukan menjadi tiga skenario dengan pembagian sebagai berikut:

**Tabel 5.1.** Skenario Pembagian Data Training Dan Testing

Skenario Dataset	Data Training	Data Testing
<b>70% : 30%</b>	28	12
<b>80% : 20%</b>	32	8
<b>90% : 10%</b>	36	4

Dapat dilihat pada **Tabel 5.1** di atas adalah pembagian dari masing-masing data untuk ketiga skenario di atas. Adapun perintah yang dilakukan untuk melakukan *penginputan* ke dalam pemrograman R Studio berdasarkan skenario tersebut adalah dengan menggunakan perintah sebagai berikut:

**1. Skenario Data Training 70% Dan Data Testing 30%**

```

train<-gambar2[c(1:28,41:68,81:108)]
test<-gambar2[c(29:40,69:80,109:120)]

```

**2. Skenario Data Training 80% Dan Data Testing 20%**

```

train<-gambar2[c(1:32,41:72,81:112)]
test<-gambar2[c(33:40,73:80,113:120)]

```

### 3. Skenario Data *Training* 90% Dan Data *Testing* 10%

```
train<-gambar2[c(1:36,41:76,81:116)]  
test<-gambar2[c(37:40,77:80,117:120)]
```

Berikut interpretasi berdasarkan dari pembagian skema *dataset* di atas, misal dengan menggunakan skenario data *training* 80% dan *testing* 20% maka pembagian *dataset* nya yaitu untuk data *training* sebanyak 32 citra penyakit padi berdasarkan daun dan untuk data *testing* sebanyak 8 citra penyakit padi berdasarkan daun. Dimana untuk data *training* pada data dari urutan 1 hingga 32 adalah citra dengan jenis penyakit padi *Bacterial Leaf Blight*, data pada urutan ke 41 hingga 72 adalah data citra dengan jenis penyakit *Brown Spot* dan pada urutan data ke 81 hingga 112 merupakan citra dengan jenis penyakit *Leaf Spot*. Sedangkan untuk *dataset* testing yaitu pada urutan data ke 33 hingga 40 merupakan citra dengan jenis penyakit *Bacterial Leaf Blight*, pada urutan data ke 73 hingga 80 merupakan data citra dengan jenis penyakit *Brown Spot* dan pada data urutan ke 117 hingga 120 merupakan data citra penyakit *Leaf Spot*.

#### 5.5 Pembuatan Model CNN

Tahap pembuatan model CNN ini dilakukan untuk mendapatkan model yang terbaik agar mampu menghasilkan klasifikasi yang bagus dari model yang telah dirancang. Pada tahapan ini dilakukan beberapa percobaan untuk melakukan perbandingan dari kemampuan model menggunakan arsitektur CNN yang tepat untuk mendapatkan hasil yang terbaik. Namun sebelum melakukan perancangan model arsitektur, agar data dapat diproses oleh *Library Keras* maka data perlu diubah terlebih dahulu kedalam skema *one hot encoding*, dimana hal tersebut berfungsi untuk merubah fitur kategori pada data menjadi sebuah format yang berfungsi lebih baik dengan pengklasifikasian. Setelah itu, dilakukan perbandingan pada beberapa parameter pada arsitektur CNN agar didapatkan model yang terbaik. Adapun parameter-parameter yang dilakukan perbandingan adalah nilai *epoch*, jenis *optimizer* dan skenario *dataset*. Rancangan model CNN ini dibandingkan dengan arsitektur awal menggunakan skenario *dataset* dengan perbandingan data *train* 80% dan data *test* 20% dengan *optimizer* Adam dan nilai *epoch* 50.

### 5.5.1 Membandingkan Nilai *Epoch*

Parameter pertama yang akan dilakukan perbandingan untuk memperoleh arsitektur terbaik adalah parameter *epoch*. *Epoch* adalah bagian pada proses pembelajaran jaringan syaraf, besarnya nilai *epoch* yang ditetapkan akan mempengaruhi proses pembelajaran dan akan berhenti tepat pada nilai *epoch* yang ditentukan, proses kerja dari *epoch* adalah serupa dengan iterasi tetapi bedanya adalah *epoch* merupakan iterasi dengan rambatan balik. Pada penelitian ini nilai *epoch* yang akan dilakukan perbandingan adalah sebesar 50, 100 dan 150. Berdasarkan pada hasil *training* yang telah dilakukan maka diperoleh hasil perbandingan nilai *accuracy* dan *loss accuracy* sebagai berikut:

**Tabel 5.2.** Hasil Perbandingan Nilai *Epoch*

<i>Epoch</i>	<i>Train</i>		<i>Test</i>	
	<i>Accuracy</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Loss</i>
50	0,813	2,128	0,75	2,546
100	0,823	2,547	0,75	1,656
150	0,813	3,236	0,79	2,366

Berdasarkan pada **Tabel 5.2** di atas dapat dilihat bahwa semakin besar nilai *epoch* yang digunakan maka hasil akurasi nya juga akan semakin baik. Hal ini dikarenakan semakin besar nilai *epoch* membuat *machine* lebih terlatih dalam mengenali citra dan mampu mengenal *pattern* lebih baik lagi. Semakin besar nilai *epoch* yang digunakan maka akan semakin lama waktu *training* yang diperlukan. Berdasarkan pada hasil dari tabel di atas maka peneliti akan menggunakan nilai *epoch* sebesar 150 karena hasil akurasi yang didapatkan adalah yang terbaik dibanding dengan yang lainnya.

### 5.5.2 Membandingkan Jenis *Optimizer*

Tahap selanjutnya adalah perbandingan terhadap parameter jenis *optimizer*, terdapat tiga jenis *optimizer* yaitu Adam, *RMSProp* dan *Sgd*. Berdasarkan dari pelatihan algoritma yang telah dilakukan terhadap ketiga jenis *optimizer* tersebut maka didapatkan hasil sebagai berikut:

**Tabel 5.3.** Hasil Perbandingan Jenis *Optimizer*

<i>Optimizer</i>	<i>Train</i>		<i>Test</i>	
	<i>Accuracy</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Loss</i>
Adam	0,813	4,003	0,833	2,314
RMSProp	0,802	3,162	0,792	2,815
Sgd	0,802	1,213	0,667	0,869

Dapat dilihat pada **Tabel 5.3** di atas adalah hasil dari analisis yang telah dilakukan dengan menggunakan ketiga jenis *optimizer* dengan menggunakan *epoch* sebesar 150, berdasarkan pada perbandingan dari ketiga hasil tersebut diketahui bahwa jenis *optimizer* Adam menghasilkan tingkat akurasi yang paling tinggi dari yang lainnya, *optimizer* Adam menghasilkan tingkat akurasi sebesar 0,813 dengan kesalahan sebesar 4,003 pada data *Train* dan pada data *Test* didapatkan tingkat akurasi sebesar 0,833 dengan tingkat kesalahan sebesar 2,314. Sedangkan pada jenis *optimizer* RMSProp didapatkan tingkat akurasi sebesar 0,802 dengan kesalahan sebesar 3,162 pada data *Train* dan pada data *Test* didapatkan tingkat akurasi sebesar 0,792 dengan kesalahan sebesar 2,815 dan pada jenis *optimizer* Sgd didapatkan tingkat akurasi sebesar 0,802 dengan kesalahan sebesar 1,213 untuk data *Train* dan untuk data *Test* didapatkan tingkat akurasi sebesar 0,667 dengan kesalahan sebesar 0,869. Sehingga secara keseluruhan dapat disimpulkan bahwa *optimizer* Adam menghasilkan tingkat akurasi yang lebih baik daripada yang lainnya karena tingkat akurasi yang didapatkan lebih besar dibanding yang lain.

### **5.5.3 Membandingkan Skenario *Dataset***

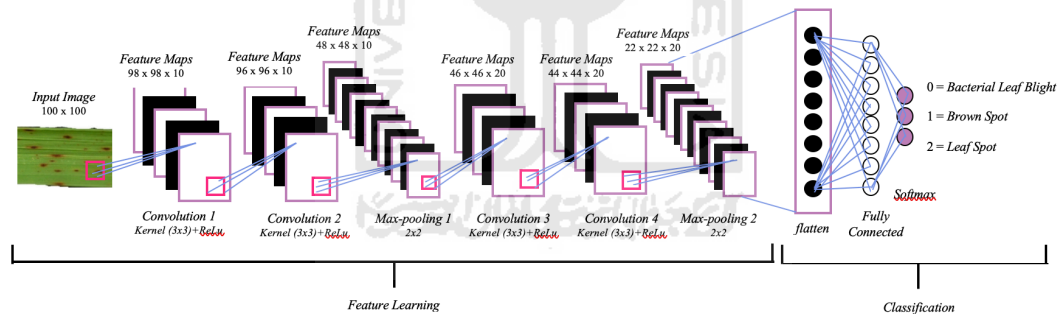
Tahap ini merupakan tahap terakhir dalam rangkaian model arsitektur CNN yang telah dirancang, pada tahap ini akan dilakukan perbandingan skenario *dataset* 70% : 30%, 80% : 20% dan 90% : 10% dengan menggunakan nilai *epoch* sebesar 150 dan dengan jenis *optimizernya* adalah Adam. Maka hasil yang didapatkan adalah sebagai berikut:

**Tabel 5.4.** Hasil Perbandingan Skenario *Dataset*

Skenario <i>Dataset</i>	<i>Train</i>		<i>Test</i>	
	<i>Accuracy</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Loss</i>
70% : 30%	0,857	3,761	0,611	6,564
80% : 20%	0,813	3,345	0,833	1,842
90% : 10%	0,815	3,687	0,917	0,088

Berdasarkan hasil perbandingan tingkat akurasi pada **Tabel 5.4** di atas dapat dilihat bahwa tingkat akurasi tertinggi terdapat pada skenario *dataset* 90% : 10% yaitu sebesar 0,917 atau 91,7% pada data *test*. Sehingga dapat disimpulkan bahwa dengan menggunakan skenario *dataset* sebesar 90% : 10% *machine* mampu mengklasifikasikan citra dengan baik.

Setelah dilakukan perbandingan pada parameter nilai *epoch*, jenis *optimizer* dan skenario *dataset*, maka didapatkan arsitektur jaringan terbaik yaitu sebagai berikut:



**Gambar 5.4.** Skema Arsitektur CNN

Dari gambar di atas dapat dilihat bahwa skema arsitektur CNN terbagi menjadi dua tahapan yaitu *feature learning* dan *classification*, berikut akan dijelaskan lebih lanjut tahapan berdasarkan dari gambaran arsitektur tersebut:

- Tahapan *Feature Learning*

1. *Convolutional Layer*

Konvolusi adalah tahapan proses pertama dalam mengolah citra yang berguna dalam mempertajam, memperhalus dan mendeteksi tepi. Lapisan konvolusi memiliki fungsi untuk menghitung produk titik bobot mereka dan wilayah kecil yang terhubung ke volume *input*. Pada tahapan ini pada

saat semua data menyinggung lapisan konvolusional maka akan terjadi proses konvolusi. Dalam penelitian ini digunakan empat kali konvolusi dengan masing-masing dua kali konvolusi pada setiap *pooling*, yaitu *convolution 1 & convolution 2* untuk *pooling* pertama dan *convolution 3 & convolution 4* untuk *pooling* kedua. Konvolusi merupakan proses awal yang dilakukan setelah menerima citra yang di *input* kedalam arsitektur. Lapisan konvolusi merupakan sebuah filter yang memiliki panjang dan lebar serta ketebalan yang sesuai dengan jumlah *channel* pada citra. Konvolusi dibantu oleh kernel dan *filter*, kernel yang peneliti gunakan adalah ukuran (3x3) yang mana akan bergeser ke semua bagian piksel citra yang di *input* yaitu sebesar 100x100, pergeseran yang terjadi tersebut yang disebut sebagai operasi konvolusi. Kemudian sebanyak 10 *filter* yang akan digunakan dalam mencari *pattern* dari bagian sebuah citra sehingga akan terdapat 10 bentuk yang berbeda yang akan dicari pada citra yang di *input* kan. Setelah proses konvolusi selesai dilakukan maka akan didapatkan hasil sebuah *output* yang disebut sebagai *activation maps* atau *features maps* yang selanjutnya dibantu dengan fungsi aktivasi ReLu untuk merubah bilangan atau *output* yang bernilai negatif menjadi 0 serta *output* yang bernilai positif akan memiliki nilai aktivasi itu sendiri.

## 2. *Pooling Layer*

Tahapan ini dilakukan setelah proses *convolutional layer*, pada penelitian ini digunakan *max-pooling* yang berfungsi untuk mereduksi ukuran citra atau mengurangi ukuran dimensi *feature maps* yang dapat mempercepat komputasi serta untuk mengatasi *overfitting*. *Max-pooling* dengan ukuran 2x2 digunakan untuk menghitung *feature maps* yang memiliki ukuran 50x50, nilai maksimumnya akan dihitung dari setiap bagian *feature maps* yang berukuran 2x2 sehingga ukuran *feature maps* berkurang menjadi 2x2 yang mana hal ini akan mengurangi jumlah dari bilangan yang harus dihitung.

- Tahapan *Classification*

1. *Flatten*

*Output* yang didapatkan dari tahapan konvolusi yang dilakukan adalah *feature maps* dengan bentuk *multidimensional array*, sehingga proses yang dilakukan selanjutnya yaitu adalah *flatten* yang mengubah *feature maps* menjadi sebuah vektor agar bisa di *input* kedalam tahap selanjutnya yaitu *fully connected layer* untuk dilakukan proses klasifikasi.

2. *Fully Connected Layer*

Pada tahapan ini terdapat *hidden layer*, *activation function*, *output layer* serta *loss function* yang digunakan untuk menerapkan *multi layer perceptron*. Proses ini menerapkan metode *dropout* yang bertujuan menonaktifkan beberapa *edge* yang berhubungan dengan neuron untuk menghindari terjadinya *overfitting* serta untuk mempercepat proses *learning*. Kemudian lanjut menuju proses klasifikasi, *input* akan diklasifikasi sesuai dengan target kategorinya dengan menggunakan bantuan aktivasi *softmax*. Target kategori tersebut terdiri dari 3 kategori yaitu 0 untuk jenis penyakit *Bacterial Leaf Blight*, 1 untuk jenis penyakit *Brown Spot* dan 2 untuk jenis penyakit *Leaf Spot*.

Setelah rancangan arsitektur jaringan telah selesai, dengan jumlah parameter yang dilatih pada model yang didapatkan adalah sebanyak 2.911.833, maka didapatkan model terbaik dengan rincian sebagai berikut:

```
> summary(model)
Model: "sequential_9"
```

Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 98, 98, 10)	280
conv2d_37 (Conv2D)	(None, 96, 96, 10)	910
max_pooling2d_18 (MaxPooling2D)	(None, 48, 48, 10)	0
dropout_27 (Dropout)	(None, 48, 48, 10)	0
conv2d_38 (Conv2D)	(None, 46, 46, 20)	1820
conv2d_39 (Conv2D)	(None, 44, 44, 20)	3620
max_pooling2d_19 (MaxPooling2D)	(None, 22, 22, 20)	0
dropout_28 (Dropout)	(None, 22, 22, 20)	0
flatten_9 (Flatten)	(None, 9680)	0
dense_18 (Dense)	(None, 300)	2904300
dropout_29 (Dropout)	(None, 300)	0
dense_19 (Dense)	(None, 3)	903

```
Total params: 2,911,833
Trainable params: 2,911,833
Non-trainable params: 0
```

**Gambar 5.5.** *Output Summary Model Terbaik*



## 5.6 Hasil Klasifikasi Model Terbaik

Model arsitektur terbaik telah didapatkan maka selanjutnya dilakukan pengujian pada data *testing*, berikut adalah hasil prediksi dari klasifikasi yang didapatkan:

**Tabel 5.5.** Hasil Klasifikasi Model Terbaik Data *Test*

Prediksi	Aktual		
	0	1	2
0	4	0	0
1	0	4	1
2	0	0	3

Dapat dilihat pada **Tabel 5.5** di atas bahwa pada data *testing* dapat melakukan prediksi dengan tepat untuk kategori penyakit *Bacterial Leaf Blight* dan *Brown Spot* dengan masing-masing data *test* sebanyak 4 citra perkategori, sedangkan untuk kategori *Leaf Spot machine* hanya dapat melakukan prediksi dengan benar sebanyak 3 citra yang berarti sebanyak 1 citra terjadi kesalahan dalam prediksi dan kesalahan tersebut terkategori masuk kedalam jenis penyakit *Brown Spot*.

Berdasarkan dari hasil klasifikasi dalam bentuk *confusion* matriks seperti pada **Tabel 5.5** di atas, dapat diketahui tingkat akurasi dengan menggunakan perhitungan secara manual seperti berikut:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}} = \frac{11}{12} = 0,917 = 91,7\%$$

**Tabel 5.6.** Output Akurasi Model Terbaik

Data	90% : 10%	
	<i>Accuracy</i>	<i>Loss</i>
<i>Testing</i>	0,917	0,088

Nilai akurasi yang didapatkan dari hasil perhitungan secara manual sama dengan hasil nilai tingkat akurasi yang didapatkan dari perhitungan algoritma *machine*, dimana nilai akurasi yang didapatkan adalah sebesar 0,917 atau 91,7% dengan kesalahan sebesar 0,088 atau 8,8% yang berarti tingkat kesalahan lebih kecil dibanding tingkat ketepatan dalam melakukan prediksi, sehingga dapat disimpulkan bahwa dengan nilai akurasi yang didapatkan tersebut telah cukup baik dan mampu mengklasifikasikan citra jenis penyakit pada padi berdasarkan daun tepat sesuai dengan kategorinya.



## BAB 6 PENUTUP

### 6.1 Kesimpulan

Berdasarkan dari tahapan analisis yang telah dilakukan dan didapatkan hasilnya sehingga diperoleh beberapa kesimpulan antara lain:

1. Implementasi algoritma CNN dalam mengklasifikasikan citra daun padi yang terserang penyakit dilakukan dengan mencari rancangan arsitektur terbaik dengan membandingkan beberapa parameter yaitu *epoch*, jenis *optimizer* dan skenario *dataset*. Arsitektur CNN terbaik yang diperoleh berdasarkan dari hasil perbandingan beberapa parameter dalam mengklasifikasikan citra jenis penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot* adalah dengan menggunakan parameter *size* 100x100 piksel, ukuran kernel 3x3, *learning rate* 0,01, jenis *optimizer* Adam, *epoch* 150, *batch size* 30 dan skenario perbandingan dataset 90% : 10% dengan jenis citra RGB (berwarna).
2. Hasil tingkat akurasi yang diperoleh dari data *testing* dengan menggunakan model arsitektur terbaik dalam pengklasifikasian citra jenis penyakit pada padi berdasarkan daunnya yaitu sebesar 91,7%.

### 6.2 Saran

Adapun saran yang diberikan berdasarkan dari penelitian ini sebagai upaya perbaikan dan pengembangan pada penelitian lanjutan adalah sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menambahkan lebih banyak citra jenis penyakit *Bacterial Leaf Blight*, *Brown Spot* dan *Leaf Spot* agar memperoleh tingkat akurasi yang lebih akurat.
2. Penelitian selanjutnya diharapkan dapat menambahkan jenis kategori penyakit padi berdasarkan daunnya yang dapat di analisis dengan menggunakan citra, mengingat masih banyak jenis penyakit padi lainnya.
3. Penelitian selanjutnya diharapkan mampu menggunakan *software Python* dan menggunakan metode analisis lain yang mampu memberikan tingkat akurasi yang lebih baik.

## DAFTAR PUSTAKA

- Afni, R. (2012). *Pengaruh Lama Pengukusan Dan Cara Penanakan Beras Pratanak Terhadap Mutu Nasi Pratanak*. Bogor: Skripsi Fakultas Teknik Pertanian Institut Pertanian Bogor.
- Agus, S. (2019). *Teknologi Produksi Tanaman Budi Daya*. Malang: UB Press.
- Ahmad, U. (2005). *Pengolahan Citra Digital Dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- Andono, P. N., Sutojo, T., & Muljono. (2017). *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
- Bulelengkab.go.id. (2018, Desember 3). *Penyakit Bercak Daun Cercospora Pada Tanaman Padi Dan Cara Pengendaliannya*. Retrieved from Website Resmi Pemerintah Kabupaten Buleleng: <https://bulelengkab.go.id/detail/artikel/penyakit-bercak-daun-cercospora-pada-tanaman-padi-dan-cara-pengendaliannya-24#:~:text=Penyakit%20bercak%20daun%20cercospora%20atau,di%20negara%2Dnegara%20penanam%20padi.&text=Menjadi%20penyakit%20utama%20pada%20per>
- Felix, Faisal, S., Butarbutar, T. F., & Sirait, P. (2019, Oktober). Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun. *Jurnal SIFO Mikroskil, Vol. 20, No. 2*.
- Harjoseputro, Y., Handarkho, Y. D., & Adie, H. T. (2019). The Javanese Letters Classifier with Mobile Client-Server Architecture and Convolution Neural Network Method. *International Journal of Interactive Mobile Technologies (iJIM), Vol. 13, No.12*.
- Hidayat, B., & Hermawan, G. (2018). Deteksi Hama Pada Daun Teh Dengan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- Ihsan, M., Agus, F., & Khairina, D. M. (2017). Penerapan Metode Dempster Shafer Untuk Sistem Deteksi Penyakit Tanaman Padi. *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi, Vol. 2, No. 1*.

- Ishaq, M., Rumiati, A. T., & Permatasari, E. O. (2017). Analisis Faktor-Faktor yang Mempengaruhi Produksi Padi di Provinsi Jawa Timur Menggunakan Regresi Semiparametrik Spline. *Jurnal Sains Dan Seni ITS, Vol. 6, No. 1*. Surabaya.
- Karim, R. (2018, 11 22). *Towards Data Science*. Retrieved 11 2020, from 10 Stochastic Gradient Descent Optimisation Algorithms + Cheat Sheet: <https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9>
- Khaeriyah, R. (2019). *Implementasi Metode Convolutional Neural Network Menggunakan Tensorflow Dalam Mendeteksi Sebuah Objek*. Yogyakarta: Skripsi UII.
- Khan, S., Rahmani, H., Shah, S. A., & Bennamoun, M. (2018). *A Guide to Convolutional Neural Networks for Computer Vision*. New York: Morgan & Claypool Publishers.
- Kusanti, J., & Haris, N. A. (2018). Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT), Vol. 03, No.01*.
- Kusumadewi, S. (2003). *Artificial Intelligence (Teknik Dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Kusumaningrum, T. F. (2018). *Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras*. Yogyakarta: Skripsi UII.
- Kusumanto, R., & Tompunu, A. N. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan (Semantik)*.
- Pham, D. (2018, 10 29). *An Introduction to Machine Learning*. Retrieved 06 2020, from Medium: <https://medium.com/@grepdennis/an-introduction-to-machine-learning-7ac2a6219ff4>

- Pradana, S. C. (2018). *Implementasi Convolutional Neural Network Terhadap Instrumen Alat Musik Gamelan Menggunakan Keras*. Yogyakarta: Skripsi UII.
- Prajapati, H. B. (2019, 04 14). *Rice Leaf Diseases Data Set*. Retrieved 03 2020, from UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases>
- Prasetyo, Y. (2012). *Budi Daya Padi Sawah Tanpa Olah Tanah*. Yogyakarta: Kanisius.
- Pratiwi, S. H. (2016, Juni). Pertumbuhan Dan Hasil Padi (*Oryza sativa* L.) Sawah Pada Berbagai Metode Tanam Dengan Pemberian Pupuk Organik. *Gontor AGROTECH Science Journal, Vol. 2, No.2*.
- Ratnasari, M. C. (2018). *Deep Learning Convolutional Neural Network Untuk Klasifikasi Pengenalan Objek Menggunakan MXNET*. Yogyakarta: Skripsi UII.
- RMSProp. (2013). Retrieved Mei 17, 2020, from [climin.readthedocs.io](https://climin.readthedocs.io): <https://climin.readthedocs.io/en/latest/rmsprop.html>
- Saha, S. (2018, 12 16). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Retrieved 06 2020, from Towards Data Science: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- Sahbana, A. (2017). *Implementasi Sistem Pakar Dalam Mengidentifikasi Penyakit Tanaman Padi Menggunakan Metode Certainty Factor*. Semarang: Skripsi Universitas Negeri Semarang.
- Semangun, H. (2008). *Penyakit-Penyakit Tanaman Pangan Di Indonesia* (Vol. Edisi Kedua). Yogyakarta: Gadjah Mada University Press.
- Shafira, T. (2018). *Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras*. Yogyakarta: Skripsi UII.
- Sudir, Nuryanto, B., & Kadir, T. S. (2012, Oktober). Epidemiologi, Patotipe, dan Strategi Pengendalian Penyakit Hawar Daun Bakteri pada Tanaman Padi. *Jurnal Iptek Tanaman Pangan, Vol. 7, No.2*.

- Sumardi, D. G. (2019). *Implementasi Algoritma CNN Dalam Klasifikasi Gangguan Mata Menggunakan Pendekatan Image Processing*. Yogyakarta: Skripsi UII.
- Suparyono. (2007). Pengendalian Hama Dan Penyakit Tanaman Padi. *Jurnal Balai Penelitian Tanaman Padi*.
- Suryanto, W. A. (2010). *Hama Dan Penyakit Tanaman Pangan, Hortikultura dan Perkebunan, Masalah Dan Solusinya*. Yogyakarta: Kanisius.
- Sutarman. (2017). *Dasar-Dasar Ilmu Penyakit Tanaman*. Sidoarjo: UMSIDA PRESS.
- Utama, M. H. (2019). *Budi Daya Padi Hitam Dan Merah Pada Lahan Marginal Dengan Sistem SBSU*. Yogyakarta: ANDI.
- Utama, M. Z. (2015). *Budidaya Padi Pada Lahan Marjinal (Kiat meningkatkan Produksi Padi)*. Yogyakarta: ANDI.
- Verzani, J. (2011). *Getting Started With RStudio*. United States Of America: O'Reilly Media, Inc.
- Whidhiasih, R. N., & Ekawati, I. (2019). Identifikasi Jenis Penyakit Daun Padi Menggunakan Adaptif Neuro Fuzzy Inferen System (ANFIS) Berdasarkan Tekstur. *Jurnal Sinergi (Seminar Nasional Energi & Teknologi)*.
- Wicaksono, G., Andrayana, S., & Benrahman. (2020). Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network. *Journal of Information Technology and Computer Science (JOINTEC), Vol. 5, No.1*.

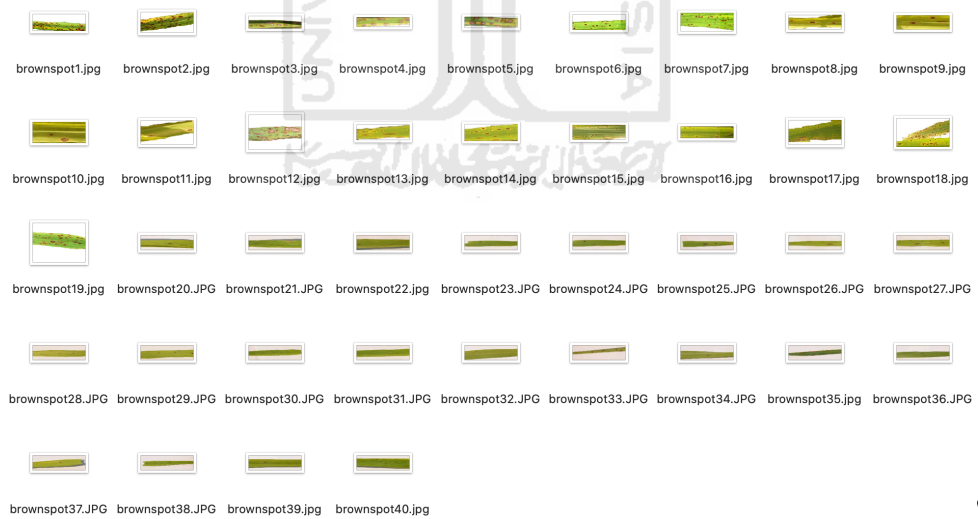
# LAMPIRAN

## Lampiran 1 Data Citra Penyakit Daun Padi

### 1. Penyakit *Bacterial Leaf Blight*

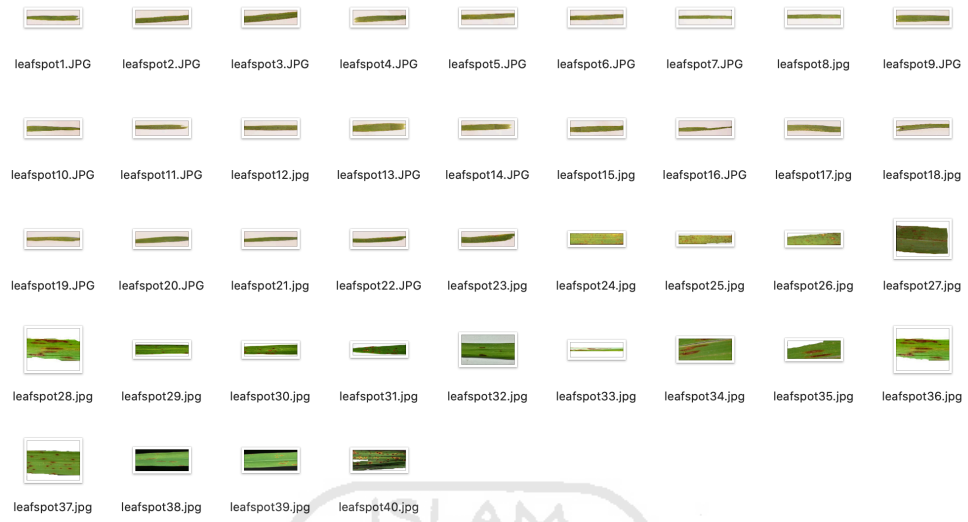


### 2. Penyakit *Brown Spot*





### 3. Penyakit *Leaf Spot*



## Lampiran 2 *Syntax* Program R Studio

```
library(keras)

library(EBImage)

library(tensorflow)

#PRE-PROCESSING (RESIZE)

setwd("~/Documents/RICE LEAF DISEASES/LEAF")

save_in<-("~/Documents/RICE LEAF DISEASES/resize/resize")

gambar<-list.files()

w<-100
h<-100
for (i in 1:length(gambar))
{result<-tryCatch({
  imgname<-gambar[i]
  img<-readImage(imgname)
  img_resized<-resize(img, w=w, h=h)
  path<-paste(save_in,imgname,sep = " ")
  writeImage(img_resized,path,quality = 80)
  print(paste("done",i,sep = ""))
},
error=function(e){print(e)}
)}

#ANALISIS CNN#

setwd("~/Documents/RICE LEAF DISEASES/resize")

gambar2<-list.files()

gambar2

summary(gambar2)
```

```

gambar2<-lapply(gambar2, readImage)

str(gambar2)

dim(gambar2[[4]])

head(gambar2)

#create train
train<-gambar2[c(1:36,41:76,81:116)]

train

#create test
test<-gambar2[c(37:40,77:80,117:120)]

test

# combine
train <- combine(train)
x <- tile(train, 108)
display(x, title='train')
dim(train)

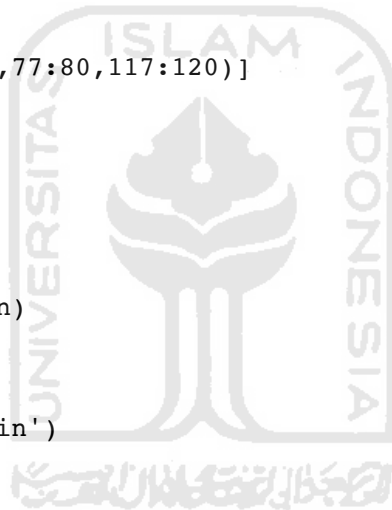
test <- combine(test)
y <- tile(test, 12)
display(y, title = 'test')
dim(test)

#reorder dimension
train<-aperm(train,c(4,1,2,3))
test<-aperm(test,c(4,1,2,3))

dim(train)

dim(test)

```



```

#TARGET/ PELABELAN DATA

#90%

trainy<-c(rep(0,36),rep(1,36),rep(2,36))

testy<-c(rep(0,4),rep(1,4),rep(2,4))

trainy

testy

#One Hot Encoding#
#CONNECT TO PYTHON
trainLabels<-to_categorical(trainy)
testLabels<-to_categorical(testy)

trainLabels

testLabels

#ARSITEKTUR CNN
model<-keras_model_sequential()
model%>%
  layer_conv_2d(filters = 10, kernel_size = c(3,3),
               activation = 'relu',
               input_shape = c(100,100,3))%>%
  layer_conv_2d(filters = 10, kernel_size = c(3,3),
               activation = 'relu')%>%
  layer_max_pooling_2d(pool_size = c(2,2))%>%
  layer_dropout(rate = 0.01)%>%
  layer_conv_2d(filters = 20,
               kernel_size = c(3,3),
               activation = 'relu')%>%

```

```

layer_conv_2d(filters = 20,
              kernel_size = c(3,3),
              activation = 'relu')%>%
layer_max_pooling_2d(pool_size = c(2,2))%>%
layer_dropout(rate = 0.0001)%>%
layer_flatten()%>%
layer_dense(units = 300, activation = 'relu')%>%
layer_dropout(rate=0.01)%>%
layer_dense(units = 3, activation = 'softmax')%>%
compile(loss = 'categorical_crossentropy',
        optimizer = 'Adam',
        metrics = c('accuracy'))
summary(model)

#PROSES RUNNING ARSITEKTUR
proses<-model%>%
  fit(train, trainLabels,
      epochs = 150,
      batch_size = 30,
      validation_split = 0.2
  )
plot(proses)

#LIHAT HASIL AKURASI MODEL
#train
model%>%evaluate(train,trainLabels)
pred<-model %>% predict_classes(train)
pred
table(Predicted=pred, Actual=trainy)

```

```
prob <- model %>% predict_proba(train)
cbind(prob, Predicted_class = pred, Actual = trainy)

#test
model%>%evaluate(test,testLabels)
pred<-model%>%predict_classes(test)
pred
table(Predicted=pred, Actual=testy)
prob <- model %>% predict_proba(test)
cbind(prob, Predicted_class = pred, Actual = testy)
```



### Lampiran 3 Hasil Klasifikasi

```

> #test
> model%>%evaluate(test,testLabels)
12/1 [=====]
=====] - 0s 6ms/sample - loss: 0.0878 - accuracy: 0.9167
$loss
[1] 0.08776108

$accuracy
[1] 0.916667

> pred<-model%>%predict_classes(test)
> pred
[1] 0 0 0 0 1 1 1 1 2 2 2 1 2
> table(Predicted=pred, Actual=testy)
      Actual
Predicted 0 1 2
      0 4 0 0
      1 0 4 1
      2 0 0 3
> prob <- model %>% predict_proba(test)
> cbind(prob, Predicted_class = pred, Actual = testy)
      Predicted_class Actual
[1,] 9.999995e-01 4.723491e-07 3.886031e-09 0 0
[2,] 9.998415e-01 1.585698e-04 7.323467e-13 0 0
[3,] 9.999974e-01 2.531747e-06 1.701277e-07 0 0
[4,] 9.999998e-01 1.568843e-07 8.099065e-08 0 0
[5,] 9.735967e-08 9.999999e-01 6.992174e-19 1 1
[6,] 1.893869e-16 1.000000e+00 3.132703e-28 1 1
[7,] 1.065095e-16 1.000000e+00 1.493820e-23 1 1
[8,] 2.217262e-21 1.000000e+00 8.961079e-29 1 1
[9,] 6.536025e-05 1.735825e-02 9.825764e-01 2 2
[10,] 7.889550e-05 1.616351e-05 9.999050e-01 2 2
[11,] 6.806742e-04 5.245068e-01 4.748125e-01 1 2
[12,] 1.576163e-05 2.520674e-01 7.479168e-01 2 2

```

