

SKRIPSI

**KLASIFIKASI PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *DEEP LEARNING***

Disusun dan diajukan oleh:

TEDI SETIADY P. BA'KA

D42116508



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

LEMBAR PENGESAHAN SKRIPSI

**KLASIFIKASI PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *DEEP LEARNING***

Disusun dan diajukan oleh

TEDI SETIADY P. BA'KA

D421 16 508

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin
Pada tanggal 2 Juni 2021
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama,

Pembimbing Pendamping,

Dr. Indrabayu, S.T., M.T., M.Bus.Sys.
NIP. 19750716 200212 1 004

Anugrahvani Bustamin, S.T., M.T.
NIP. 19901201 201807 4 001



Program Studi,

Dr. Amir Alham, S.T., M.IT.
NIP. 19731010 199802 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : TEDI SETIADY P. BA'KA

NIM : D421 16 508

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul:

KLASIFIKASI PENYAKIT TANAMAN PADI MENGGUNAKAN METODE *DEEP LEARNING*

Adalah karya tulisan saya sendiri, bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 2 Juni 2021

Yang Menyatakan,



TEDI SETIADY P. BA'KA

SKRIPSI

**KLASIFIKASI PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *DEEP LEARNING***

Disusun dan diajukan oleh:

TEDI SETIADY P. BA'KA

D42116508



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga skripsi yang berjudul “**Klasifikasi Penyakit Tanaman Padi menggunakan Metode *Deep Learning***” ini dapat terselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa banyak kendala yang dihadapi dalam menyelesaikan skripsi ini, namun berkat dorongan, dukungan, bimbingan serta motivasi yang diberikan sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Ucapan terima kasih serta penghargaan yang setinggi-tingginya penulis sampaikan kepada:

1. Tuhan Yang Maha Esa yang melalui berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Kedua Orang tua penulis, Bapak Daud Pakiding dan Ibu Oktofiana Barwati, yang selalu memberikan dukungan, doa, semangat dan kasih sayang serta selalu sabar dalam mendidik penulis sejak kecil.
3. Bapak Dr. Indrabayu S.T., M.T., M.Bus.Sys., selaku pembimbing I dan Ibu Anugrayani Bustamin, S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan memberikan bimbingan dalam penyusunan skripsi ini.
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku ketua Prodi Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas ilmu, bimbingan dan arahnya selama masa perkuliahan penulis.

5. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.
6. Teman-teman Laboratorium AIMP *Research Group* FT-UH yang telah memberikan bantuan selama penelitian dan diskusi mengenai penyusunan skripsi.
7. Teman-teman igniter16 atas dukungan dan semangat yang telah diberikan.
8. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Akhir kata, penulis berharap semoga Tuhan Yang Maha Esa membalas segala kebaikan dari semua pihak yang telah membantu penulis dalam penyusunan skripsi ini, dan semoga skripsi ini dapat memberikan hal yang bermanfaat serta menambah wawasan ilmu untuk pembaca dan juga bagi penulis sendiri.

Makassar, Juni 2021

Penulis

ABSTRAK

Sektor pertanian merupakan tulang punggung dari perekonomian dan pembangunan di Indonesia. Salah satu masalah yang dihadapi petani yaitu cara mengatasi serangan penyakit terhadap tanaman padi mereka. Penyakit pada tanaman padi dapat menurunkan kualitas beras bahkan dapat menyebabkan kematian pada padi. Tanaman padi yang terinfeksi akan menampilkan gejala berupa bercak yang memiliki pola dan warna tertentu pada beberapa bagian dari tubuh padi. Daun adalah bagian yang paling mudah diidentifikasi, sebab memiliki penampang yang lebih luas dibandingkan bagian tanaman padi yang lain. Identifikasi gejala awal pada daun padi masih dilakukan secara konvensional dengan cara melihat langsung pada daun padi. Hal ini dapat menimbulkan kesalahan dalam mengidentifikasi karena gejala awal yang ditimbulkan beberapa penyakit memiliki kemiripan dalam segi warna maupun bentuk bercak pada daun padi. Dengan bantuan *artificial intelligence*, identifikasi dan klasifikasi penyakit padi dapat dilakukan dengan efisien serta dapat mengurangi kesalahan dalam indentifikasi penyakit padi. Pada penelitian ini, dibuat sistem untuk mengklasifikasikan penyakit pada tanaman padi dari sampel gambar daun padi. Sistem ini menggunakan *deep learning* dengan menggunakan algoritma *Convolutional Neural Network*. Arsitektur *VGG16* yang digunakan pada penelitian ini telah dimodifikasi, kemudian *training* menggunakan 192 data citra dengan 4 label kelas dari penyakit *blast*, *blight*, *tungro* dan *healthy*. Parameter dalam proses *training* menggunakan konfigurasi *batch size* 16, *epoch* 30 dan *learning rate* 0,0001. Proses *training* ini menghasilkan model yang optimal pada proses validasi dan akan digunakan pada pengujian. Model kemudian diuji menggunakan 96 data *testing* mendapatkan rata-rata akurasi sebesar 97,6% dalam lima kali percobaan.

Kata kunci: Klasifikasi penyakit padi, *Deep Learning*, *VGG16*.

ABSTRACT

The agricultural sector is the backbone of the economy and development in Indonesia. One of the problems faced by farmers is how to overcome disease attacks on their rice plants. Diseases in rice plants can reduce the quality of rice and can even cause death in rice. Infected rice plants will show symptoms in the form of spots that have certain patterns and colors on some parts of the rice body. Leaves are the most easily identifiable part, because they have a wider cross-section than other parts of the rice plant. Identification of early symptoms on rice leaves is still done conventionally by looking directly at rice leaves. This can lead to errors in identification because the initial symptoms caused by several diseases have similarities in terms of color and shape of spots on rice leaves. With the help of artificial intelligence, identification and classification of rice diseases can be done efficiently and can reduce errors in rice disease identification. In this study, a system was developed to classify diseases in rice plants from rice leaf image samples. This system uses deep learning by using the Convolutional Neural Network algorithm. The VGG16 architecture used in this study has been modified, then trained using 192 image data with 4 class labels from blast, blight, tungro and healthy disease. Parameters in the training process use a configuration of batch size 16, epoch 30 and a learning rate of 0.0001. This training process produces an optimal model in the validation process and will be used in testing. The model was then tested using 96 testing data to get an average accuracy of 97.6% in five trials.

Keywords: rice disease classification, *deep learning*, *VGG16*.

DAFTAR ISI

KATA PENGANTAR	ii
ABSTRAK.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ixx
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penyakit Padi.....	6
2.1.1 Hawar Daun Bakteri (HDB atau <i>blight</i>)	6
2.1.2 Blas (<i>blast</i>).....	7
2.1.3 Tungro	7
2.2 Pengolahan Citra Digital	8
2.2.1 Citra Digital.....	8
2.2.2 Citra <i>Grayscale</i>	9
2.2.3 Citra Warna	10

2.2.4 <i>Thresholding</i>	11
2.3 <i>Deep Learning</i>	11
2.3.1 <i>Activation Function</i>	14
2.3.2 <i>Convolutional Neural Network</i>	16
2.3.3 <i>Cross Entropy Loss Function</i>	28
2.3.4 <i>Transfer Learning</i>	28
2.4 <i>Confusion Matrix</i>	30
2.5 Penelitian Terkait	32
BAB III METODOLOGI PENELITIAN	35
3.1 Tahapan Penelitian	35
3.2 Waktu dan Lokasi Penelitian	36
3.3 Instrumen Penelitian.....	36
3.4 Teknik Pengambilan Data	37
3.5 Perancangan Sistem	37
3.5.1 Tahapan <i>Training</i>	39
3.5.2 Strategi Evaluasi	46
BAB IV HASIL DAN PEMBAHASAN	50
4.1 Hasil Penelitian	50
4.2 Pembahasan.....	54
BAB V PENUTUP	61
5.1 Kesimpulan	61

5.2 Saran.....	61
DAFTAR PUSTAKA	62
LAMPIRAN.....	64

DAFTAR GAMBAR

Gambar 1. Contoh daun padi yang terkena penyakit <i>blight</i> (BPTP Kalimantan Tengah, 2015)	6
Gambar 2. Contoh daun padi yang terkena penyakit <i>blast</i> (BPTP Kalimantan Tengah, 2015)	7
Gambar 3. Contoh daun padi yang terkena penyakit tungro (BPTP Kalimantan Tengah, 2015)	8
Gambar 4. Citra Grayscale (MathWorks, 2018)	10
Gambar 5. Matriks citra warna (Pamungkas, 2017)	11
Gambar 6. Perbedaan antara lapisan layer pada Jaringan Saraf Tiruan dengan Jaringan <i>Deep Learning</i> (CS231n)	14
Gambar 7. Fungsi Linear (Sena, 2018)	15
Gambar 8. Fungsi <i>Sigmoid</i> (Sena, 2018)	15
Gambar 9. Fungsi ReLU (Sena, 2018)	16
Gambar 10. Struktur CNN (Patel & Pingel, 2017)	18
Gambar 11. <i>Convolutional Layer</i> (CS231n)	22
Gambar 12. <i>Max Pooling</i> (CS231n)	23
Gambar 13. <i>Processing of a Fully-Connected Layer</i> (CS231n)	25
Gambar 14. <i>Processing of Dropout</i> (Srivastava dkk, 2014)	26
Gambar 15. <i>The architecture of VGG16</i> (Loukadakis dkk, 2018)	30
Gambar 16. Diagram Alur Penelitian	35
Gambar 17. Tahapan <i>Training</i> dan <i>Testing</i> Sistem	38
Gambar 18. Ilustrasi Rancangan Sistem	38
Gambar 19. Contoh data dari 4 kelas	39

Gambar 20. Contoh hasil <i>augmentation</i> pada salah satu data <i>train</i>	41
Gambar 21. Potongan <i>source code data augmentation</i>	41
Gambar 22. Arsitektur (a) <i>VGG16</i> (Khandelwal, 2020) (b) <i>VGG16</i> modifikasi.....	42
Gambar 23. Paramater yang digunakan untuk <i>training</i>	45
Gambar 24. Arsitektur <i>VGG16</i> yang telah dimodifikasi	46
Gambar 25. Potongan <i>source code</i> untuk <i>predict image</i>	47
Gambar 26. Perbandingan grafik <i>training accuracy</i> (a) <i>VGG16</i> (b) <i>VGG16</i> modifikasi	50
Gambar 27. <i>Confusion matrix</i> pada <i>data testing</i> menggunakan <i>VGG16</i> yang telah dimodifikasi	52
Gambar 28. Tampilan <i>Interface</i> Sistem Klasifikasi Penyakit Tanaman Padi.....	60

DAFTAR TABEL

Tabel 1. Letak TP, TN, FP dan FN pada kelas A dalam <i>confusion matrix</i>	31
Tabel 2. Jumlah <i>dataset</i> pada setiap kelas	40
Tabel 3. <i>Confusion matrix data testing</i>	47
Tabel 4. Perbandingan <i>Training Time VGG16</i>	51
Tabel 5. <i>Testing VGG16</i> modifikasi	52
Tabel 6. <i>Training VGG16</i> modifikasi dengan <i>Image enhacement</i>	53
Tabel 7. Perbandingan Penelitian Terkait dengan Penelitian yang dilakukan.....	53
Tabel 8. Perbandingan nilai <i>Epoch</i> saat <i>training</i>	55
Tabel 9. Perbandingan nilai <i>Learning rate</i> saat <i>training</i>	55
Tabel 10. Hasil kesalahan klasifikasi.....	57
Tabel 11. Hasil klasifikasi benar.....	58

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sektor pertanian di Indonesia merupakan tulang punggung dari perekonomian dan pembangunan nasional. Hal tersebut dapat dilihat dalam penyerapan tenaga kerja, penyediaan pangan, dan penyediaan bahan baku industri. Sektor pertanian juga telah menjadi salah satu pembentuk budaya bangsa dan penyeimbang ekosistem. Beras adalah salah satu produk makanan pokok paling penting di Indonesia. Berdasarkan data dari *Food and Agriculture Organization of the United Nations* dan Kementerian Agrikultur mencatat 79,2 ton produksi beras di Indonesia pada tahun 2016 hingga saat ini masih menempati posisi ketiga di dunia dalam hal produksi beras. Adapun Sulawesi Selatan menempati posisi keempat terbanyak di Indonesia dalam hal produksi beras berdasarkan data dari Badan Pusat Statistik tahun 2019 (Badan Pusat Statistik, 2020).

Salah satu masalah yang dihadapi petani secara umum yaitu cara mengatasi serangan penyakit terhadap tanaman padi mereka. Gejala penyakit tanaman padi tidak separah yang ditimbulkan oleh serangan hama, namun dampak yang dihasilkan sama-sama memberikan kerugian. Meskipun penyakit pada tanaman padi ini musiman tapi dapat mengakibatkan tanaman rusak dan terserang penyakit sehingga mengakibatkan jumlah produksi berkurang.

Penyakit pada tanaman padi dapat menurunkan kualitas beras yang dihasilkan oleh padi, bahkan dapat menyebabkan kematian pada padi. Tanaman padi yang terinfeksi penyakit akan menampilkan gejala berupa bercak yang

memiliki pola dan warna tertentu pada beberapa bagian dari tubuh padi. Gejala tersebut dapat dilihat dari beberapa bagian padi seperti malai, daun maupun batang. Daun padi adalah bagian yang paling mudah diidentifikasi, sebab daun padi memiliki penampang yang lebih luas dibandingkan bagian tubuh tanaman padi yang lain sehingga perubahan warna dan bentuk bercak dapat terlihat jelas. Oleh karena itu, bagian daun dapat digunakan untuk mengidentifikasi penyakit pada padi.

Gejala penyakit pada padi penting untuk diidentifikasi lebih awal secara akurat agar dapat menemukan solusi untuk dilakukan penanganan. Salah satu cara yang dapat dilakukan dengan mengidentifikasi bentuk dan warna bercak pada daun padi. Identifikasi gejala awal pada daun padi masih dilakukan secara konvensional dengan cara melihat langsung pada daun padi. Hal ini dapat menimbulkan kesalahan dalam mengidentifikasi suatu penyakit padi sebab gejala awal yang ditimbulkan beberapa penyakit memiliki kemiripan dalam segi warna maupun bentuk bercak pada daun padi. Oleh karena itu harus diteliti lebih lanjut agar tidak melakukan kesalahan dalam penanganannya. Pengerjaan dilakukan secara manual untuk mengidentifikasi penyakit pada padi, maka sangat bergantung pada keahlian serta pengalaman manusia. Untuk mengatasi kendala itu diusulkan penggunaan teknologi *deep learning* untuk mendeteksi penyakit pada padi. *Deep Learning* adalah bagian dari keluarga metode *Machine Learning* yang merupakan pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas seperti deteksi objek, klasifikasi, pengenalan suara, terjemahan bahasa dan lain-lain. Beberapa peneliti telah melakukan percobaan mendeteksi berbagai penyakit maupun

klasifikasi berdasarkan warna dengan metode *deep learning* dan mendapat akurasi yang sangat memuaskan. Untuk itu penulis mengusulkan penggunaan *deep learning* untuk mendeteksi penyakit pada padi dengan harapan akan memberikan akurasi yang lebih tinggi agar tepat dalam menangani penyakit padi tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana cara mengklasifikasi penyakit pada tanaman padi dengan metode *deep learning*?
2. Bagaimana unjuk kerja akurasi sistem dalam klasifikasi penyakit pada tanaman padi menggunakan metode *deep learning*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah pada sub-bab sebelumnya maka dapat dibuat tujuan sebagai berikut:

1. Untuk mengimplementasikan metode *deep learning* pada proses pengklasifikasian penyakit tanaman padi.
2. Untuk mengukur tingkat akurasi metode *deep learning* dalam pengklasifikasian penyakit pada tanaman padi.

1.4 Manfaat Penelitian

Dengan dilakukannya penelitian ini, manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi masyarakat, penelitian ini dapat digunakan sebagai alat bantu untuk

mengklasifikasikan penyakit pada tanaman padi agar dapat mengurangi kesalahan dalam mengidentifikasi penyakit sehingga dapat ditindaklanjuti dengan penanganan yang tepat.

2. Bagi peneliti, penelitian ini dapat dijadikan referensi dalam menggunakan metode *deep learning* dalam mengklasifikasikan penyakit pada tanaman padi.
3. Bagi institusi pendidikan, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian-penelitian selanjutnya.

1.5 Batasan Masalah

Yang menjadi batasan masalah dari penelitian ini adalah sebagai berikut:

1. Objek penelitian difokuskan pada daun padi.
2. Penyakit yang akan diteliti yaitu penyakit blas (*blast*), hawar daun bakteri (*blight*) dan tungro.
3. Sistem klasifikasi dibuat menggunakan arsitektur *VGG16*.

1.6 Sistematika Penulisan

Adapun sistematika penulisan pada penelitian ini adalah:

BAB I PENDAHULUAN

Bab ini berisi penjelasan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori terkait hal-hal yang mendasari dan yang berhubungan dengan penelitian, termasuk di dalamnya penyakit pada tanaman padi, proses *deep learning* dan metode-metode yang digunakan dalam penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang apa saja yang akan dilakukan pada saat penelitian yang meliputi tahapan, waktu dan lokasi, instrumen penelitian, perancangan sistem, dan analisis kerja sistem.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang hasil penelitian dan pembahasan terkait pengolahan data yang telah dilakukan yang disertai dengan hasil penelitian.

BAB V PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian yang dilakukan serta saran-saran untuk pengembangan sistem yang lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Penyakit Padi

Penyakit pada tanaman padi dapat menurunkan kualitas beras yang dihasilkan oleh padi, bahkan menyebabkan kematian pada padi. Tanaman penyakit yang terinfeksi penyakit, akan menampilkan gejala berupa bercak yang memiliki pola dan warna tertentu pada beberapa bagian tubuh padi. Berikut merupakan beberapa jenis penyakit pada tanaman padi yang akan diteliti (Zahrah dkk, 2016):

2.1.1 Hawar Daun Bakteri (HDB atau *blight*)

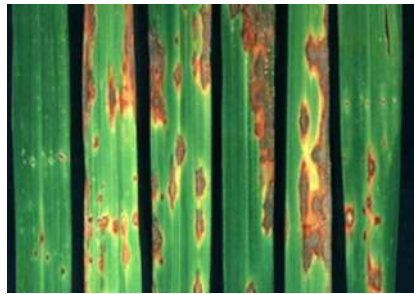
Penyebab penyakit hawar daun bakteri atau *blight* adalah bakteri *Xanthomonas campestris* pv. *Oryzae*, yang tersebar hampir diseluruh daerah pertanaman padi baik di dataran rendah maupun dataran tinggi dan selalu timbul pada musim kemarau maupun hujan. Kerugian hasil yang disebabkan oleh penyakit hawar daun bakteri dapat mencapai 60%. Kerusakan terberat terjadi apabila penyakit menyerang tanaman muda, dapat menyebabkan tanaman mati (Bogorkab.go.id, 2014).



**Gambar 1. Contoh daun padi yang terkena penyakit *blight* (BPTP
Kalimantan Tengah, 2015)**

2.1.2 Blas (*blast*)

Penyakit blas disebabkan oleh jamur *Pyricularia, sp.* Spora disebarkan melalui benih (*seed borne*) dan angin (*airborne*). Penyakit Blas menginfeksi tanaman padi pada setiap fase pertumbuhan menyerang daun, batang dan malai padi kecuali akar. *Leaf blast* (blas daun) berbentuk seperti belah ketupat, dimulai dengan bercak kecil ungu pada daun muda, kemudian berkembang menjadi belah ketupat, dapat menyebabkan tanaman kerdil dan pertumbuhan malai yang kecil/abnormal (BPTP Kalimantan Tengah, 2015).



Gambar 2. Contoh daun padi yang terkena penyakit *blast* (BPTP Kalimantan Tengah, 2015)

2.1.3 Tungro

Tungro adalah penyakit virus pada padi yang menyebabkan tanaman tumbuh kerdil dan berkurangnya jumlah anakan. Pelepah dan helaian daun memendek dan daun yang terserang sering berwarna kuning sampai kuning-oranye. Wereng hijau adalah serangga utama yang menyebarkan virus tungro (BPTP Kalimantan Tengah, 2015).



**Gambar 3. Contoh daun padi yang terkena penyakit tungro (BPTP
Kalimantan Tengah, 2015)**

2.2 Pengolahan Citra Digital

Pengolahan citra digital merupakan metode yang digunakan untuk mengolah citra pada komputer sehingga dapat menghasilkan gambar sesuai dengan yang dibutuhkan. Misalnya terdapat sebuah citra digital berwarna dengan ukuran 1280x720 piksel. Dengan pengolahan citra digital, gambar tersebut dapat diubah ukurannya menjadi 640x360 piksel tanpa mengurangi kualitas gambar. Citra merupakan fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Secara matematis, fungsi ini dapat dilambangkan dengan $f(x,y)$ dimana (x,y) merupakan koordinat pada bidang dua dimensi tersebut, dan $f(x,y)$ merupakan intensitas cahaya pada titik (x,y) . Ukuran terkecil dalam citra digital biasa disebut *picture element* atau *pixel* (Syawaluddin, 2016).

2.2.1 Citra Digital

Citra adalah kombinasi antara titik, garis, bidang, dan warna yang mewakili suatu objek atau benda. Sedangkan citra digital merupakan keluaran yang dihasilkan melalui perangkat pencitraan digital seperti kamera dan dapat disimpan dan diolah komputer.

Citra perlu diterjemahkan menjadi angka-angka yang dapat dipahami oleh komputer, maka citra dibagi menjadi bagian-bagian kecil yang disebut dengan piksel.

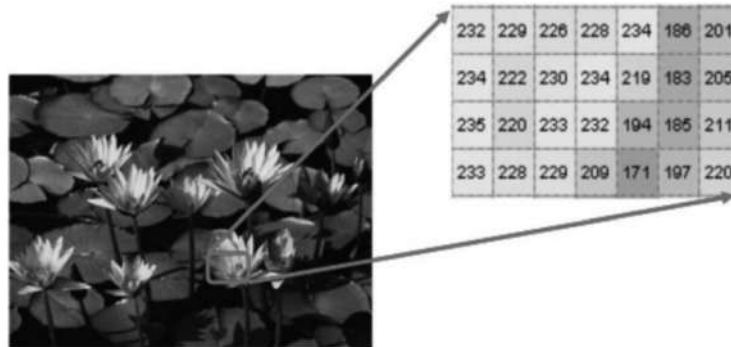
Dalam setiap piksel, perangkat pencitraan merekam sebuah angka ataupun sekumpulan angka yang merepresentasikan beberapa properti pada piksel yang dituju, seperti intensitas cahaya maupun warna, angka-angka ini diatur dalam sebuah baris-baris dan kolom-kolom array yang merujuk pada posisi vertikal dan horizontal dari piksel-piksel yang ada dalam citra. Representasi citra jika dibuat dalam bentuk matriks, dapat dilihat pada Persamaan 1 (Syawaluddin, 2016).

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

2.2.2 Citra *Grayscale*

Sesuai dengan namanya, jenis citra ini memiliki gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan 255 menyatakan putih. Citra RGB dengan matriks penyusun citra yang sebelumnya 3 matriks akan berubah menjadi 1 matriks saja. Citra *grayscale* merupakan citra yang nilai intensitas piksel nya didasarkan pada derajat keabuan. Pada citra *grayscale* 8-bit, derajat warna hitam sampai dengan putih dibagi ke dalam 256 derajat keabuan dimana warna hitam sempurna direpresentasikan dengan nilai 0 dan putih sempurna dengan

nilai 255. Citra RGB dapat dikonversi menjadi citra *grayscale* sehingga dihasilkan hanya satu kanal warna (Andono, 2017).

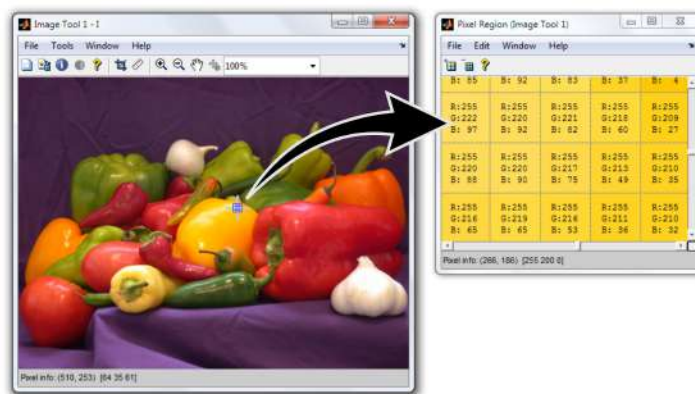


Gambar 4. Citra Grayscale (MathWorks, 2018)

2.2.3 Citra Warna

Dalam citra warna, setiap piksel yang terdapat di dalamnya mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Biasanya warna-warna dasar ini disebut sebagai *channel*. Setiap warna dasar menggunakan penyimpanan $8 \text{ bit} = 1 \text{ byte}$, yang berarti setiap warna mempunyai gradasi 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 \times 2^8 \times 2^8 = 2^{24} = 16 \text{ juta}$ warna lebih. Itu sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

Penyimpanan *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* yang terdiri atas 256 gradasi warna hanya diwakili oleh 1 *byte*. Sedangkan 1 piksel citra *true color* diwakili oleh 3 *byte*, di mana masing-masing *byte* merepresentasikan tiap R, G, dan B (Andono, 2017).



Gambar 5. Matriks citra warna (Pamungkas, 2017)

2.2.4 Thresholding

Thresholding merupakan salah satu teknik dasar dalam melakukan segmentasi citra. *Thresholding* digunakan untuk mengonversi citra *grayscale* menjadi citra biner. Citra biner hanya terdiri dari dua intensitas warna yaitu hitam yang memiliki representasi nilai 0 dan putih yang memiliki nilai representasi nilai 1. Sehingga jenis citra ini hanya membutuhkan 1bit memori untuk menyimpan kedua memori ini.

Thresholding secara umum digunakan untuk memisahkan beberapa objek target dari latar belakangnya. Objek target biasanya direpresentasikan dengan warna putih (Sutoyo dkk, 2009).

2.3 Deep Learning

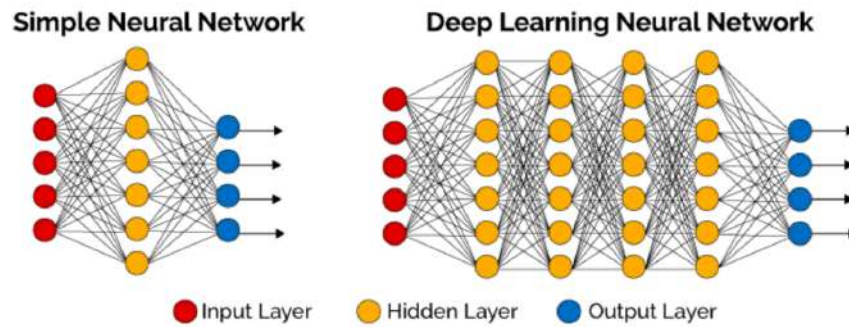
Deep Learning adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linier yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-*

supervised learning) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. *Deep Learning* disebut sebagai *Deep* (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan. *Deep Learning* adalah salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input dan mengolahnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi non linier dari data masukan untuk menghitung nilai output. Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*).

Jaringan saraf yang dimiliki oleh *Deep Learning* terbentuk dari hirarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi layer*). Berdasarkan hal itulah *Deep Learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi non-linier. *Deep Learning* bekerja berdasarkan pada arsitektur jaringan dan prosedural optimal yang digunakan pada arsitektur. Setiap *output* dari lapisan per lapisan yang tersembunyi dapat dipantau dengan menggunakan grafik khusus yang dirancang untuk setiap *output* neuron.

Kombinasi dan rekombinasi dari setiap neuron yang saling terhubung dari semua unit lapisan tersembunyi dilakukan menggunakan gabungan dari fungsi aktivasi. Prosedur-prosedur tersebut dikenal sebagai Transformasi Non-Linier yang digunakan untuk prosedur optimal untuk menghasilkan bobot optimal pada setiap unit lapisan guna mendapatkan nilai target yang dibutuhkan.

Dalam proses perancangan, apabila jumlah saraf yang ditambahkan sangat banyak, hal tersebut tidak akan pernah cocok untuk menyelesaikan setiap masalah. Persoalan terpenting dalam *Deep Learning* adalah jaringan sarafnya dilatih dengan cara penurunan gradien secara sederhana. Pada saat kita menambahkan lapisan jaringan yang semakin banyak, maka sebaliknya penurunan dari gradien semakin berkurang sehingga dapat mempengaruhi nilai outputnya. Jaringan Saraf Tiruan adalah jaringan saraf yang biasanya menggunakan jaringan seperti umpan maju (*feed forward*) atau *recurrent network* yang hanya memiliki 1 atau 2 lapisan tersembunyi. Tetapi, jika lapisan jaringan sarafnya lebih dari 2 *layer* atau bahkan mencapai ratusan lapisan itulah yang disebut sebagai *Deep Learning*. Pada Jaringan Saraf Tiruan, arsitektur jaringan yang dimilikinya kurang kompleks dan membutuhkan lebih banyak informasi tentang data input sehingga dapat menentukan algoritma mana yang dapat digunakan yang dapat dilihat pada Gambar 6. Dalam Jaringan Saraf Tiruan terdiri dari beberapa algoritma yaitu *Model Hebb*, *Perceptron*, *Adaline*, *Forward Propagation*, dll. Sedangkan pada algoritma jaringan saraf *Deep Learning* tidak memerlukan informasi apapun terhadap data yang akan dipelajarinya, dan algoritmanya dapat secara mandiri melakukan tuning (penyetelan) dan pemilihan model yang paling optimal (Wayan Dadang, 2018).



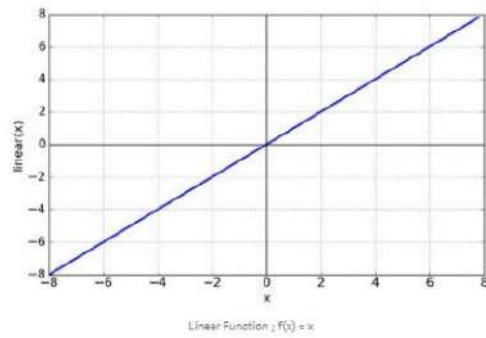
Gambar 6. Perbedaan antara lapisan layer pada Jaringan Saraf Tiruan dengan Jaringan *Deep Learning* (CS231n)

2.3.1 *Activation Function*

Activation Function merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas internal (*summation function*) yang mungkin berbentuk linear ataupun non-linier. Fungsi ini bertujuan untuk menentukan apakah neuron diaktifkan atau tidak. Ada beberapa fungsi aktivasi yang sering digunakan dalam *Neural Network*, yaitu sebagai berikut (Sena, 2018):

1. Fungsi Aktivasi Linear

Fungsi aktivasi linear merupakan fungsi yang memiliki nilai *output* yang sama dengan nilai inputnya. Hal ini berkaitan dengan, jika sebuah neuron menggunakan *linear activation*, maka keluaran dari neuron tersebut adalah *weighted sum* dari input + bias. Grafik fungsi linear ditunjukkan oleh gambar berikut.

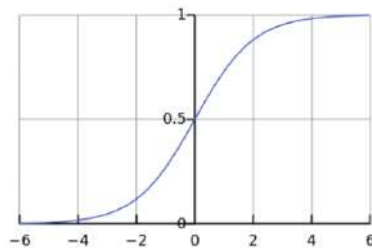


Gambar 7. Fungsi Linear (Sena, 2018)

2. Fungsi Aktivasi *Sigmoid*

Fungsi aktivasi *sigmoid* merupakan fungsi non-linear. Masukan untuk fungsi aktivasi ini berupa bilangan riil dan *output* dari fungsi tersebut memiliki nilai antara 0 sampai 1. Berikut ini grafik fungsi aktivasi *sigmoid*:

$$A = \frac{1}{1+e^{-x}}$$

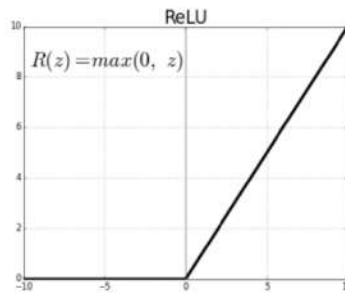


Gambar 8. Fungsi *Sigmoid* (Sena, 2018)

Jika input dari suatu node pada *neural network* bernilai negatif maka keluaran yang didapatkan adalah 0, sedangkan jika masukannya bernilai positif maka keluaran nilainya adalah 1. Kelemahan dari fungsi ini adalah kemungkinan besar *sigmoid* mematikan *gradient* (*gradient* bernilai 0). Hal ini dapat terjadi ketika neuron mengeluarkan nilai antara 0 dan 1. Selain itu, *output* dari *sigmoid* tidak berpusat pada 0 (*zero-centered*).

3. Activation ReLU

Pada dasarnya fungsi ReLU (*Rectified Linear Unit*) melakukan *threshold* dari 0 hingga *infinity*. Fungsi ini menjadi salah satu fungsi yang populer saat ini. Berikut ini grafik fungsi aktivasi ReLU.



Gambar 9. Fungsi ReLU (Sena, 2018)

Pada fungsi ini, jika masukan dari neuron-neuron berupa bilangan negatif, maka fungsi ini akan menerjemahkan nilai tersebut kedalam nilai 0, jika masukan bernilai positif maka *output* dari neuron adalah nilai aktivasi itu sendiri. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan dengan fungsi *sigmoid* dan *tan h*. Namun aktivasi ini juga memiliki kelemahan yaitu aktivasi ini bisa menjadi rapuh pada proses *training* dan bisa membuat unit tersebut mati.

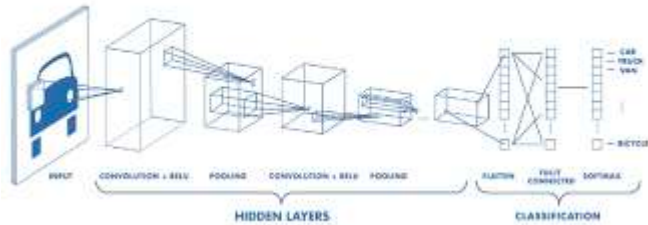
2.3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma dari *deep learning* yang merupakan hasil pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk melakukan olah data

menjadi bentuk dua dimensi, misalnya gambar atau suara. CNN digunakan untuk melakukan klasifikasi data yang berlabel dengan menggunakan metode *supervised learning* yang dimana terdapat data yang dilatih dan terdapat variabel yang ditargetkan sehingga tujuan dari metode ini yaitu mengelompokkan suatu data ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atas pemandangan dan melakukan deteksi dan melakukan segmentasi objek. CNN belajar langsung melalui data citra, sehingga dapat menghilangkan ekstraksi ciri dengan cara manual. Penelitian awal yang menjadi dasar penemuan ini yaitu pertama kali dilakukan oleh Hubel dan Wiesel yang melakukan penelitian *visual cortex* pada indera penglihatan kucing. *Visual cortex* pada hewan sangat *powerful* kemampuannya dalam sistem pemrosesan visual yang pernah ada. Sehingga, banyak penelitian yang terinspirasi oleh cara kerjanya dan menghasilkan banyak model-model baru yang beberapa diantaranya yaitu, *Neocognitron*, *HMAX*, *LeNet-5*, dan *AlexNet* (Hubel & Wiesel, 1968).

CNN juga merupakan saraf yang dikhususkan untuk memproses data yang memiliki struktur kotak (*grid*). Sebagai contoh yaitu berupa citra dua dimensi. Nama konvolusi merupakan operasi dari aljabar linear yang mengalikan matriks dari filter pada citra yang akan diproses. Proses ini disebut dengan lapisan konvolusi dan merupakan salah satu jenis dari banyak lapisan yang bisa dimiliki dalam suatu jaringan. Meskipun begitu, lapisan konvolusi ini merupakan lapisan utama yang paling penting digunakan. Jenis lapisan yang lain yang biasa digunakan adalah Pooling

Layer, yakni lapisan yang digunakan untuk mengambil suatu nilai maksimum atau nilai rata-rata dari bagian-bagian lapisan piksel pada citra. Berikut merupakan gambaran umum arsitektur CNN:



Gambar 10. Struktur CNN (Patel & Pingel, 2017)

Pada Gambar 10 yaitu di setiap lapisan input yang dimasukkan mempunyai susunan neuron 3 dimensi, yaitu lebar, tinggi, dan kedalaman. Lebar dan tinggi yaitu ukuran lapisan, sedangkan untuk ke dalam yaitu mengacu pada jumlah lapisan. Setiap besaran yang didapat tergantung dari hasil filtrasi dari lapisan sebelumnya dan banyaknya filter yang digunakan. Model jaringan seperti ini sudah terbukti efektif dalam menangani permasalahan klasifikasi citra. Sebuah CNN mampu memiliki puluhan hingga ratusan lapisan yang masing-masing lapisan mempelajari deteksi berbagai gambar. Pengolahan citra diterapkan pada setiap citra latih pada resolusi yang berbeda, dan *output* dari masing-masing data gambar yang diolah dan digunakan sebagai input ke lapisan berikutnya. Pengolahan citra dapat dimulai sebagai fitur yang sederhana, seperti ukuran kecerahan dan tepi atau meningkatkan kekompleksan pada fitur secara unik untuk menentukan objek sesuai ketebalan lapisan. Secara umum tipe lapisan CNN dibagi menjadi dua bagian, yaitu:

1. *Feature extraction layer*

Lapisan pada jenis pertama yaitu adalah *convolutional layer* dan lapisan kedua adalah *pooling layer*. Pada setiap lapisan diberlakukan fungsi aktivasi dengan posisinya yang berselang-seling antara jenis pertama dan jenis kedua. Lapisan ini menerima input gambar secara langsung dan memprosesnya sampai menghasilkan *output* berupa vektor untuk diolah pada lapisan berikutnya.

- *Convolutional Layer*

Tahap ini melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain dilakukan secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bertipe riil. Operasi ini menerapkan fungsi *output* sebagai *feature map* dari input citra. Input dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Operasi konvolusi dapat dituliskan sebagai berikut:

$$s(t) = (x * t)(t) = \sum_{\alpha}^{\infty} x(\alpha) * w(t - \alpha) \quad (2)$$

Keterangan:

$s(t)$ = Fungsi hasil operasi konvolusi

x = Input

w = bobot (kernel)

Fungsi $s(t)$ memberikan *output* tunggal berupa *feature map*. Argumen pertama adalah input yang merupakan x dan argumen kedua w sebagai kernel atau filter. Apabila input dilihat sebagai citra dua dimensi, maka t dapat dianggap sebagai piksel dan nilainya dapat dikonversi ke dalam nilai i dan j . Jadi sebuah operasi konvolusi dengan input yang mempunyai dimensi lebih besar dari satu dapat ditulis sebagai berikut.

$$s(i,j) = (K*I) (i,j) = \sum_{\infty} \sum_{\infty} I(i - m, j - n)K(m, n) \quad (3)$$

$$s(i,j) = (K*I) (i,j) = \sum_{\infty} \sum_{\infty} I(i + m, j + n)K(m, n) \quad (4)$$

Kedua persamaan di atas merupakan perhitungan dasar dalam operasi konvolusi, dengan i dan j adalah sebuah piksel dari citra. Perhitungan tersebut bersifat kumulatif dan muncul saat K sebagai kernel, kemudian I sebagai input dan kernel yang dapat dibalik relatif terhadap input. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai operasi perkalian matriks antara citra masukan dan kernel dimana keluarannya dihitung dengan operasi dot. Selain itu, penentuan volume *output* juga dapat ditentukan dari masing-masing lapisan dengan *hyperparameters*. *Hyperparameter* yang digunakan pada persamaan di bawah ini digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali *output*. Perhatikan Persamaan 5 berikut.

$$\frac{W - F + 2P}{S + 1} \quad (5)$$

Keterangan :

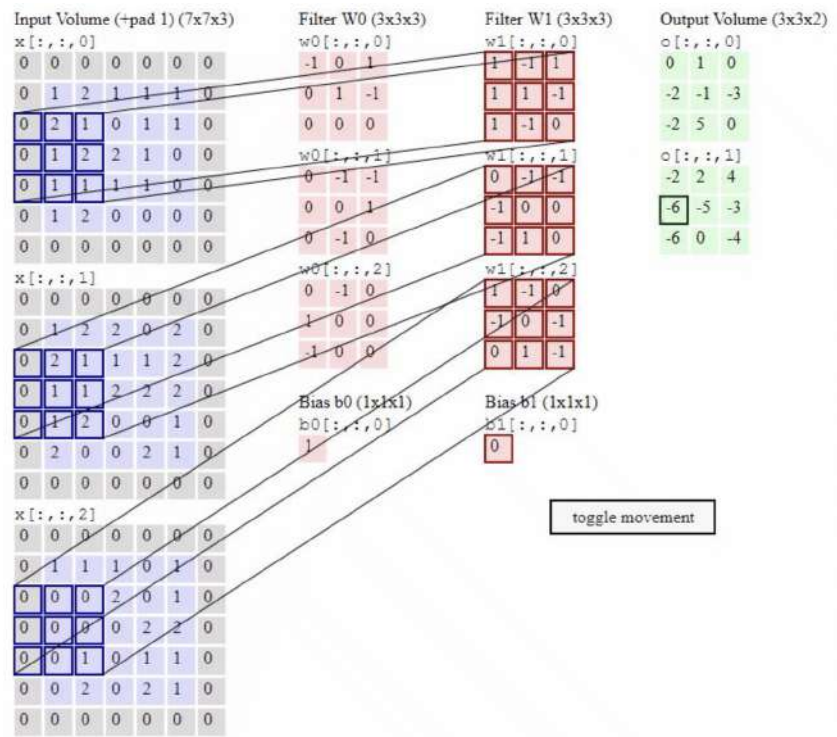
W = Ukuran volume gambar

F = Ukuran Filter

P = Nilai *Padding* yang digunakan

S = Ukuran Pergeseran (*Stride*)

Berdasarkan persamaan di atas, dapat dihitung ukuran spasial dari volume *output* dimana *hyperparameter* yang dipakai adalah ukuran volume (W), filter (F), *Stride* yang diterapkan (S) dan jumlah padding nol yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser filter melalui input citra dan *Zero Padding* adalah nilai untuk mendapatkan angka nol di sekitar border citra. Operasi *Convolutional Layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (piksel). Sebagai contoh, *layer* pertama pada *feature extraction layer* biasanya adalah *conv layer* dengan ukuran 5x5x3. Panjang 5 piksel, tinggi 5 piksel dan tebal/jumlah 3 buah sesuai dengan *channel* dari image tersebut. Ketiga filter ini akan digeser ke seluruh bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara input dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*. Perhatikan Gambar 11.

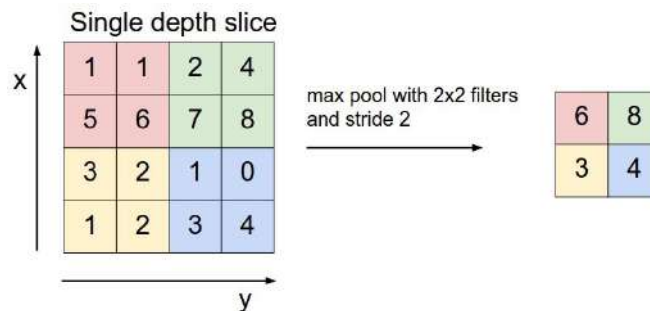


Gambar 11. Convolutional Layer (CS231n)

- Pooling Layer

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling Layer* biasanya berada setelah *conv layer*. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan di antara lapisan konvolusi secara berturut-turut dalam arsitektur CNN dapat secara progresif mengurangi

ukuran volume *output* pada *feature map*, sehingga mengurangi jumlah parameter pada jaringan untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja pada setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah (*stride*) sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Berikut ini adalah contoh gambar operasi *max-pooling*:



Gambar 12. Max Pooling (CS231n)

Gambar 12 di atas menunjukkan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan *pooling* akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Operasi *max-pooling* pada gambar di atas menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4. Dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2.

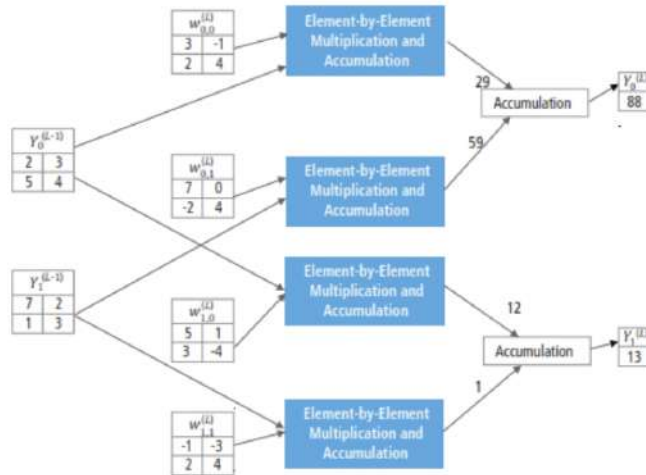
2. *Classification Layer*

Layer ini tersusun atas beberapa lapisan yang di setiap lapisan tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan lapisan yang lainnya. *Layer* ini menerima input dari hasil *output layer* ekstraksi fitur gambar berupa vektor yang kemudian ditransformasikan seperti pada *Multi Neural Network* dengan tambahan beberapa *hidden layer*. Hasil *output* berupa akurasi kelas untuk klasifikasi. Dengan ini, CNN merupakan metode untuk melakukan transformasi gambar asli lapisan per lapisan dari nilai piksel gambar ke dalam nilai skoring kelas untuk klasifikasi. Setiap lapisan ada yang memiliki *hyperparameter* dan ada yang tidak memiliki parameter (bobot dan bias pada neuron).

- *Fully-Connected Layer*

Fully-Connected Layer adalah sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya sama seperti halnya dengan *neural network* biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada input, sementara lapisan *fully-connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua

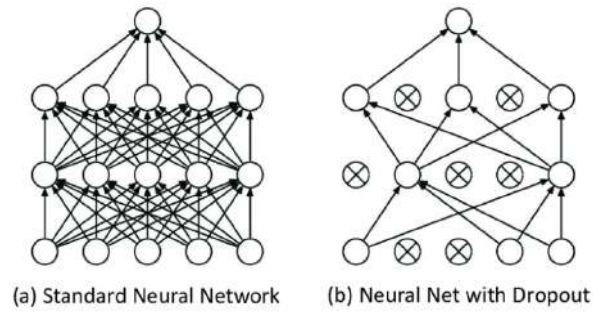
lapisan tersebut masih menggunakan operasi dot, sehingga fungsinya tidak begitu berbeda. Berikut ini adalah proses *fully-connected*:



Gambar 13. Processing of a Fully-Connected Layer (CS231n)

- **Dropout Regulation**

Dropout merupakan sebuah teknik regulasi jaringan saraf dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan *backpropagation*. Berikut adalah gambar proses *dropout* (Srivastava dkk, 2014):



Gambar 14. *Processing of Dropout* (Srivastava dkk, 2014)

Gambar 14(a) di atas merupakan jaringan saraf biasa yang memiliki dua *hidden layer*. Sedangkan pada bagian 14(b) merupakan jaringan saraf dengan menggunakan *dropout*. Dari gambar tersebut terlihat terdapat beberapa neuron aktivasi yang tidak dipakai lagi. Penggunaan teknik ini akan berdampak pada performa model dalam melatih serta mengurangi *overfitting*. Pada jaringan saraf tiruan biasa, jika y^l adalah nilai keluaran dari suatu lapisan l dan z^l adalah nilai masukan pada *layer* l dengan W_i dan b_i adalah bobot dan bias dari lapisan l pada unit ke i , maka perhitungan proses *feedforward* menggunakan fungsi aktivasi f dapat dilakukan dengan Persamaan 6 berikut.

$$\begin{aligned}
 z_i^{l+1} &= W_i^{(l+1)} y^l + b_i^{(l+1)} \\
 y_i^{l+1} &= f(z_i^{(l+1)})
 \end{aligned}
 \tag{6}$$

Sementara pada jaringan yang mengimplementasikan teknik *Dropout*, variable r^l melambangkan vektor sepanjang j yang menyimpan nilai yang diperoleh dari distribusi *Bernoulli*. Proses *feedforward* dilakukan dengan Persamaan 7.

$$\begin{aligned}
\tilde{y}^l &= r_j^l * y^l \\
z_i^{l+1} &= W_i^{(l+1)} y^l + b_i^{(l+1)} \\
y_i^{l+1} &= f(z_i^{(l+1)})
\end{aligned} \tag{7}$$

- ***Softmax Classifier***

Softmax Classifier adalah generalisasi dari fungsi logistik. *Output* dari *softmax* ini dapat digunakan untuk mewakili distribusi sebuah kategori. *Softmax function* digunakan dalam berbagai macam metode klasifikasi, contohnya *multinomial logistic regression*, *multiclass linear discriminant analysis*, *naive bayes classifier*, dan *neural network*. Secara spesifiknya fungsi ini biasa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*. Berikut adalah fungsi yang diberikan:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{8}$$

Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke- j pada vektor keluaran kelas. Argumen z adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi *Softmax*. *Softmax* juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. *Softmax* memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya

menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

2.3.3 Cross Entropy Loss Function

Loss Function merupakan salah satu komponen penting dalam *neural network*. *Loss* menggambarkan kemungkinan kesalahan yang dihasilkan oleh model. *Loss Function* yang baik adalah fungsi yang diharapkan menghasilkan *error* yang paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebenaran yang asli. *Categorical Cross Entropy* merupakan salah satu pilihan terbaik untuk menghitung nilai *loss* pada permasalahan *multi-class classification*. *Categorical Cross Entropy* (CE) biasa juga disebut *Softmax Loss* yang merupakan gabungan dari *softmax activation* dan *cross-entropy loss*. Berikut merupakan rumus *Categorical Cross Entropy* (Gomez, 2018).

$$CE = -\log \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} \right) \quad (9)$$

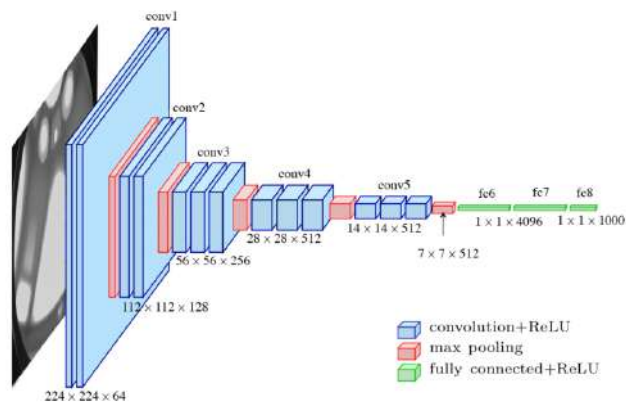
2.3.4 Transfer Learning

Transfer learning adalah suatu teknik atau metode yang memanfaatkan model yang sudah dilatih terhadap suatu *dataset* untuk menyelesaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai *starting point*, memodifikasi dan mengupdate parameternya sehingga sesuai dengan dataset yang baru (Sena, 2018).

- *VGG16 architecture*

VGG adalah singkatan dari *Visual Geometric Group* dari *Oxford University* dan *VGG16* adalah jaringan dengan 16 lapisan yang diusulkan oleh *Visual Geometric Group*. 16 lapisan ini berisi parameter yang dapat dilatih dan ada lapisan lain yang juga seperti *max pool layer*, tetapi lapisan tersebut tidak berisi parameter yang dapat dilatih. Arsitektur ini merupakan *runner up* pertama dari *Visual Recognition Challenge 2014* yaitu *ILSVRC-2014* dan dikembangkan oleh *Simonyan* dan *Zisserman*.

Kelompok riset *VGG* merilis serangkaian model jaringan konvolusi mulai dari *VGG11* hingga *VGG19*. Variasi *VGG* yang berbeda persis sama di tiga lapisan terakhir yang terhubung sepenuhnya. Struktur keseluruhan mencakup 5 set lapisan konvolusional, diikuti oleh *max pool*. Tetapi perbedaannya adalah bahwa saat kedalaman meningkat yaitu saat berpindah dari *VGG11* ke *VGG19*, semakin banyak lapisan konvolusional bertingkat ditambahkan dalam lima set lapisan konvolusional (Khandelwal, 2020).



Gambar 15. The architecture of VGG16 (Loukadakis dkk, 2018)

Pada Gambar 15 di atas, semua persegi panjang biru mewakili lapisan konvolusi bersama dengan fungsi aktivasi non-linier yang merupakan *rectified linear unit (ReLU)*. Seperti yang terlihat pada gambar bahwa terdapat 13 buah persegi panjang biru dan 5 buah merah yaitu 13 buah *convolution layer* dan 5 buah *max-pooling layer*. Pada bagian akhir, ada 3 persegi panjang hijau yang mewakili 3 lapisan yang terhubung sepenuhnya. Jadi, jumlah total lapisan yang memiliki parameter adalah 16 di antaranya 13 untuk lapisan konvolusi dan 3 untuk lapisan *fully-connected*, sehingga nama tersebut diberikan sebagai *VGG16*. Pada *output*, arsitektur ini memiliki lapisan *softmax* yang memiliki 1000 output per kategori gambar dalam dataset *imagenet*.

2.4 Confusion Matrix

Secara umum penentuan baik atau tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu *accuracy*, *recall*, dan *precision*. Untuk menghitung faktor-faktor tersebut

diperlukan sebuah matriks yang biasa disebut dengan *confusion matrix*. *Confusion matrix* adalah cara untuk memvisualisasikan kinerja model dalam bentuk tabel. Dalam pengukuran kinerja menggunakan *confusion matrix*, terdapat empat kondisi untuk merepresentasikan hasil proses klasifikasi, yaitu (Mohajon, 2020):

- *True Positive* (TP): Keadaan di mana *classifier* memprediksi dengan benar kelas positif sebagai positif.
- *True Negative* (TN): Keadaan di mana *classifier* memprediksi dengan benar kelas negatif sebagai negatif.
- *False Positive* (FP): Keadaan di mana *classifier* salah, yaitu memprediksi kelas negatif sebagai positif.
- *False Negative* (FN): Keadaan di mana *classifier* salah, yaitu memprediksi kelas positif sebagai negatif.

Penelitian ini merupakan *multi-class classification* yang mana memiliki 4 kelas keluaran, berikut merupakan contoh penempatan TP, TN, FP dan FN pada salah satu kelas dalam *confusion matrix*.

Tabel 1. Letak TP, TN, FP dan FN pada kelas A dalam *confusion matrix*

		Predicted			
		A	B	C	D
True	A	TP		FN	
	B				
	C	FP		TN	
	D				

Untuk menghitung nilai akurasi sistem dapat menggunakan rumus pada Persamaan 10.

$$Accuracy = \frac{TP+FN}{TP+FP+TN+FN} \quad (10)$$

2.5 Penelitian Terkait

Berikut ini merupakan beberapa penelitian yang berkaitan dengan penelitian yang akan dilakukan:

2.5.1 *Detection and Classification of Rice Plant Diseases* (Prajapati dkk, 2016).

Pada tahun 2016, Prajapati dkk melakukan penelitian dengan judul *Detection and Classification of Rice Plant Diseases*. Mereka meneliti 3 jenis penyakit yaitu *bactery leaf blight*, *brown spot* dan *leaf smut*. Penelitian tersebut menggunakan *Multi Support Vector Machine* (SVM), fitur yang diekstrak yaitu *color*, *shape* dan *texture*. Adapun hasil yang didapatkan yaitu 93,33% akurasi pada data *train* dan 73,33% pada data *test*.

2.5.2 Identifikasi Gejala Penyakit Padi Menggunakan Operasi Morfologi Citra (Zahrah dkk, 2016).

Zahrah dkk melakukan penelitian pada 2016 mengenai Identifikasi Gejala Penyakit Padi. Penelitian ini menggunakan metode operasi morfologi citra untuk mengidentifikasi bentuk gejala penyakit padi. Data citra melalui proses *smoothing* dan deteksi tepi, dengan teknik *Gaussian*, dan deteksi tepi Sobel. Kemudian, bentuk gejala penyakit diidentifikasi dengan operasi morfologi closing. Selanjutnya,

latar dan objek data citra dipisahkan menggunakan operasi *thresholding*. Data yang digunakan dalam penelitian ini adalah data citra daun padi sebanyak 20 data. Dari 20 data citra padi tersebut, menghasilkan 12 data citra yang dapat diidentifikasi bentuk dari gejala penyakit padi, sedangkan 8 data citra lainnya memerlukan perlakuan tambahan agar mendapatkan hasil yang diinginkan.

2.5.3 Sistem Identifikasi Penyakit Tanaman Padi Dengan Menggunakan Metode *Forward Chaining* (Wulandari dkk, 2015).

Pada penelitian ini dirancang sistem pakar berbasis web dengan metode inferensi forward chaining dan faktor kepastian (*certainty factor*). Data yang diolah dalam sistem ini adalah sebanyak 16 penyakit dengan 26 gejala penyakit. Sistem yang telah dikembangkan mempunyai keunggulan dalam kemudahan akses dan kemudahan pemakaian serta menampilkan tingkat kepastian dari hasil diagnosa dengan persentasi kepuasan pengguna sebesar 81%.

2.5.4 *Identification of Rice Diseases using Deep Convolutional Neural Networks* (Yang Lu dkk, 2017).

Dalam penelitian ini, menggunakan 500 data gambar daun dan batang padi yang sakit dan sehat. Data yang diambil kemudian dilatih menggunakan CNN untuk mengidentifikasi 10 jenis penyakit padi. Dengan menggunakan strategi *10-fold cross validation*, model berbasis CNN yang diusulkan mencapai akurasi sebesar 95,48%. Akurasi ini jauh lebih tinggi daripada model konvensional *machine learning*.

2.5.5 Rice Blast Disease Recognition Using a Deep Convolutional Neural Network (Wan-jie Liang dkk, 2018).

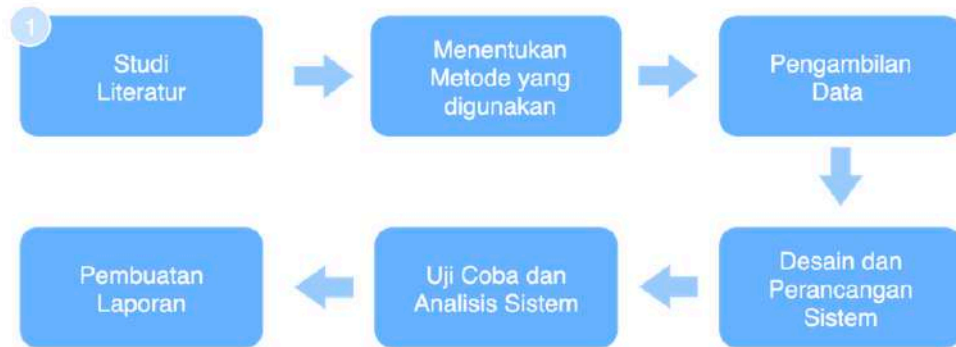
Penelitian ini menggunakan 2906 sampel positif dan 2902 sampel negatif untuk pelatihan dan pengujian model CNN. Hasil evaluasi menunjukkan bahwa fitur tingkat tinggi yang diekstraksi oleh CNN lebih diskriminatif dan efektif daripada fitur kerajinan tangan tradisional termasuk *Local Binary Patterns Histograms* (LBPH) dan Haar-WT (*Wavelet Transform*). Selain itu, hasil evaluasi kuantitatif menunjukkan bahwa CNN dengan *Softmax* dan CNN dengan support vector machine (SVM) memiliki performa yang sama. Hasil akurasi tertinggi yang didapat yaitu menggunakan CNN+SVM sebesar 95,82%.

BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan pada penelitian ini sebagaimana ditunjukkan pada Gambar 16.



Gambar 16. Diagram Alur Penelitian

Tahapan Penelitian pada Diagram dijelaskan sebagai berikut.

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahap ini digunakan untuk mengumpulkan penelitian terkait penggunaan teknologi *deep learning* untuk melakukan klasifikasi terhadap penyakit pada daun padi.
2. Menentukan metode *deep learning* yang digunakan yaitu menggunakan *Convolutional Neural Network* (CNN).
3. Pengambilan data terbagi atas 2 yaitu untuk data *training* dan *testing* menggunakan data yang diambil secara langsung oleh penulis menggunakan kamera Nikon D5100 pada daun padi yang terkena penyakit.
4. Perancangan sistem dilakukan dengan pembuatan rancangan alur kerja sistem. Implementasi sistem dilakukan sesuai dengan rancangan alur kerja

sistem yang telah dibuat. Sistem dibuat dengan menggunakan bahasa Pemrograman Python dan Algoritma CNN.

5. Uji coba sistem dilakukan untuk mengetahui seberapa akurat sistem yang dibuat.
6. Tahap akhir dari penelitian ini adalah melakukan penulisan laporan penelitian dalam bentuk skripsi sebagai bahan publikasi.

3.2 Waktu dan Lokasi Penelitian

Waktu penelitian dimulai sejak disetujuinya proposal penelitian ini pada bulan Agustus 2020 hingga proses penulisan pada bulan Januari 2021. Penelitian ini dilakukan di Laboratorium Kecerdasan Buatan Departemen Teknik Informatika Universitas Hasanuddin. Lokasi pengambilan data citra dilakukan di Desa Pombulaa Jaya Kabupaten Konawe Selatan, Sulawesi Tenggara dan di Dusun Bangkala, Pattalassang, Kabupaten Gowa, Sulawesi Selatan.

3.3 Instrumen Penelitian

Instrumen yang digunakan pada penelitian ini adalah:

1. Software
 - macOS 11.0
 - Google Colab
 - Python 3.7
2. Hardware
 - Macbook Air 2020, M1 13”, RAM 8GB
 - Nikon D5100

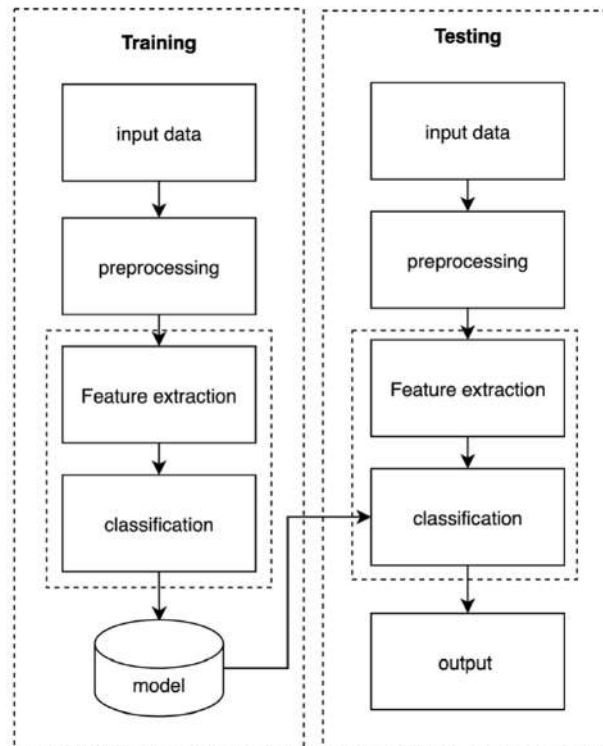
3.4 Teknik Pengambilan Data

Data yang digunakan pada penelitian ini berupa citra daun pada tanaman padi. Data citra diperoleh dari proses pengambilan data secara langsung yaitu sebanyak 320 Citra, data diambil menggunakan kamera Nikon D5100 dengan bantuan kertas karton berwarna putih yang berfungsi sebagai *background*. Hasil gambar yang diperoleh disimpan dalam format .jpg dengan masing-masing resolusi dalam piksel 1920x1080. Data diambil saat kondisi siang hingga sore hari dalam keadaan cuaca terang. Masing-masing data diambil pada jarak 25 cm antara kamera dengan daun padi.

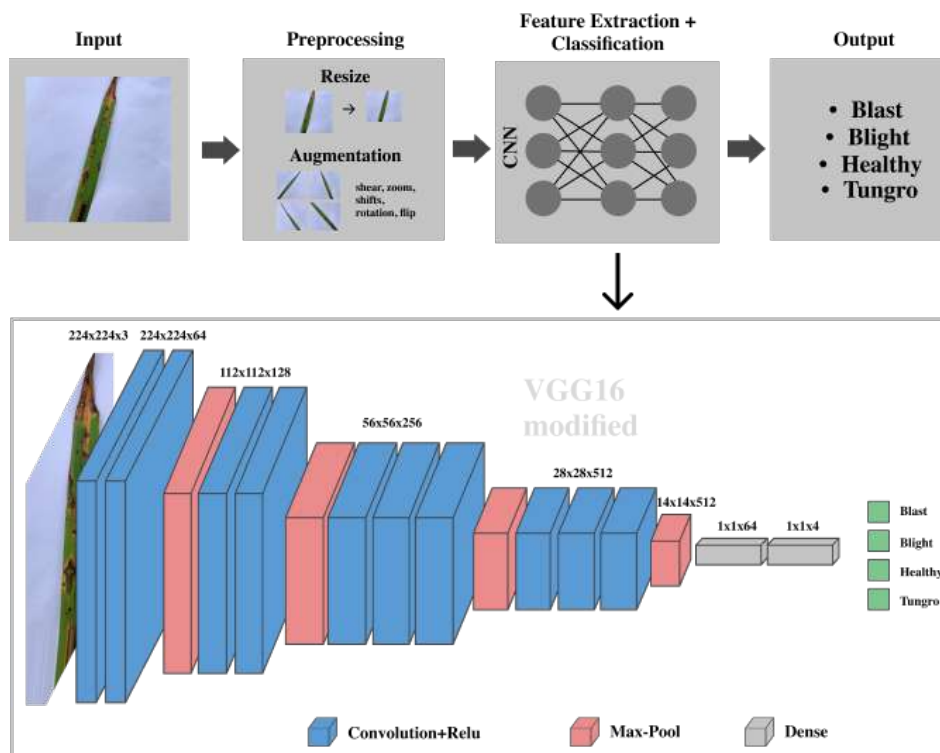
3.5 Perancangan Sistem

Pada penelitian ini, sistem terbagi menjadi dua bagian, yaitu *training* dan *testing*. Pada bagian *training*, data akan melalui proses *preprocessing*, kemudian dilakukan *feature extraction* menggunakan *convolutional layer* sampai didapatkan *feature maps* yang kemudian akan dijadikan input ke dalam *fully connected layer* sehingga menghasilkan klasifikasi. Hasil dari *fully connected layer* ini yang kemudian akan digunakan untuk mengklasifikasikan penyakit padi pada penelitian ini. Model dari hasil proses *training* ini akan disimpan dan digunakan untuk proses *testing*.

Rangkaian tahapan *training* dan *testing* sistem dapat dilihat pada Gambar 17.



Gambar 17. Tahapan *Training* dan *Testing* Sistem



Gambar 18. Ilustrasi Rancangan Sistem

3.5.1 Tahapan *Training*

Data yang digunakan pada penelitian ini berupa citra daun padi. Data citra diperoleh dari proses pengambilan data secara langsung yaitu sebanyak 320 Citra. Data citra penyakit pada tanaman padi terdiri dari 4 kelas yaitu *blast*, *blight*, tungro serta daun padi yang sehat (*healthy*).

- **Input Data**

Untuk mengawali proses *training*, *dataset* yang telah disiapkan dibagi menjadi 4 folder sesuai dengan label yang diberi pada masing-masing data yaitu sebanyak 320 data, jadi masing-masing kelas berisi 80 data. Data *training* yang akan digunakan sebanyak 60% dari jumlah data dan 40% untuk data *testing*. *Data validation* digunakan untuk proses validasi akurasi pada model. Dalam tiap *epoch*, setelah selesai dilakukan *training*, dilanjutkan dengan proses validasi. Contoh data training dapat dilihat pada gambar berikut.



Gambar 19. Contoh data dari 4 kelas

Berikut merupakan jumlah data dari tiap kelas yang digunakan pada penelitian ini.

Tabel 2. Jumlah *dataset* pada setiap kelas

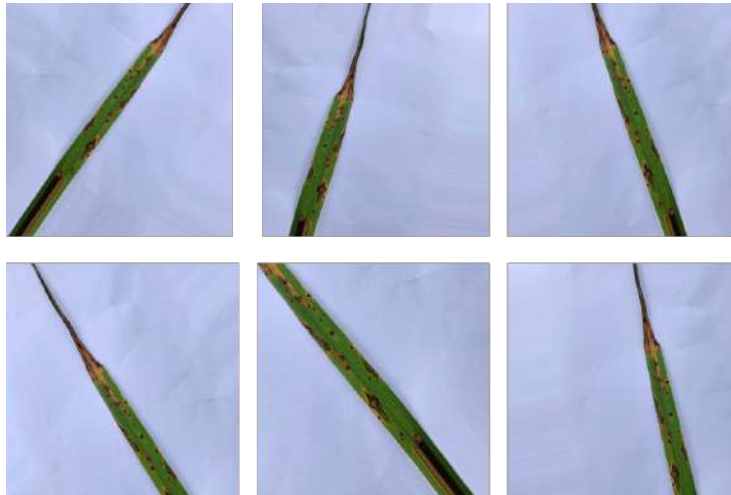
Label	<i>Data</i>	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
<i>Blast</i>	80	48	8	24
<i>Blight</i>	80	48	8	24
<i>Healthy</i>	80	48	8	24
Tungro	80	48	8	24
Total	320	192	32	96

- ***Preprocessing***

Tahap *preprocessing* dilakukan untuk mempersiapkan data yang telah dikumpulkan. Tahap awal yaitu melakukan proses *resize* menjadi 224x224 piksel pada masing-masing data. Hal ini dilakukan agar data yang digunakan menjadi optimal sehingga proses komputasi menjadi lebih cepat tanpa kehilangan akurasi dari sistem.

Pada proses ini, penulis juga melakukan *data augmentation* pada tahap *training*. *Data augmentation* berfungsi untuk menciptakan artifisial data sehingga menghasilkan lebih banyak variasi *data train*. Hasil dari *data augmentation* diterapkan ke dalam *mini batch* pada saat proses *training*. Ada beberapa variasi *augmentation* yang digunakan yaitu *shear*, *zoom*, *shifts*, *rotation* dan *flip*. Jumlah data hasil *augmentation* sangat bergantung pada

banyaknya variasi *augmentation* yang digunakan. Beberapa contoh ilustrasi hasil dari *augmentation* yang dilakukan pada salah satu data dapat dilihat pada Gambar 20.



Gambar 20. Contoh hasil *augmentation* pada salah satu data *train*

Berikut merupakan potongan *source code data augmentation*:

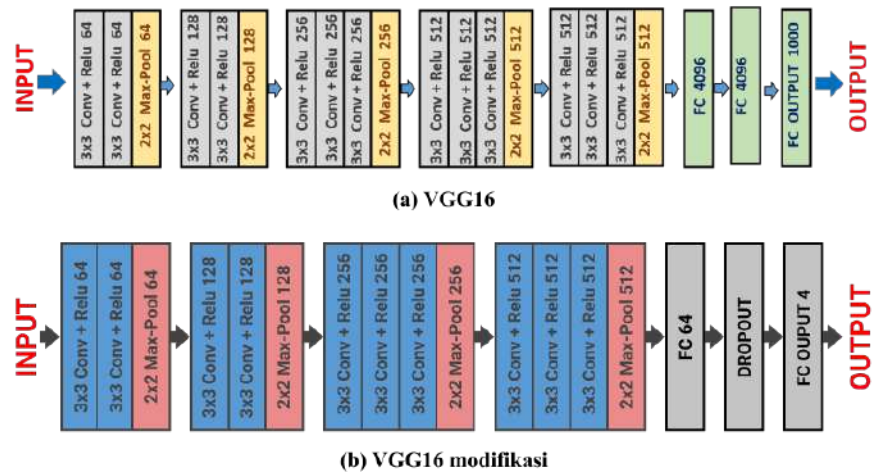
```
train_datagen = ImageDataGenerator(rescale=1. / 255,  
                                   shear_range=20,  
                                   zoom_range=[0.5,1.5],  
                                   width_shift_range=0.2,  
                                   height_shift_range=0.2,  
                                   rotation_range=45,  
                                   horizontal_flip=True)
```

Gambar 21. Potongan *source code data augmentation*

- ***Training Model***

Setelah tahap *preprocessing* telah selesai, maka akan dilakukan proses *training*. Sistem yang dibuat menggunakan algoritma *Convolutional Neural Network (CNN)* dan memakai *VGG16 architecture* sebagai *base model*. Pemilihan *VGG16* sebagai *base model*, karena memiliki arsitektur yang sederhana sehingga

diharapkan proses *training* dapat berjalan lebih cepat. Sebelum melakukan *training*, penulis melakukan modifikasi pada arsitektur *VGG16* yaitu mengurangi beberapa *layer* agar parameter yang *ditraining* lebih sedikit sehingga dapat mengefisiensikan waktu pada saat *training*. Kemudian ditambahkan *layer classification* sesuai dengan label *dataset* yaitu 3 kelas penyakit padi serta data padi yang sehat. Gambar 22 adalah gambaran mengenai arsitektur dari *VGG16*.



Gambar 22. Arsitektur (a)*VGG16* (Khandelwal, 2020) (b)*VGG16* modifikasi

Dalam arsitektur *VGG16* yang dimodifikasi, penulis menerapkan beberapa layer, yaitu *flatten*, *fully connected-layer* serta *dropout*.

Dalam proses *training* ini terjadi tahapan proses yang berdasar pada *Convolutional Neural Network* (CNN) yang ditunjukkan pada Gambar 18 seperti diuraikan di bawah ini:

a. *Feature Extraction*

Tahapan *feature extraction* ini, ada 2 proses yang akan dilakukan terulang beberapa kali sampai menghasilkan *activation map*, yaitu:

- *Convolution*

Convolution (conv) ini merupakan *layer* pertama pada *feature extraction* seperti yang dijelaskan pada bab 2 sebelumnya. *Conv layer* terdiri dari neuron yang membentuk sebuah filter yang memiliki panjang dan tinggi piksel. Filter ini akan digeser ke seluruh bagian gambar, setiap pergeseran dilakukan operasi dot antara input dan nilai dari filter tersebut sehingga menghasilkan *activation map*.

Gambar 22(b) merupakan arsitektur yang digunakan pada penelitian ini. Pada konvolusi pertama digunakan filter 64 menggunakan 2 *layer conv2d* dan *activation relu*. Kemudian pada konvolusi kedua, digunakan filter sebanyak 128 dengan kernel matriks 3x3. Konvolusi ketiga penggunaan filter sebanyak 256 dan konvolusi keempat menggunakan filter sebanyak 512.

- *Max Pooling*

Layer ini terdiri dari filter dengan ukuran dan *stride* tertentu yang akan bergeser ke seluruh *activation map* yang

berasal dari proses sebelumnya yaitu *conv layer*. Pooling *layer* digunakan untuk mempercepat komputasi karena parameter yang harus di-*update* akan semakin sedikit setelah proses *pooling* dilakukan. *Layer Max-Pooling* yang digunakan pada arsitektur ini yaitu berukuran 2x2.

Kedua proses tersebut akan dilakukan berulang hingga membentuk *activation map* yaitu gambar yang sudah terbentuk setelah melewati filter yang mengekstrak fitur yang berbeda dari gambar tersebut.

b. *Classification*

Pada tahap ini hasil dari ekstraksi fitur akan menjadi input pada proses ini, dimana proses *Fully Connected Layer* (FC) yang memiliki susunan yaitu *input layer*, *hidden layer*, *output layer*, *activation function*, dan *loss function*. Input ini nantinya berbentuk vektor sehingga *activation map* yang hasilnya berbentuk *array* harus melewati proses *reshape*. Hasil dari *reshape* kemudian digunakan sebagai input pada *Fully Connected Layer* pada proses klasifikasi.

Pada *Classification Layer* ini, ada beberapa tahapan yang diterapkan yaitu *flatten* yang berfungsi untuk merubah *output* dari proses konvolusi berupa matriks menjadi vektor. Selanjutnya diteruskan pada proses klasifikasi menggunakan jumlah neuron yang telah ditentukan. *Layer dropout* juga digunakan untuk mencegah terjadinya *overfitting* (kondisi

dimana model terlalu sempurna mendeskripsikan data *training*). *Dropout* bekerja untuk menghilangkan beberapa neuron yang tidak terlalu berguna sehingga dapat mempercepat proses *training*. Kelas penyakit padi kemudian diklasifikasikan menjadi 4 kelas berdasarkan nilai dari neuron menggunakan *activation softmax*.

```
# input size
dim = (224, 224)
channel = (3, )
input_shape = dim + channel

#batch size
batch_size = 16

#Epoch
epoch = 30

#learning rate
lr = 0.0001
```

Gambar 23. Paramater yang digunakan untuk *training*

Gambar 23 di atas merupakan penentuan awal parameter yang akan digunakan pada model. Input gambar pada model ini adalah 224x224. Untuk *batch size* nya berukuran 16, *batch size* adalah jumlah sampel yang disebarkan ke dalam arsitektur *neural network*. *Sample per epoch* adalah jumlah sampel yang digunakan dalam tahap pelatihan. Jumlah sampel yang digunakan sebanyak 320 data. Dalam penelitian ini, digunakan 4 kelas yaitu 3 kelas penyakit padi serta kelas yang tidak terkena penyakit. Kemudian *learning rate* menggunakan nilai 0,0001.

Berikut merupakan total *layer* arsitektur *VGG16* yang digunakan pada penelitian ini:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 64)	6422592
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 4)	260
Total params: 14,058,116		
Trainable params: 14,058,116		
Non-trainable params: 0		

Gambar 24. Arsitektur VGG16 yang telah dimodifikasi

3.5.2 Strategi Evaluasi

Algoritma *Convolutional Neural Network* membutuhkan proses *training* dan *testing*. Proses *training* ini bertujuan untuk melatih algoritma CNN dalam mengenali *dataset*-nya dan membentuk sebuah model berdasarkan pelatihan tersebut. Proses *testing* yang dilalui kurang lebih sama dengan proses *training* yaitu *input data*, *preprocessing*, kemudian data diklasifikasikan menggunakan model dari proses *training*. Proses ini bertujuan menguji seberapa akurat sebuah model yang dibentuk dari hasil *training*. Data yang digunakan pada proses *testing* berbeda dengan data pada proses *training* seperti skenario pembagian *dataset* yang sudah diperlihatkan pada Tabel 1.

Berikut merupakan potongan *source code* untuk melakukan *testing* pada salah satu data.

```

im = Image.open('/content/dataset/test/blast/IMG_0599.jpg')
X = preprocess(im,input_size1)
X = reshape([X])
y = model.predict(X)

print( labels[np.argmax(y)], np.max(y) )
fig = plt.imshow(im)
plt.axis('off')
fig.axes.get_xaxis().set_visible(False)
fig.axes.get_yaxis().set_visible(False)

blast 0.9065929

```



Gambar 25. Potongan *source code* untuk *predict image*

Hasil prediksi yang dilakukan model pada data *testing* akan ditampilkan menggunakan *confusion matrix* seperti yang terlihat pada Tabel 3.

Tabel 3. *Confusion matrix data testing*

		Predicted			
		Blast	Blight	Healthy	Tungro
True	Blast	TP			
	Blight		TP		
	Healthy			TP	
	Tungro				TP

Untuk menghitung nilai akurasi sistem, dapat dilakukan dengan Persamaan 10.

3.5.2.1 Evaluasi Konfigurasi Paramater

Pada penelitian ini, penulis melakukan skenario pengujian pada parameter yang akan digunakan untuk menentukan nilai yang optimal.

- Perbandingan performa *Accuracy VGG16* dan *VGG16* modifikasi

Pada pembahasan sebelumnya, dalam penelitian ini dilakukan modifikasi pada arsitektur *VGG16* yaitu mengurangi 1 blok konvolusi dan menambah beberapa layer pada bagian *classification layer*. Untuk itu, penulis ingin menampilkan apakah perubahan yang terjadi menjadi lebih baik atau buruk.

- Perbandingan *Training time VGG16* dan *VGG16* modifikasi

Setelah menampilkan performa pada arsitektur *VGG16* yang dimodifikasi, penulis juga akan menampilkan perubahan yang terjadi pada *training time*.

- Perbandingan nilai *Epoch*

Pada bagian ini, penulis menguji jumlah *epoch* untuk melihat nilai yang paling optimal.

- Perbandingan nilai *Learning rate*

Rentang nilai untuk *learning rate* adalah 0 dan 1. Untuk menentukan nilai *learning rate* yang akan digunakan, penulis melakukan pengujian antara nilai 0,0001 dan 0,00001.

3.5.2.2 Evaluasi *Prototype*

Pada bagian akhir dari evaluasi sistem, penulis melakukan pengujian kepada beberapa *user* (petani). Pengujian ini dilakukan untuk mengevaluasi kinerja sistem yang dibuat pada penelitian ini. *Usability Testing* dilakukan dengan cara memberi *task* serta kuesioner kepada 15 *user*. Pengisian kuesioner dibantu dengan wawancara serta memberi edukasi atau informasi mengenai jenis-jenis penyakit padi jika ada responden/*user* yang belum paham. Pengujian dilakukan di Dusun Bangkala dan Panaikang, Pattalassang, Kabupaten Gowa, Sulawesi Selatan.

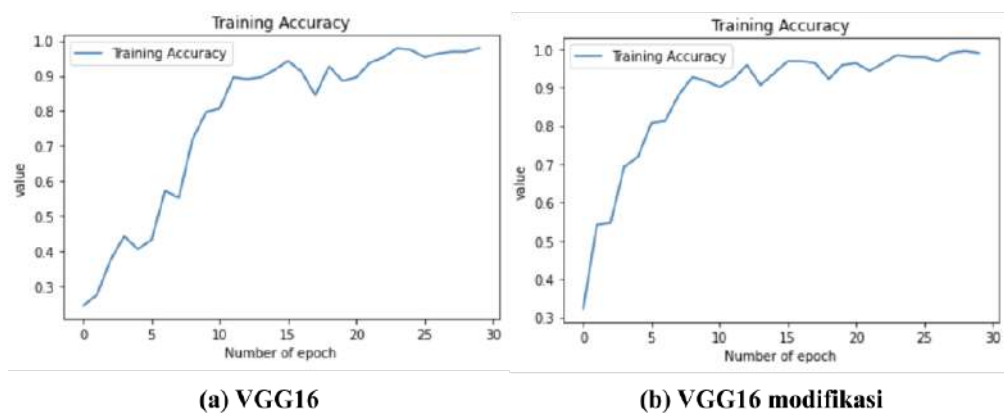
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Penelitian ini dilakukan dengan melakukan klasifikasi empat kelas penyakit padi, yaitu *blast*, *blight* dan tungro serta *healthy* untuk data padi yang tidak terkena penyakit. Sistem ini menggunakan algoritma *Convolutional Neural Network (CNN)* dengan *VGG16* sebagai arsitekturnya. Langkah utama dalam pembentukan model ialah dengan proses *training* data. Proses ini bertujuan agar model dapat mengenali *dataset* yang akan digunakan. Kemudian setelah *training* nantinya model yang telah terbentuk akan diuji menggunakan data *testing*. Adapun jumlah data yang digunakan adalah sebanyak 320 data, dengan perbandingan 60% untuk *training* dan 40% yang digunakan menjadi data *testing* terhadap sistem.

Berikut merupakan perbandingan hasil *accuracy* dari arsitektur *VGG16* yang asli dan *VGG16* yang telah dimodifikasi dengan menggunakan nilai parameter *batch size* 16, *epoch* 30 dan *learning rate* 0,0001.



Gambar 26. Perbandingan grafik *training accuracy* (a) *VGG16* (b) *VGG16* modifikasi

Pada bagian sub bab 3.5.1, Gambar 22(a) merupakan arsitektur *VGG16*, sedangkan Gambar 22(b) adalah arsitektur *VGG16* yang dimodifikasi. Penulis melakukan sedikit perubahan pada arsitektur, yaitu dengan mengurangi 1 blok proses konvolusi sehingga menjadi 4 blok dari yang sebelumnya 5 blok. Perubahan ini dilakukan dengan tujuan untuk mengurangi *training time* seperti yang terlihat pada Tabel 4, namun tidak mengurangi performa sesuai dengan Gambar 26 yang ditampilkan di atas.

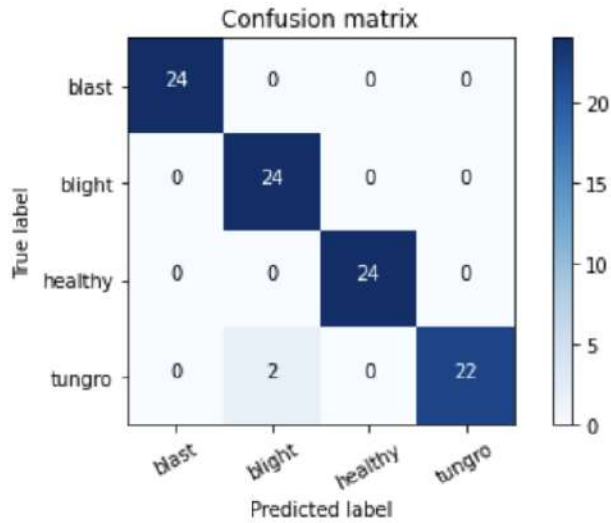
Berikut adalah hasil perbandingan *training time VGG16* dan *VGG16* yang telah dimodifikasi, hasil dibawah merupakan rata-rata yang diperoleh setelah 5 kali percobaan.

Tabel 4. Perbandingan *Training Time VGG16*

<i>Architecture</i>	<i>Training Time</i>
VGG16	7 menit 17 detik
VGG16 modifikasi	6 menit 15 detik

Pada penelitian ini, parameter untuk mengukur tingkat keberhasilan model adalah nilai akurasi. Nilai akurasi model dapat ditentukan dengan melakukan pengujian menggunakan data *testing*. *Confusion matrix* akan digunakan untuk membantu dalam melakukan perhitungan akurasi sistem.

Adapun hasil kinerja sistem terhadap 96 data *testing* dapat dilihat pada gambar berikut.



Gambar 27. Confusion matrix pada data testing menggunakan VGG16 yang telah dimodifikasi

Gambar 27 merupakan hasil *confusion matrix* salah satu dari lima kali percobaan yang dilakukan. Pada gambar tersebut, dapat dilihat bahwa *accuracy* yang didapatkan dari *confusion matrix* sistem tersebut adalah sebesar 98% dengan menggunakan arsitektur VGG16 yang telah dimodifikasi.

Tabel 5 berikut merupakan hasil *testing* yang dilakukan sebanyak 5 kali percobaan.

Tabel 5. Testing VGG16 modifikasi

Percobaan	Akurasi
1	98%
2	98%
3	97%
4	97%
5	98%
Rata-rata akurasi	97.6%

Tabel 6 di bawah menunjukkan hasil dari percobaan yang dilakukan pada proses *training* menggunakan *image enhancement* yaitu *gaussian blur* dan *sharpen*. Jika dilihat pada kolom akurasi, percobaan ini tidak terlalu memberi dampak yang signifikan terhadap hasil dari proses *training*, artinya data yang digunakan pada penelitian ini tidak membutuhkan *pre-processing gaussian blur* maupun *sharpen*.

Adapun hasil 5 kali percobaan pada *training* yang dilakukan adalah sebagai berikut:

Tabel 6. Training VGG16 modifikasi dengan Image enhacement

Percobaan	Akurasi <i>training</i>	Waktu <i>training</i>
VGG16	100%	6 menit 15 detik
VGG16+a	97,81%	7 menit 3 detik
VGG16+b	98,96%	6 menit 51 detik
VGG16+c	98,44%	7 menit 12 detik

Keterangan: a= *gaussian blur*, b= *sharpen*, c= a+b

Berikut merupakan perbandingan penelitian terkait dengan penelitian yang dilakukan saat ini.

Tabel 7. Perbandingan Penelitian Terkait dengan Penelitian yang dilakukan

Judul (Penulis, Tahun)	Metode yang digunakan	Hasil
<i>Identification of Rice Diseases using Deep Convolutional Neural</i>	<i>Convolutional Neural Network</i>	95,48% Akurasi.

<i>Networks</i> (Yang Lu dkk, 2017).		
<i>Rice Blast Disease Recognition Using a Deep Convolutional Neural Network</i> (Wanjie Liang dkk, 2018).	<i>Convolutional Neural Network + Support Vector Machine</i>	95, 82% Akurasi.
Klasifikasi Penyakit Tanaman Padi Menggunakan Metode <i>Deep Learning</i> (Tedi Setiady, 2021)	<i>Convolutional Neural Network</i>	97,6% Akurasi.

4.2 Pembahasan

Dalam pembentukan model yang terbaik diperlukan penentuan parameter optimal dalam setiap proses metode CNN. Parameter yang dimaksud adalah pengaruh jumlah *epoch*, pengaruh jumlah *data train* dan *learning rate*. Tujuan dilakukannya percobaan ini ialah ingin melihat nilai parameter yang paling cocok untuk kemudian digunakan pada proses *training*. Berikut merupakan parameter-parameter yang akan digunakan dalam proses *training*:

- Pengaruh Jumlah *Epoch*

Satu *epoch* adalah ketika seluruh *dataset* sudah melalui proses *training* pada *neural network* dalam satu putaran. Untuk mempermudah dan mempercepat proses *training* biasanya, *dataset* dibagi per *batch* (*batch size*). Penentuan nilai dari *batch size* biasanya tergantung peneliti dengan melihat banyak sampel. Berikut merupakan hasil perbandingan *epoch* dari hasil *training* dilakukan.

Tabel 8. Perbandingan nilai *Epoch* saat *training*

<i>Epoch</i>	<i>Accuracy</i>
20	92,53%
30	100%

Berdasarkan Tabel 8, penulis melakukan percobaan menggunakan jumlah *epoch* 20 dan 30, jumlah yang optimal berada pada nilai 30. Penulis juga sudah melakukan percobaan dengan menggunakan di atas jumlah 30 namun performa yang didapatkan hampir serupa sehingga penulis memilih nilai *epoch* tersebut untuk alasan efisiensi *training time*.

- Pengaruh *Learning Rate*

Penelitian ini melakukan uji coba dengan menggunakan nilai *learning rate* yang berbeda. Dalam klasifikasi gambar pada umumnya banyak menggunakan nilai *learning rate* sebesar 0,1 sampai 0,00001. Penulis menggunakan dua nilai yaitu 0,0001 dan 0,00001. Penentuan nilai dari *learning rate* ini sangat berpengaruh pada performa akurasi. Hasil perbandingan *learning rate* dapat dilihat pada Tabel 9.

Tabel 9. Perbandingan nilai *Learning rate* saat *training*

<i>Learning rate</i>	<i>Accuracy</i>
0,0001	100%
0,00001	96,35%

Berikut merupakan parameter-parameter yang digunakan dalam proses *training*:

1. *Width dan Height*

Parameter ini merupakan ukuran data citra yang diinput ke dalam sistem. Nilai *width x height* yang digunakan adalah 224x224 piksel. Ukuran tersebut merupakan batasan maksimal yang diterima oleh arsitektur *VGG16*.

2. *Channel*

Parameter *channel* yang bernilai 3 adalah menunjukkan bahwa citra yang akan diproses adalah citra 3 *channel* RGB (*Red, Green, Blue*).

3. *Batch size*

Sebelum menentukan nilai parameter *batch size* ini, penulis terlebih dahulu percobaan dengan menggunakan nilai 8, 16, 32. Dari percobaan yang dilakukan, nilai 16 merupakan nilai yang optimal. Nilai 16 ini berarti bahwa dalam satu *steps per epoch* terdapat 16 data yang digunakan dalam mengupdate *weight* yang digunakan sebagai model dari hasil proses *training* data.

4. *Learning rate*

Sistem yang dibuat menggunakan nilai *learning rate* 0,0001. Jika *learning rate* yang digunakan lebih besar maka akan berdampak pada *weights* yang dijadikan sebagai model sehingga bisa mengakibatkan nilai *loss* dapat bertambah. Nilai *learning rate* yang terlalu besar dapat menyebabkan performa model kurang optimal, namun nilai yang terlalu kecil juga dapat menyebabkan *stuck* pada proses *training*.

5. Epoch


Pada proses *training* dari sistem *epoch* yang digunakan yaitu 30 artinya seluruh data melewati seluruh rangkaian *neural network* sebanyak 30 kali.


Dengan menggunakan parameter-parameter di atas, sistem yang telah dibuat dilakukan *training* dengan 30 *epoch* dengan jumlah data *training* sebanyak 192 data, kemudian dilakukan proses *testing* yang memberikan hasil akurasi rata-rata sebesar 97,6%. Sistem yang dibuat hanya memiliki 4 jenis kelas antara lain: *blast*, *blight*, tungro dan *healthy*.

Sistem yang dibuat memiliki nilai *false positive* sebanyak 2 data, menandakan bahwa ada 2 data yang salah prediksi dari data *testing*, yang seharusnya kelas tungro tetapi diprediksi oleh sistem menjadi kelas *blight*. Hal ini disebabkan karena kedua penyakit tersebut memiliki kemiripan ciri-ciri yaitu bercak atau bentuk dari kerusakan pada daun bagian samping hingga ujung daun.

Tabel 10 merupakan hasil klasifikasi yang salah pada data *testing*.




Tabel 10. Hasil kesalahan klasifikasi


No.	Data	Aktual	Prediksi
1	<div>blight 0.54137594</div> 	Tungro	Blight

2	blight 0.6636834 	Tungro	Blight
---	---	--------	--------

Adapun untuk hasil klasifikasi penyakit padi yang benar dapat dilihat pada Tabel 11.

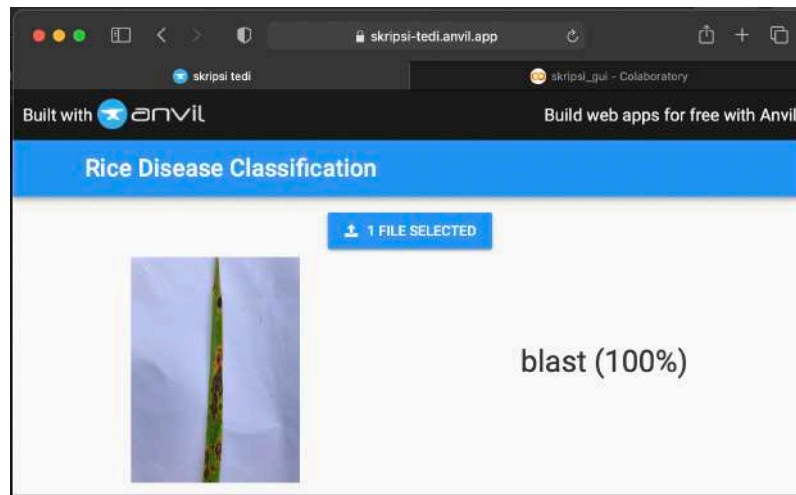
Tabel 11. Hasil klasifikasi benar

No	Data	Aktual	Prediksi
1	blast 0.9999987 	Blast	Blast
2	blight 0.98479176 	Blight	Blight
3	tungro 0.9847245 	Tungro	Tungro

4		Healthy	Healthy
---	---	---------	---------

Setelah dilakukan evaluasi sistem kepada sejumlah responden/*user*, penelitian yang dilakukan ini mendapat respon yang baik dari responden. Mayoritas responden menjawab bahwa mereka mengetahui beberapa penyakit padi yang dijadikan sebagai bahan penelitian yaitu *blast*, *blight* (hawar daun bakteri/kresek) dan tungro. 12 dari 15 responden menjawab melakukan pengamatan pada bagian daun dan batang dari tanaman padi untuk melihat jika terindikasi terserang penyakit. Namun terkadang mereka membutuhkan bantuan dari sesama teman petani untuk mengidentifikasi penyakit maupun cara penanganannya. Sebanyak 40% responden mengalami kesulitan dalam membedakan penyakit *blight* dan tungro ketika diperlihatkan contoh gambar penyakit tersebut. Hal yang sama juga terjadi pada sistem, dapat dilihat pada Gambar 27 yaitu sistem mengalami kesalahan klasifikasi pada kelas tungro. Hal ini disebabkan karena kedua penyakit tersebut memiliki kemiripan dari bentuk bercak pada daun padi. Selain itu pada saat uji *prototype*, mayoritas *user* mengalami kesulitan dalam mengerjakan *task*. Hal ini dikarenakan mayoritas *user* jarang menggunakan *smartphone* dalam aktivitas sehari-hari. Namun setelah diperlihatkan cara penggunaan aplikasi, 11 *user* berpendapat mereka nantinya akan terbantu dengan penelitian ini.

Berikut merupakan tampilan *Interface* dari sistem yang dibuat penelitian yang dilakukan. *Interface* ini dibuat menggunakan *Anvil*, yang mana merupakan *web app builder*. *Anvil* dapat langsung dihubungkan ke *Google Colab*. *Interface* tersebut dapat diakses melalui link ini. <https://skripsi-tedi.anvil.app>



Gambar 28. Tampilan *Interface* Sistem Klasifikasi Penyakit Tanaman Padi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis terhadap pengujian pada sistem klasifikasi penyakit padi, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Penelitian ini menggunakan arsitektur *VGG16* yang telah dimodifikasi. Sistem yang dibuat dengan menggunakan *input size* 224x224, *channel* 3, *epoch* 30, *batch size* 16 dan *learning rate* 0,0001. Sistem ini mampu mengklasifikasikan penyakit padi menjadi 4 kelas yaitu *blast*, *blight*, *tungro* dan juga *healthy*.
2. Penelitian ini dilakukan menggunakan data *testing* sebanyak 96 data. Hasil *testing* pada sistem dengan menggunakan ukuran input yang sama dengan data *training* mendapatkan nilai rata-rata akurasi sebesar 97,6% dengan 5 kali percobaan menggunakan arsitektur *VGG16* yang telah dimodifikasi.

5.2 Saran

Sehubungan dengan selesainya penulisan skripsi ini, penulis bermaksud untuk menyampaikan beberapa saran terhadap pengembangan sistem, antara lain:

1. Diharapkan pada penelitian selanjutnya dapat menambahkan lebih banyak jumlah data dan kelas penyakit padi, agar sistem lebih akurat dan presisi.
2. Sistem ini dapat dimodifikasi untuk mengatasi permasalahan lain.

DAFTAR PUSTAKA

- Andono, P.N., & Sutojo, T. (2017). Pengolahan citra digital. Penerbit Andi.
- Badan Pusat Statistik. (2020). Luas Panen dan Produksi Padi di Indonesia 2019. Diakses pada 15 Desember 2020, dari <https://www.bps.go.id/publication/2020/12/01/21930121d1e4d09459f7e195/luas-panen-dan-produksi-padi-di-indonesia-2019.html>.
- Bogorkab.go.id, (2014). Penyakit Kresek/Hawar Daun (Bacterial Leaf Blight). Diakses pada 29 September 2020, dari <https://bogorkab.go.id/post/detail/penyakit-kresek-hawar-daun-bacterial-leaf-blight>.
- BPTP Kalimantan Tengah. (2015). Hama dan Penyakit Utama Tanaman Padi di Kalimantan Tengah. Diakses pada 29 September 2020, dari <http://kalteng.litbang.pertanian.go.id/ind/index.php/publikasi-mainmenu-47-47/teknologi/525-hama-dan-penyakit-utama-tanaman-padi-di-kalimantan-tengah20>.
- CS231n. Convolutional Neural Network for Visual Recognition. Diakses pada 23 Desember 2020, dari <http://cs231n.github.io/convolutional-networks/>.
- Gomez, Raul. (2018). *Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names*. Diakses pada 12 Februari 2021, dari https://gombru.github.io/2018/05/23/cross_entropy_loss/.
- Hubel, D. & Wiesel, T. (1968). *Receptive Fields and Functional architecture of monkey striate cortex. Journal of Physiology (London)*, 195, 215-243.
- Khandelwal, Vaibhav. (2020). *The Architecture and Implementation of VGG-16*. Diakses pada 2 Januari 2021, dari <https://medium.com/towards-artificial-intelligence/the-architecture-and-implementation-of-vgg-16-b050e5a5920b>.
- Liang, W. J., Zhang, H., Zhang, G. F., & Cao, H. X. (2019). Rice blast disease recognition using a deep convolutional neural network. *Scientific reports*, 9(1), 1-10.
- Loukadakis, Manolis, Cano, José & O'Boyle, Michael. (2018). *Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures*. Diakses pada 10 Februari 2021, dari https://www.researchgate.net/figure/VGG-16-neural-network-architecture_fig1_327070011.

- Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267, 378-384.
- MathWorks. (2018). Diakses pada 26 Desember 2020, dari <https://www.mathworks.com/help/images/ref/graycoprops.html?searchHighlight>.
- Mohajon, J. (2020). *Confusion Matrix for Your Multi-Class Classification Machine Learning Model*. Diakses pada 12 Februari 2021, dari <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>.
- Patel, S. & Pingel, J. (2017). *Introduction to Deep Learning*. Diakses pada 23 Desember 2020, dari <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>.
- Pamungkas, Adi. (2017). Pengolahan Citra Digital. Diakses pada 23 Desember 2020, dari <https://pemrogramanmatlab.com/2017/07/26/pengolahan-citra-digital/>.
- Prajapati, H. B., Shah, J. P., & Dabhi, V. K. (2016). A survey on detection and classification of rice plant diseases. In *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)* (pp. 1-8). IEEE.
- Sena, Samuel. (2018). Pengenalan Deep Learning: Neural Network. Diakses pada 26 Desember 2020, dari <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>.
- Srivastava, N., Hinton, G., & Krizhevsky, A. (2014). *Dropout: A Simple Way to Prevent Neural Network*. Journal Conference Learning Research, 1929- 1958.
- Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto. (2009) “Teori Pengolahan Citra Digital”, Andi Yogyakarta dan UDINUS Semarang.
- Syawaluddin. (2016). Pengolahan Citra Digital. Jakarta.
- Wayan Dadang, 2018. *Pengaplikasian Deep Learning*. Jakarta
- Wulandari, I. A., Aristoteles, A., & Suharjo, R. (2015). Sistem Identifikasi Penyakit Tanaman Padi dengan Menggunakan Metode *Forward Chaining*. Jurnal Komputasi, 4.
- Zahrah, S., Saptono, R., & Suryani, E. (2016). Identifikasi Gejala Penyakit Padi Menggunakan Operasi Morfologi Citra. *SNIK, UNNES*, 100-106.

LAMPIRAN

1. *Source code*

Source code untuk penelitian ini dapat dilihat pada pranala berikut.

https://github.com/tedsetpb/skripsi/blob/main/skripsi_penyakit_padi.ipynb

2. *Dataset*

Dataset yang digunakan pada penelitian ini dapat dilihat pada pranala berikut. <https://www.kaggle.com/tedisetiady/leaf-rice-disease-indonesia>

3. *Task (User Testing)* dan Kuesioner

Untuk melihat lampiran *task* dan kuesioner silakan mengunjungi pranala berikut. https://github.com/tedsetpb/skripsi/blob/main/task_kuesioner.pdf



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SEMINAR PROPOSAL


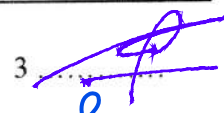

Nama/Stambuk : 1.Tedi Setiady D42116 508

Judul Skripsi/T.A : “ Deteksi Penyakit Hawar Daun Bakteri pada Tanaman Padi dengan Metode Deep Learning“

Hari/Tanggal : Rabu, 7 Oktober, 2020

Jam : 13.00 Wita – Selesai

Tempat : Ruang Lab.AIMP Departemen Teknik Informatika


No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Indrabayu, ST., M.T., M.Bus .Sys	1.....
	Pembimbing II	2. Anugrayani Bustamin, ST., M.T	2 
II.	Anggota Penguji	3. Dr.Eng.Ady Wahyudi P.ST.,M.T	3 
		4. Iqra Aswad, ST., M.T	4 

PANITIA UJIAN

Ketua,

Dr.Indrabayu, ST.M.T.M.Bus.Sys

Sekretaris,


Anugrayani Bustamin, ST., M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SEMINAR PROPOSAL

Pada hari ini **Rabu**, tanggal **7 Oktober 2019** Pukul **13.00 WITA-Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika**, telah dilaksanakan Ujian Seminar Proposal bagi Saudara :

Nama : Tedi Setiady

No. Stambuk : D421 16 508

Fakultas/Departemen : Teknik/Teknik Informatika

Judul Skripsi : “ **Deteksi Penyakit Hawar Daun Bakteri pada Tanaman Padi dengan Metode Deep Learning**”

Yang dihadiri oleh panitia Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Indrabayu,ST.,M.T.,M.Bus .Sys	Pemb I/Ketua	1.....
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2.....
3.	Dr.Eng.Ady Wahyudi P.ST.,M.T	Anggota	3.....
4.	Iqra Aswad,ST.,M.T	Anggota	4.....

Hasil keputusan panitia penilai Ujian Seminar Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka ... **84,71** ... dan huruf **A-**

Gowa, 7 Oktober, 2020

Ketua/Sekretaris Panitia Ujian,

Dr. Indrabayu,ST.,M.T.,M.Bus .Sys



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SEMINAR PROPOSAL

Nomor :/ UN4.7.7.TI/PK.03.06/2020

Pada hari ini **Rabu**, tanggal **7 Oktober 2020** Pukul **13.00 WITA-Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika**, telah dilaksanakan Ujian Seminar Proposal bagi Saudara :

Nama : Tedi Setiady

No. Stambuk : D421 16 508

Fakultas/Departemen : Teknik/Teknik Informatika

Judul Skripsi : “ **Deteksi Penyakit Hawar Daun Bakteri pada Tanaman Padi dengan Metode Deep Learning**”

Setelah pembawa ujian seminar proposal menguraikan tugas akhirnya dan menjawab pertanyaan dengan dinyatakan lulus/tidak lulus, Baik/Cukup/Sedang.

Maka berdasarkan hasil penilaian dinyatakan lulus / tidak lulus

Dengan nilai :

A — **A-** — B+ — B — B- — C+ — C — D — E
84,71

Mengetahui:

Ketua Departemen Tek.Informatika,

Dr. Amil Ahmad Ilham, ST., M.I.T
Nip. 19731010 199802 1 001

Dosen Penguji,

Dr. Indrabayu, ST., M.T., M.Bus.Sys
Nip. 19750716 200212 1 004

Diketahui oleh,
Dekan,
Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN

No. 14132/UN4.7.1/TD.06/2020

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2003 Pasal 36 butir 3 point a, b (SK. Rektor Unhas Nomor : 1067 /J04/PP.08/2008), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR PROPOSAL Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Indrabayu, ST., M.T., M.Bus,Sys
Pembimbing II / Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Dr.Eng. Ady Wahyudi P., ST., M.T
4. Iqra Aswad, ST., M.T

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Tedi Setiady D421 16 508

Departemen : Teknik Informatika

Judul Thesis/Skripsi : **" Deteksi Penyakit Hawar Daun Bakteri Pada Tanaman Padi Dengan Metode Deep Learning "**

2. Waktu seminar ditetapkan oleh Panitia Seminar Proposal Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 2 Oktober 2020
a.n. Dekan.
Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 19690308 199512 1 001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : 1. Tedy Setiady P.Ba'ka D4211 6 508

Judul Skripsi/T.A : "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"

Hari/Tanggal : Kamis, 11 Februari 2021

Jam : 10- 00 Wita – Selesai

Tempat : Ruang Lab.AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
<hr/>			
L.	Pembimbing I	1. Dr. Indrabayu,ST.,M.T.,M.Bus.Sys	1.....
	Pembimbing II	2. Anugrayani Bustamin,ST.,M.T	2
<hr/>			
II.	Anggota Penguji	3. Dr.Eng. Ady Wahyudi Paundu,ST.,M.T	3
		4. Iqra Aswad,ST.,M.T	4

PANITIA UJIAN

Ketua,

Sekretaris,

Dr. Indrabayu,ST.,M.T.,M.Bus.Sys

Anugrayani Bustamin,ST.,M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini **Kamis**, tanggal **11 Februari 2021** Pukul **10.00 WITA** - **Selesai** bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Tedy Setiady P.Ba'ka
No. Stambuk : D4211 6 508
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Indrabayu,ST.,M.T.,M.Bus.Sys	Pemb I/Ketua	1.....
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2.....
3.	Dr.Eng. Ady Wahyudi Paundu,ST.,M.T	Anggota	3.....
4.	Iqra Aswad,ST.,M.T	Anggota	4.....

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus / Tidak lulus** dengan nilai angka **86**
dan huruf **A**

Makassar, 11 Februari 2021
Ketua/Sekretaris Panitia Ujian,

Dr. Indrabayu,ST.,M.T.,M.Bus.Sys



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa

Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI SEMINAR HASIL

Nomor : 054 / UN4.7.7.TI/PK.03.06/2021

Pada hari ini Kamis, tanggal 11 Februari 2021 Pukul 10.00 WITA - Selesai bertempat di Ruang Lab. AIMP Departemen Teknik Informatika, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Tedy Setiady P.Ba'ka
No. Stambuk : D4211 6 508
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"

Setelah pembawa seminar hasil menguraikan tugas akhirnya dan menjawab pertanyaan dari Tim Penguji dinyatakan Lulus / Tidak Lulus dengan nilai :

A- B+ B B- C+ C D E

Megetahui:

Ketua Departemen Tek.Informatika,

Dosen Penguji,

Dr. Amil Ahmad Ilham, ST., M.I.T
Nip. 19731010 199802 1 001

Dr. Indrabayu, ST., M.T., M.Bus.Sys
Nip. 197507162002121 004

Diketahui oleh,
a.n Dekan,
Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan

Telp. (0411) 586015, 586262 Fax (0411) 586015

<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN

No. 1880/UN4.7.1/TD.06/2021

- Dari : Dekan Fakultas Teknik Universitas Hasanuddin
- Kepada : Mereka yang tercantum namanya dibawah ini
- Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 butir point a, b (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

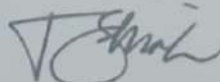
Pembimbing I/ Ketua : 1. Dr. Indrabayu, ST., M.T., M.Bus,Sys
Pembimbing II/ Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Dr.Eng. Ady Wahyudi Paundu, ST., M.T
4. Iqra Aswad, ST., M.T

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Tedy Setiady P. Ba'ka D421 16 508
Program Studi : Teknik Informatika
Judul thesis/Skripsi : Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa
Pada tanggal 8 Pebruari 2021
a.n. Dekan,
Wakil Dekan Bidang Akademik



Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 196903081995121001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN**

KAMPUS TAMALANREA
JALAN PERINTIS KEMERDEKAAN KM.10 MAKASSAR 90245
TELEPON : 0411-586200 (6 SALURAN), 584002, FAX. 585188

SURAT PERSETUJUAN


Nomor : 14607/UN4.1.1.2.1.1/PK.02.03/2021

Berdasarkan Peraturan Rektor Universitas Hasanuddin tentang Penyelenggaraan Program Sarjana Nomor : 2781/UN4.1/KEP/2018 tanggal 16 Juli 2018, dengan ini menerangkan bahwa :

NIK : 7471080711970001 ✓
Nama : TEDI SETIADY PAKIDING BA'KA' ✓
Tempat/Tanggal Lahir : KENDARI, 7 NOVEMBER 1997 ✓
NIM : D42116508 ✓
Fakultas : TEKNIK ✓
Program Studi : TEKNIK INFORMATIKA ✓

Telah memenuhi syarat untuk Ujian Skripsi Strata I (S1). Demikian Surat Persetujuan ini dibuat untuk digunakan dalam proses pelaksanaan ujian skripsi, dengan ketentuan dapat mengikuti wisuda **PERIODE IV JUNI 2021 TAHAP II**, jika **persyaratan kelulusan/wisuda telah dipenuhi**. Terima Kasih.

Makassar, 4 JUNI 2021

Kepala Biro Administrasi Akademik
dan Kepala Sub Bagian Pendidikan dan Evaluasi
Universitas Hasanuddin,

MURSALIM, S.Sos.
NIP. 19730216 199601 1001

Keterangan :

Nomor User : D42116508
Nomor password/pin : 204217230
Alamat Website : <http://unhas.ac.id/akad/wisuda/>
Layanan E-Mail : alimkomath@gmail.com
Catatan

1. Bagi Mahasiswa yang telah melaksanakan ujian Sarjana dan dinyatakan lulus, segera menyerahkan lembar pengesahan Skripsi dan Berita Acara Ujian Sarjana ke Sub Bagian Akademik Fakultas, untuk memperoleh nomor Alumni dan didaftar sebagai Wisudawan pada periode berjalan.
2. Jika terjadi perubahan Judul Skripsi agar melaporkan ke Kasubag. Pendidikan Fakultas sebelum didaftar sebagai Wisudawan pada Periode berjalan
3. Pada saat ON-LINE Mahasiswa diharapkan mengisi identitas diri sesuai surat izin ujian ini
4. Surat izin ini hanya berlaku untuk Wisuda periode berjalan (WISUDA PERIODE IV JUNI 2021 TAHAP II)



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : Tedi Setiady pakiding Ba'ka D421 16 508

Judul Skripsi/T.A : "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"

Hari/Tanggal : Rabu, 2 Juni 2021

Jam : 13.00 Wita – Selesai

Tempat : Lab. AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Indrabayu, ST., M.T., M.Bus.Sys	1
	Pembimbing II	2. Anugrayani Bustamin, ST., M.T	2
II.	Anggota Penguji	3. Dr. Eng.Ady Wahyudi Paundu, ST., M.T	3
		4. Iqra Aswad, S.T., M.T	4

PANITIA UJIAN

Ketua,

Sekretaris,

Dr. Indrabayu, ST., M.T., M.Bus.Sys

Anugrayani Bustamin, ST., M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini **Rabu** tanggal **2 Juni 2021** Pukul **13.00 WITA** - Selesai bertempat di **Lab.AIMP Departemen Teknik Informatika Gowa** , telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Tedi Setiady pakiding Ba'ka
No. Stambuk : D421 16 508
Program Studi : Teknik Informatika
Judul Skripsi : **"Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"**

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Indrabayu,ST.,M.T.,M.Bus.Sys	Pemb I/Ketua	1.....
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2.....
3.	Dr. Eng.Ady Wahyudi Paundu,ST.,M.T	Anggota	3.....
4.	Iqra Aswad,S.T.,M.T	Anggota	4.....

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka **85,5** dan huruf **A**

Gowa, 2 Juni 2021

Ketua/Sekretaris Panitia Ujian,

Dr. Indrabayu,ST.,M.T.,M.Bus.Sys



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SKRIPSI

Nomor : / UN4.7.7.TI/PK.03.06/2021

Pada hari ini **Rabu**, tanggal 2 **Juni 2021** Pukul **13.00 WITA** - Selesai bertempat di **Lab.AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Tedi Setiady pakiding Ba'ka
No. Stambuk : D421 16 508
Program Studi : Teknik Informatika
Judul Skripsi : "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning"

Setelah pembawa ujian Skripsi menguraikan tugas akhirnya dan menjawab pertanyaan dari Tim Penguji dinyatakan Lulus / Tidak Lulus dengan nilai :

A- B+ B B- C+ C D E
05,5

Mengetahui:

Ketua Departemen Tek.Informatika,

Dosen Penguji,

Dr.Amil Ahmad Ilham,ST.,M.I.T
Nip. 19731010 199802 1 001

Dr. Indrabayu,ST.,M.T.,M.Bus.Sys
Nip. 19610813 198811 2 001

Diketahui oleh,
a.n. Dekan,
Wakil Dekan Bidang Akademik,Riset dan Inovasi

Prof. Baharuddin Hamzah,ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6 Bontomarannu (92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 8593/UN4.7.1/TD.06/2021

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 butir point a, b (SK. Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA UJIAN SARJANA Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Indrabayu, ST., M.T., M.Bus, Sys
Pembimbing II / Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Dr. Eng. Ady Wahyudi Paundu, ST., M.T
4. Iqra Aswad, ST., M.T

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Tedi Setiady Pakiding Ba'ka D421 16 508
Program Studi : Teknik Informatika
Judul Thesis/Skripsi : Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Deep Learning

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 31 Mei 2021
a.n. Dekan

Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP.1969030819951210011

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH



LEMBAR PERBAIKAN SKRIPSI

“KLASIFIKASI PENYAKIT TANAMAN PADI MENGUNAKAN METODE *DEEP LEARNING*”





OLEH:

TEDI SETIADY P. BA'KA
D421 16 508


Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 2 Juni 2021.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari
penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Indrabayu, S.T., M.T., M.Bus.Sys.	
Sekretaris	Anugrahyani Bustamin, S.T., M.T.	
Anggota	Dr. Eng. Ady Wahyudi, S.T., M.T.	
	Iqra Aswad, S.T., M.T.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Indrabayu, S.T., M.T., M.Bus.Sys.	
II	Anugrahyani Bustamin, S.T., M.T.	