

PROPOSAL TUGAS AKHIR

PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA *RICE DISEASES DATASET* UNTUK APLIKASI PADI CARE

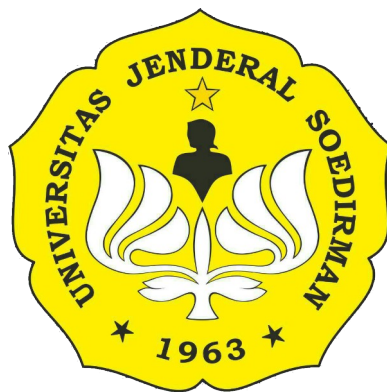
Disusun untuk memenuhi persyaratan memperoleh gelar sarjana
di Jurusan Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Barra Malik Wibowo
H1A020032

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2023**



PROPOSAL TUGAS AKHIR

PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA *RICE DISEASES DATASET* UNTUK APLIKASI PADI CARE

Disusun untuk memenuhi persyaratan memperoleh gelar sarjana
di Jurusan Teknik Elektro Universitas Jenderal Soedirman



Disusun oleh:

Barra Malik Wibowo
H1A020032

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO
PURBALINGGA
2023**

HALAMAN PENGESAHAN

Proposal Tugas Akhir dengan Judul:

PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA RICE *DISEASES DATASET* UNTUK APLIKASI PADI CARE

Disusun oleh:

Barra Malik Wibowo
H1A020032

Diajukan untuk memenuhi salah satu persyaratan
memperoleh gelar Sarjana Teknik pada
Jurusan/Program Studi Teknik Elektro
Fakultas Teknik
Universitas Jenderal Soedirman

Diterima dan disetujui

Pada Tanggal : _____

Pembimbing I

Pembimbing II

Agung Mubyarto S.T., M.T.
(NIP 1974100620002121001)

Muhammad Syaiful Aliim S.T., M.T.
(NIP 19780211912001121001)

Mengetahui:
Dekan Fakultas Teknik

Prof. Dr. Eng. Ir. Agus Maryoto, S.T., M.T.
NIP 197109202006041001

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Proposal Tugas Akhir dengan judul ***“PERANCANGAN MODEL DEEP LEARNING DENGAN ARSITEKTUR CNN PADA RICE DISEASES DATASET UNTUK APLIKASI PADI CARE”*** ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Purbalingga, 20 Desember 2023

Barra Malik Wibowo

NIM H1A020032

HALAMAN MOTTO DAN PERSEMBAHAN

MOTTO

“Kesempatan hanya diperuntukkan untuk orang-orang yang siap.”

PERSEMBAHAN

Laporan tugas akhir ini disusun dan dipersembahkan untuk:

1. Tuhan yang Maha Esa,
2. Nabi Muhammad SAW,
3. Ayah, Bunda, saudara-saudari, serta teman-teman,
4. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis,
5. Siapapun yang mungkin mendapatkan manfaat dari Tugas Akhir ini.

RINGKASAN

PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA *RICE DISEASES DATASET* UNTUK APLIKASI PADI CARE

Barra Malik Wibowo

Penyakit padi dapat menyebabkan kerugian berupa gagal panen, yang berimbas pada penurunan produksi tanaman padi. Oleh karena itu, deteksi dini penyakit pada tanaman padi sangat penting untuk mencegah kerugian yang lebih besar. Padi Care merupakan aplikasi yang dapat membantu dalam mendeteksi penyakit padi. Padi Care ini memberikan 1 fitur utama yaitu mendeteksi penyakit padi dan 2 fitur tambahan yaitu blog dan edukasi.

Dalam membangun aplikasi tersebut, penelitian ini bertujuan untuk melakukan perancangan model dengan akurasi terbaik pada *rice leaf diseases dataset* dengan arsitektur CNN dengan cara membandingkan tiga arsitektur, yaitu ResNet, MobileNet, dan Inception. Perancangan melalui 3 tahapan yaitu Persiapan Dataset (menyimpan di Kaggle dan Preprocessing Data), Pemilihan Layer untuk model, Training Dataset dengan model, dan Evaluasi atau pengujian akurasi dengan dataset itu sendiri menggunakan confusion matrix.

Kata kunci : *Rice Leaf Diseases*, CNN, MobileNet, Inception dan ResNet

SUMMARY

PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA *RICE DISEASES DATASET* UNTUK APLIKASI PADI CARE

Barra Malik Wibowo

Rice diseases can cause losses in the form of crop failure, which results in a decrease in rice crop production. Therefore, early detection of diseases in rice plants is very important to prevent greater losses. Padi Care is an application that can help in detecting rice diseases. Padi Care provides 1 main feature, namely detecting rice diseases and 2 additional features, namely blogs and education.

In building the application, this research aims to design a model with the best accuracy on rice leaf diseases dataset with CNN architecture by comparing three architectures, namely ResNet, MobileNet, and Inception. The design goes through 3 stages, namely Dataset Preparation (storing in Kaggle and Preprocessing Data), Layer Selection for the model, Dataset Training with the model, and Evaluation or accuracy testing with the dataset itself using a confusion matrix.

Keywords : Rice Leaf Diseases, CNN, MobileNet, Inception and ResNet.

PRAKATA

Puji syukur kehadiran Allah S.W.T. yang telah melimpahkan berkah dan rahmat-Nya Laporan Tugas Akhir dengan judul “Perancangan Model Deep Learning dengan Arsitektur CNN pada Dataset Rice Leaf Diseases untuk Capstone Project Aplikasi Padi Care” ini dapat disusun. Terimakasih kami sampaikan kepada seluruh pihak yang telah membantu terwujudnya Laporan Tugas Akhir ini, diantaranya: Dekan FT Unsoed, Wakil Dekan Akademik FT Unsoed, Kajur Teknik Elektro Unsoed, Sekretaris Jurusan Teknik Elektro Unsoed, bapak-ibu dosen Teknik Elektro Unsoed, teman-teman Teknik Elektro Unsoed dan pihak-pihak lain yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih banyak kekurangan dan jauh dari kesempurnaan, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan ini. Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Laporan Tugas Akhir ini membawa manfaat bagi pengembangan ilmu Teknik Elektro khususnya di bidang *Deep Learning*.

Purbalingga, 19 Desember 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN MOTTO DAN PERSEMBAHAN.....	iv
RINGKASAN.....	v
SUMMARY.....	vi
PRAKATA.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.4.1 Tujuan.....	4
1.4.2 Manfaat.....	4
1.5 Sistematika Penulisan.....	4
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terdahulu.....	6
2.2 Padi Care.....	8
2.3 Deep Learning.....	8
2.4 Convolutional Neural Networks.....	8
2.4.1 Layer Pembentuk CNN.....	9
2.5 Arsitektur CNN.....	11
2.5.1 ResNet.....	11
2.5.2 Inception.....	12
2.5.3 MobileNet.....	14
2.6 Pelatihan dan Pengujian.....	15
2.6.1 Confusion Matrix.....	15

2.6.2 Akurasi.....	16
2.7 Kaggle.....	16
2.8 Python.....	16
2.9 Tensorflow dan Keras.....	17
BAB 3 METODE PENELITIAN.....	18
3.1 Tempat Penelitian.....	18
3.2 Alat dan Bahan.....	18
3.2.1 Hardware.....	18
3.2.2 Software.....	18
3.2.3 Dataset.....	18
3.3 Alur dan Tahap Penelitian.....	18
3.3.1 Studi Literatur.....	21
3.3.2 Tahap Perancangan.....	21
3.3.3 Tahap Pelatihan dan Pengujian.....	22
3.3.4 Tahap Akhir.....	22
3.4 Waktu dan Jadwal Penelitian.....	22
DAFTAR PUSTAKA.....	24
LAMPIRAN.....	28
BIODATA PENULIS.....	30

DAFTAR TABEL

Tabel 2.1: Penelitian Terdahulu.....	6
--------------------------------------	---

DAFTAR GAMBAR

Gambar 2.1 <i>Arsitektur CNN Sempel</i>	9
Gambar 2.2 <i>Pooling Layer</i>	10
Gambar 2.3 <i>Fully Connected Layers</i>	11
Gambar 2.4 <i>Arsitektur ResNet34</i>	12
Gambar 2.5 <i>Arsitektur Inception Sempel</i>	13
Gambar 2.6 <i>Arsitektur MobileNet</i>	14
Gambar 2.7 <i>Confusion Matrix</i>	15
Gambar 2.8 <i>Accuracy Formula</i>	16
Gambar 3.1 <i>Diagram alur penelitian</i>	19
Gambar 3.2 <i>Diagram Alur Penelitian</i>	20
Gambar 3.3 <i>Diagram alur perancangan model</i>	21

DAFTAR LAMPIRAN

Lampiran 1. Contoh Lampiran 1.....	28
Lampiran 2. Contoh Lampiran 2.....	29

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Padi adalah tanaman pangan utama di Indonesia dan beras yang dihasilkan dari padi merupakan makanan pokok bagi sebagian besar penduduk Indonesia. Permintaan akan beras terus meningkat seiring dengan pertumbuhan jumlah penduduk, dan perubahan pola makanan pokok. Padi memiliki nilai historis yang tinggi dan telah menjadi makanan pokok utama bagi Indonesia sejak lama. Meskipun sumber karbohidrat tidak hanya pada padi, namun padi tetap menjadi tanaman yang populer dan banyak diminati masyarakat di Indonesia. Produksi padi yang cukup untuk memenuhi kebutuhan pangan nasional menjadi tantangan utama, terutama dengan adanya alih fungsi lahan pertanian menjadi kawasan industri dan pemukiman yang menyebabkan penurunan produktivitas beras[1]–[3].

Padi rentan terhadap beberapa penyakit seperti hispa[4], blast[5], dan brown spot[6]. Penyakit ini dapat menyebabkan kerugian berupa gagal panen yang berimbas pada penurunan produksi tanaman padi. Penyakit blast, brown spot, dan hispa merupakan beberapa jenis penyakit pada tanaman padi yang dapat terjadi pada permukaan daun. Masing-masing penyakit memerlukan penanganan yang berbeda, dan jika tidak ditangani dengan benar, dapat menyebabkan kerugian berupa gagal panen. Oleh karena itu, deteksi dini penyakit pada tanaman padi sangat penting untuk mencegah kerugian yang lebih besar[7].

Dalam upaya mengatasi permasalahan ini, diperlukan untuk membuat sistem yang dapat mendeteksi penyakit pada tanaman padi secara efektif melalui deteksi visual menggunakan perangkat seluler, memberikan informasi pengendalian yang tepat, serta mudah diakses dan dipahami oleh petani. Penulis, bersama dengan kelompok Capstone Smart Farming dari mata kuliah Capstone Design, sedang mengembangkan aplikasi Padi Care yang dapat digunakan oleh petani untuk mendeteksi penyakit pada tanaman padi. Aplikasi ini menggunakan

teknologi deep learning untuk melakukan deteksi penyakit pada padi. *Deep learning* adalah cabang dari *machine learning* yang menggunakan algoritma yang terinspirasi dari struktur otak manusia untuk mempelajari data secara mendalam[8]. Deep learning dapat mengolah data gambar dengan menggunakan arsitektur CNN (*Convolutional Neural Network*), yang dapat mengekstraksi fitur penting dari gambar dan mengklasifikasikannya ke dalam kategori yang diinginkan[9]. Dalam penelitian ini saya akan menggunakan teknik *transfer learning*, yaitu teknik yang memanfaatkan model yang telah dilatih sebelumnya untuk kasus yang berbeda[10], [11]. Teknik ini dapat menghemat waktu komputasi dan sumber daya, serta meningkatkan kinerja model. Beberapa model yang dapat digunakan dengan teknik *transfer learning* adalah ResNet50[12], ResNet50V2[13], VGG-16[14], Inception[15], MobileNet[16], dan lainnya.

Dalam penelitian ini akan dilakukan perancangan sebuah model deep learning dengan arsitektur CNN pada *Rice Leaf Diseases Dataset* untuk Aplikasi Padi Care dengan akurasi terbaik. Metode yang akan digunakan adalah dengan membandingkan tiga arsitektur, yaitu ResNet50[12], [17]–[19], Inception[20], dan MobileNet[16], [21]. Berikut adalah alasan mengapa memilih ketiga arsitektur tersebut:

1. ResNet50 adalah arsitektur *Convolutional Neural Network* (CNN) yang memiliki 50 layer. ResNet50 memanfaatkan konsep *residual blocks* yang memungkinkan pelatihan jaringan saraf yang lebih dalam tanpa mengalami masalah gradien yang menghilang[22]. ResNet50 juga telah terbukti memiliki tingkat kesalahan yang rendah, yaitu sebesar 3.57% pada Top-5[23]. Oleh karena itu, ResNet50 sering digunakan dalam berbagai aplikasi *deep learning*.

2. Inception dirancang untuk mencapai akurasi tinggi, efisiensi komputasi, dan fleksibilitas. Arsitektur ini memanfaatkan teknik seperti *inception module* dan *factorized convolution*[24], yang memungkinkannya untuk mencapai keseimbangan yang tepat antara ketiga hal tersebut. Oleh karena itu, Inception menjadi pilihan yang tepat untuk berbagai aplikasi, termasuk deteksi penyakit tanaman[15], [20].

3. MobileNet dirancang khusus untuk perangkat mobile dan sistem embedded. Arsitektur ini memanfaatkan *depthwise convolution* dan *pointwise convolution*, yang memungkinkannya untuk berjalan efisien pada perangkat dengan sumber daya komputasi terbatas. Oleh karena itu, MobileNet menjadi pilihan yang tepat untuk aplikasi yang membutuhkan efisiensi komputasi tinggi[16], [25].

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang sudah dijelaskan di atas, maka rumusan masalah pada penelitian tugas akhir ini adalah sebagai berikut.

1. Bagaimana perancangan algoritma deep learning menggunakan arsitektur CNN dengan akurasi terbaik pada *rice leaf diseases* dataset?
2. Bagaimana pelatihan dan pengujian algoritma deep learning dengan arsitektur CNN pada *rice leaf diseases* dataset?
3. Bagaimana perbandingan hasil akurasi pelatihan dan pengujian beberapa arsitektur CNN pada *rice leaf diseases* dataset?

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut.

1. Arsitektur CNN pada penelitian ini menggunakan teknik transfer learning dengan 3 buah arsitektur, yaitu ResNet50, Inception, dan MobileNet.
2. Penelitian ini akan mengeksplorasi performa akurasi dari ketiga arsitektur tanpa mendalami ke struktur arsitektur.
3. *Output* yang dihasilkan berupa 4 *class* yaitu Brown Spot, Healthy, Hispa, Leaf Blast.
4. Perancangan model *deep learning* ini menggunakan *Framework Keras* dan *Tensorflow*.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan pembuatan Proposal Tugas Akhir ini adalah sebagai berikut.

1. Merancang algoritma *deep learning* menggunakan arsitektur CNN dengan akurasi terbaik pada *rice leaf diseases dataset*.
2. Melakukan pelatihan dan pengujian algoritma *deep learning* dengan arsitektur CNN pada *rice leaf diseases dataset*.
3. Melakukan perbandingan hasil akurasi pelatihan dan pengujian beberapa arsitektur CNN pada *rice leaf diseases dataset*.

1.4.2 Manfaat

Manfaat yang diharapkan dalam pelaksanaan Tugas Akhir sebagai berikut.

1. Bagi mahasiswa yang mampu menerapkan ilmu yang didapat pada mata kuliah yang bersangkutan untuk menyelesaikan tugas akhir.
2. Membantu dalam mendeteksi penyakit padi yang menyerang pertanian.
3. Membantu mahasiswa lain dalam studi literatur.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan penulis dalam tugas akhir ini adalah sebagai berikut.

Bab I Pendahuluan.

Pendahuluan ini berisikan mengenai judul penelitian, latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

Bab II Tinjauan Pustaka.

Tinjauan Pustaka ini berisikan berbagai gagasan yang menjadi landasan ide tentang penelitian terdahulu serta berbagai teori pendukung dalam proses penelitian.

Bab III Metode Penelitian.

Bab Metode Penelitian ini berisikan mengenai metode penelitian yang dilakukan oleh penulis, seperti waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, metode penelitian, alur penelitian, dan *timeline* penelitian.

Bab IV Hasil dan Pembahasan.

Bab Hasil dan Pembahasan membahas mengenai hasil pelatihan dan pengujian beberapa model deep learning dengan arsitektur CNN menggunakan metode transfer learning serta perbandingan hasil akurasi ketiga model tersebut.

Bab V Penutup.

Bab penutup berisikan mengenai kesimpulan dan saran dari hasil yang didapat dari penelitian yang dilaksanakan oleh penulis.

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada sub bab ini akan dilakukan perbandingan antara penelitian yang terdahulu dengan penelitian yang dilakukan peneliti. Perbandingan ini meliputi judul dari penelitian, rincian dari penelitian terdahulu serta rincian penelitian ini. Terdapat beberapa kesamaan pada penelitian terdahulu seperti metode dan objek yang digunakan. Berikut ini pada tabel 2.1 terdapat penelitian terdahulu yang dijadikan acuan penelitian :

Tabel 2.1: Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Penelitian Terdahulu	Perbedaan Penelitian
1	Alya Nur Afni [26]	Perancangan model <i>Deep learning</i> dengan arsitektur <i>CNN</i> pada <i>Dataset Drowsiness</i> untuk <i>Capstone Project Aplikasi Drive Safely (Drive.Ly)</i>	<ul style="list-style-type: none"> • Penelitian mengembangkan sistem deteksi kantuk • Penelitian menggunakan arsitektur <i>CNN</i> yang telah dilatih sebelumnya berupa VGG-16, MobileNet, MobileNetV2, dan sederhana. 	<ul style="list-style-type: none"> • Penelitian mengembangkan sistem deteksi penyakit padi • Penelitian menggunakan arsitektur <i>CNN</i> yang telah dilatih berupa ResNet, MobileNet, dan Inception.
2	Ulfah Nur Oktaviana, Ricky	Klasifikasi Penyakit Padi berdasarkan Citra	<ul style="list-style-type: none"> • Objek penelitian berupa 	<ul style="list-style-type: none"> • Objek penelitian berupa

	Hendrawan, Alfian Dwi Khoirul Annas, dan Galih Wasis Wicaksono[27]	Daun Menggunakan Model Terlatih Resnet101	<p>penyakit padi , yaitu Bacterial Leaf Blight, Brown Spot, dan Leaf Smut</p> <ul style="list-style-type: none"> • Penelitian menggunakan arsitektur CNN yang telah dilatih sebelumnya berupa ResNet. 	<p>penyakit padi, yaitu Hispa, Brown Spot, dan Blast.</p> <ul style="list-style-type: none"> • Penelitian menggunakan arsitektur CNN yang telah dilatih berupa ResNet, MobileNet, dan Inception.
3	Endang Anggiratih, Sri Siswanti, Saly Kurnia Octaviani, dan Arumsari[28]	Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 Dengan Transfer Learning	<ul style="list-style-type: none"> • Objek penelitian berupa penyakit padi , yaitu Bacterial Leaf, dan Brown Spot • Penelitian menggunakan arsitektur CNN yang telah dilatih sebelumnya berupa 	<ul style="list-style-type: none"> • Objek penelitian berupa penyakit padi, yaitu Hispa, Brown Spot, dan Blast. • Penelitian menggunakan arsitektur CNN yang

			EfficientNet.	telah dilatih berupa ResNet, MobileNet, dan Inception.
--	--	--	---------------	---

2.2 Padi Care

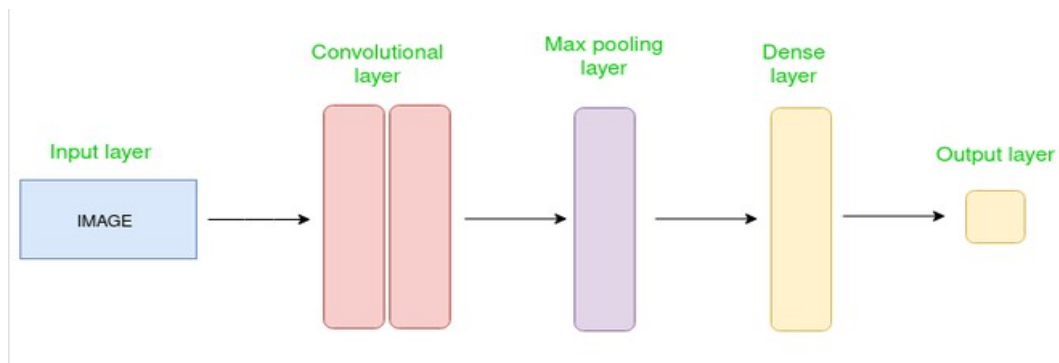
PadiCare merupakan sebuah aplikasi android untuk membantu para petani dengan fitur utama dari aplikasi ini yaitu deteksi penyakit padi dan memberikan informasi terkait dengan penanganannya.

2.3 Deep Learning

Deep learning adalah subbidang dari *machine learning* yang menggunakan algoritma berbasis *artificial neural networks* (ANN) dengan *representation learning*. Istilah "deep" merujuk pada penggunaan *multiple layers* dalam jaringan tersebut. *Deep learning* memerlukan jumlah data yang besar untuk memberikan hasil yang lebih baik daripada algoritma *machine learning* tradisional. *Deep learning* menggunakan struktur algoritma yang kompleks yang terinspirasi dari otak manusia untuk memproses data tak terstruktur seperti dokumen, gambar, dan teks[8], [29], [30].

2.4 Convolutional Neural Networks

Convolutional Neural Network (CNN) adalah versi *artificial neural networks* (ANN) yang sebagian besar digunakan untuk mengekstrak fitur dari kumpulan data matriks mirip grid. Misalnya kumpulan data visual seperti gambar atau video di mana pola data memainkan peran yang luas.



Gambar 2.1 Arsitektur CNN Sempel

Lapisan Konvolusional menerapkan filter ke gambar masukan untuk mengekstrak fitur, lapisan *Pooling* menurunkan sampel gambar untuk mengurangi komputasi, dan lapisan yang terhubung sepenuhnya membuat prediksi akhir.[31]

2.4.1 Layer Pembentuk CNN

Arsitektur *Convolution Neural Networks* yang lengkap juga dikenal sebagai CNN. CNN adalah rangkaian lapisan, dan setiap lapisan mengubah satu volume ke volume lainnya melalui fungsi terdiferensiasi. [31]

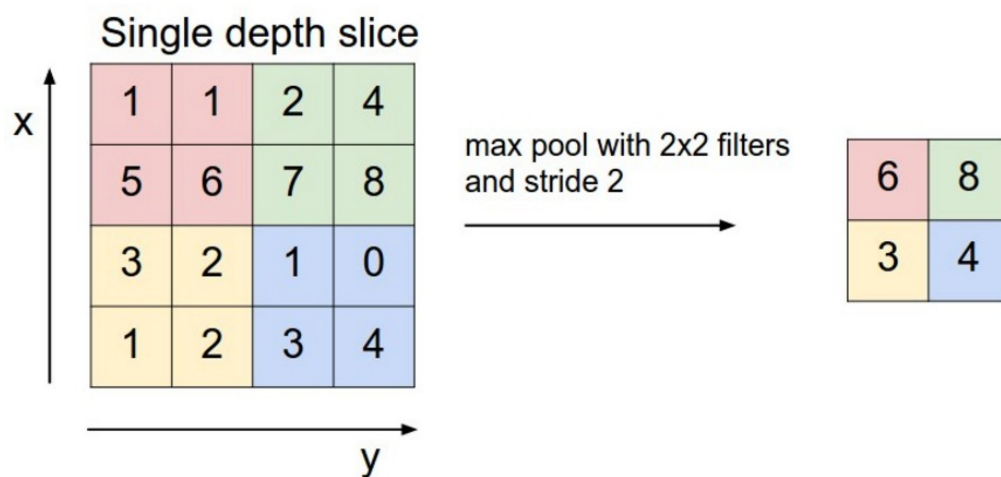
Jenis Lapisan:

Mari kita ambil contoh dengan menjalankan CNN pada gambar berdimensi 32 x 32 x 3.

- **Input Layer:** Ini adalah *layer* tempat kita memberikan input ke model kita. Di CNN, Umumnya inputnya berupa gambar atau rangkaian gambar. Lapisan ini menampung input mentah gambar dengan lebar 32, tinggi 32, dan kedalaman 3
- **Convolutional Layer:** Ini adalah *layer* yang digunakan untuk mengekstrak fitur dari kumpulan data input. Ini menerapkan serangkaian filter yang dapat dipelajari yang dikenal sebagai kernel ke input gambar. Filter/*kernel* adalah matriks yang lebih kecil biasanya berbentuk 2×2, 3×3, atau 5×5. itu meluncur di atas data gambar masukan dan menghitung perkalian titik antara bobot kernel dan *patch* gambar masukan yang sesuai. Output dari lapisan ini disebut peta fitur iklan. Misalkan kita

menggunakan total 12 filter untuk lapisan ini, kita akan mendapatkan volume keluaran berdimensi $32 \times 32 \times 12$.

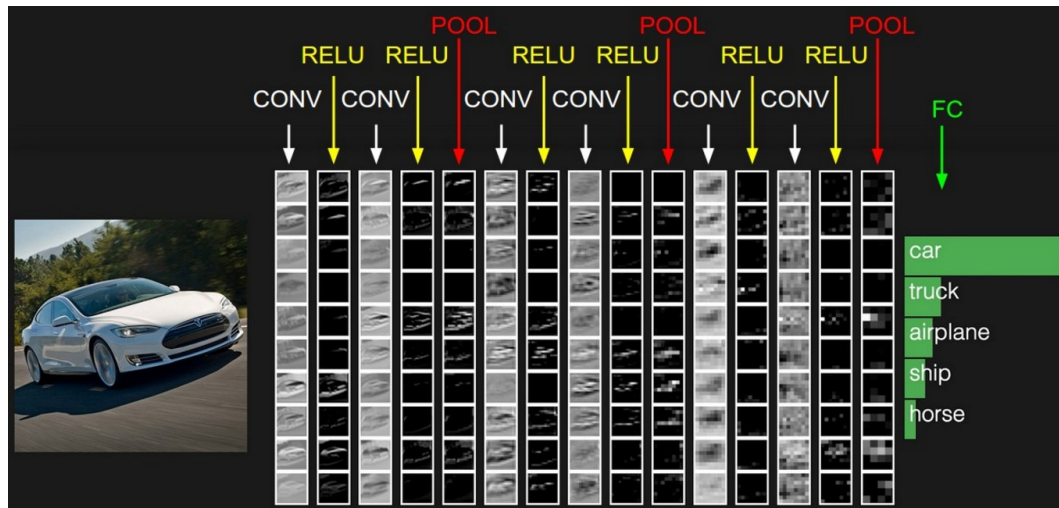
- **Activation Layer:** Dengan menambahkan *activation function* ke keluaran *layer* sebelumnya, *activation layer* menambah nonlinier pada jaringan. itu akan menerapkan *activation function* berdasarkan elemen ke keluaran *convolution layer*. Beberapa *activation function* yang umum adalah RELU: $\max(0, x)$, Tanh, Leaky RELU, dll. Volume tetap tidak berubah sehingga volume keluaran akan berdimensi $32 \times 32 \times 12$.
- **Pooling Layer:** Layer ini dimasukkan secara berkala ke dalam CNN dan fungsi utamanya adalah untuk memperkecil ukuran volume yang membuat komputasi menjadi cepat, mengurangi memori dan juga mencegah *overfitting*. Dua jenis *pooling layer* yang umum adalah penggabungan maksimal dan penggabungan rata-rata. Jika kita menggunakan *max pool* dengan filter 2×2 dan langkah 2, volume yang dihasilkan akan berdimensi $16 \times 16 \times 12$.



Gambar 2.2 Pooling Layer

- **Flattening:** *feature maps* yang dihasilkan diratakan menjadi vektor satu dimensi setelah *convolution* and *pooling layers* sehingga dapat diteruskan ke lapisan yang tertaut sepenuhnya untuk kategorisasi atau regresi.

- **Fully Connected Layers:** Mengambil masukan dari lapisan sebelumnya dan menghitung tugas klasifikasi atau regresi akhir.



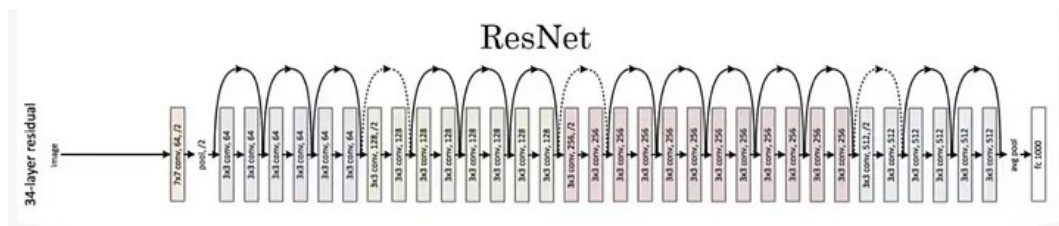
Gambar 2.3 Fully Connected Layers

- **Output Layer:** Output dari *fully connected layers* sepenuhnya kemudian dimasukkan ke dalam fungsi logistik untuk tugas klasifikasi seperti *sigmoid* atau *softmax* yang mengubah keluaran setiap kelas menjadi skor probabilitas setiap kelas.

2.5 Arsitektur CNN

2.5.1 ResNet

Arsitektur ResNet (*Residual Network*) adalah jenis arsitektur jaringan saraf yang telah merevolusi deep learning dengan memungkinkan pelatihan jaringan yang jauh lebih dalam dengan akurasi yang lebih baik dari sebelumnya. Arsitektur ResNet pertama kali diperkenalkan oleh He et al. dalam artikel mereka yang berjudul "Deep Residual Learning for Image Recognition" pada tahun 2015.



Gambar 2.4 Arsitektur ResNet34

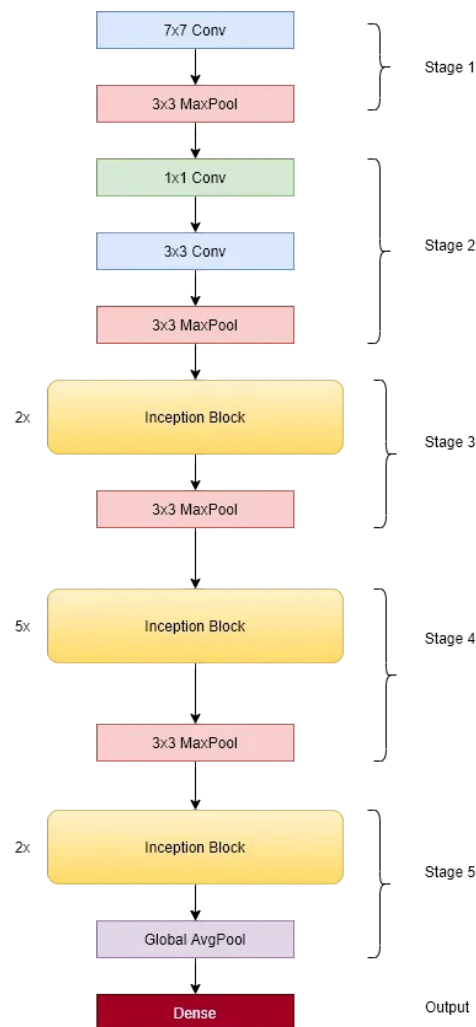
Arsitektur ResNet terdiri dari beberapa blok residu, yang masing-masing berisi banyak lapisan konvolusional, lapisan normalisasi batch, dan fungsi aktivasi. Blok residu ini dihubungkan dengan koneksi skip, yang memungkinkan informasi dari lapisan sebelumnya untuk diteruskan langsung ke lapisan berikutnya.

Keunggulan utama arsitektur ResNet adalah kemampuannya untuk mengatasi masalah *vanishing gradient*. Masalah *vanishing gradient* adalah masalah yang sering terjadi dalam *deep learning*, di mana gradien yang digunakan untuk memperbarui bobot jaringan cenderung menjadi sangat kecil seiring dengan bertambahnya kedalaman jaringan. Hal ini dapat menyebabkan pelatihan jaringan menjadi sulit atau bahkan tidak mungkin.

Koneksi skip dalam arsitektur ResNet membantu mengatasi masalah *vanishing gradient* dengan memungkinkan informasi dari lapisan sebelumnya untuk diteruskan langsung ke lapisan berikutnya. Hal ini memungkinkan gradien untuk mengalir lebih lancar melalui jaringan, sehingga pelatihan jaringan menjadi lebih mudah.[18], [32]

2.5.2 Inception

Arsitektur Inception adalah jenis arsitektur jaringan saraf konvolusional (CNN) yang diperkenalkan oleh Szegedy et al. dalam artikel mereka yang berjudul "Going Deeper with Convolutions" pada tahun 2015. Arsitektur ini telah menjadi salah satu arsitektur CNN paling populer dan sukses, dan telah digunakan untuk berbagai tugas *machine learning*, termasuk klasifikasi citra, pengenalan objek, dan segmentasi citra.



Gambar 2.5 Arsitektur Inception Sempel

Arsitektur Inception terdiri dari beberapa modul Inception, yang masing-masing menggabungkan beberapa operasi konvolusional ke dalam satu lapisan. Modul Inception ini dirancang untuk mengoptimalkan penggunaan sumber daya komputasi dan memori, sambil tetap mempertahankan kinerja yang tinggi.

Keunggulan utama arsitektur Inception adalah kemampuannya untuk mengatasi masalah *vanishing gradient*. Masalah *vanishing gradient* adalah masalah yang sering terjadi dalam pembelajaran mendalam, di mana gradien yang digunakan untuk memperbarui bobot jaringan cenderung menjadi sangat kecil seiring dengan bertambahnya kedalaman jaringan. Hal ini dapat menyebabkan pelatihan jaringan menjadi sulit atau bahkan tidak mungkin.

Modul Inception dalam arsitektur Inception membantu mengatasi masalah *vanishing gradient* dengan menggabungkan operasi konvolusional dengan ukuran filter yang berbeda ke dalam satu lapisan. Hal ini memungkinkan informasi dari berbagai skala untuk diproses secara bersamaan, sehingga gradien dapat mengalir lebih lancar melalui jaringan.[33]

2.5.3 MobileNet

Arsitektur MobileNet adalah jenis arsitektur jaringan saraf konvolusional (CNN) yang dirancang untuk efisiensi komputasi. Arsitektur ini diperkenalkan oleh Howard et al. dalam artikel mereka yang berjudul "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" pada tahun 2017.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
	Conv dw / s2	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 1024$
	Conv dw / s2	$3 \times 3 \times 1024$ dw
	Conv / s1	$1 \times 1 \times 1024 \times 1024$
	Avg Pool / s1	Pool 7×7
	FC / s1	1024×1000
	Softmax / s1	Classifier

Gambar 2.6 Arsitektur MobileNet

Arsitektur MobileNet terdiri dari beberapa blok MobileNet, yang masing-masing berisi beberapa lapisan konvolusional. Blok MobileNet ini dirancang

untuk mengurangi jumlah parameter dan operasi komputasi, sambil tetap mempertahankan kinerja yang tinggi.

Keunggulan utama arsitektur MobileNet adalah kemampuannya untuk mengurangi konsumsi daya dan memori. Hal ini menjadikan MobileNet ideal untuk aplikasi komputasi seluler, seperti pengenalan wajah, pengenalan objek, dan augmented reality.[26], [34], [35]

2.6 Pelatihan dan Pengujian

2.6.1 Confusion Matrix

Confusion Matrix adalah matriks yang merangkum performa model pembelajaran mesin pada sekumpulan data pengujian. Hal ini sering digunakan untuk mengukur kinerja model klasifikasi, yang bertujuan untuk memprediksi label kategorikal untuk setiap contoh masukan. Matriks tersebut menampilkan jumlah *true positif* (TP), *true negative* (TN), *false positif* (FP), dan *false negative* (FN) yang dihasilkan oleh model pada data pengujian.[26], [36]

Untuk klasifikasi biner, matriksnya akan berbentuk tabel 2x2, Untuk klasifikasi kelas jamak, bentuk matriksnya akan sama dengan jumlah kelasnya, yaitu untuk n kelas adalah nXn. *Confusion Matrix* 2x2 ditampilkan di bawah untuk pengenalan gambar yang memiliki gambar Anjing atau gambar Bukan Anjing.

		Actual	
		Dog	Not Dog
Predicted	Dog	True Positive (TP)	False Positive (FP)
	Not Dog	False Negative (FN)	True Negative (TN)

Gambar 2.7 Confusion Matrix

- *True Positive (TP)*: Ini adalah jumlah total yang memiliki nilai prediksi dan nilai aktual adalah Dog.
- *True Negative (TN)*: Ini adalah jumlah total yang memiliki nilai prediksi dan nilai aktual yang Bukan Anjing.
- *False Positive (FP)*: Jumlah total yang diprediksi adalah Anjing padahal sebenarnya Bukan Anjing.
- *False Negative (FN)*: Jumlah total yang diprediksi Bukan Anjing, padahal sebenarnya Anjing.

2.6.2 Akurasi

Akurasi klasifikasi adalah akurasi yang umumnya kami maksudkan, setiap kali kami menggunakan istilah akurasi. Kami menghitungnya dengan menghitung rasio prediksi yang benar terhadap jumlah total Sampel masukan.[37]

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total number of input samples}}$$

Gambar 2.8 Accuracy Formula

2.7 Kaggle

Kaggle adalah platform *cloud IDE* yang menyediakan berbagai macam fitur untuk membantu para *data scientist* dan praktisi *machine learning* dalam mengembangkan model *machine learning*. Fitur-fitur tersebut meliputi akses ke GPU dan TPU secara gratis, pustaka AI yang lengkap, fitur *debugging* yang kuat, dan fitur berbagi dan kolaborasi.[38]

2.8 Python

Python adalah bahasa pemrograman yang serbaguna dan mudah dipelajari. Python dapat digunakan untuk berbagai macam tugas, seperti ilmu data, pengembangan web, aplikasi desktop, dan aplikasi seluler. Python memiliki komunitas pengguna yang besar dan aktif, yang menyediakan beragam sumber daya untuk pembelajaran dan kolaborasi.[39]

2.9 Tensorflow dan Keras

Tensorflow adalah framework komputasi numerik open-source yang populer, dikembangkan oleh Google. TensorFlow digunakan untuk berbagai macam tugas, termasuk *machine learning*, komputasi kuantum, dan sinyal sinyal. TensorFlow dibangun di atas mesin tensor, yang memungkinkan representasi dan manipulasi data multi-dimensi secara efisien. Ini sangat cocok untuk *machine learning*, di mana data sering kali direpresentasikan sebagai tensor. TensorFlow juga mendukung operasi diferensiabel otomatis, yang memungkinkan Anda menghitung gradien dan memperbarui model parameter Anda secara efisien. Ini penting untuk melatih model *machine learning* untuk mencapai kinerja optimal. TensorFlow dapat berjalan di berbagai platform, mulai dari CPU dan GPU hingga akselerator khusus seperti TPU dan perangkat keras kopling. Ini memungkinkan Anda untuk menskalakan komputasi Anda sesuai dengan kebutuhan dan memanfaatkan sumber daya yang tersedia.[40]

Keras adalah pustaka deep learning tingkat tinggi yang dibangun di atas TensorFlow. Keras menawarkan antarmuka pemrograman tingkat tinggi (API) yang intuitif dan mudah dipelajari, memungkinkan Anda membangun dan melatih model pembelajaran mesin dengan kode yang lebih ringkas dan mudah dibaca. Keras menyediakan berbagai model dan lapisan pra-bangun yang siap digunakan, termasuk model klasik seperti regresi linier dan jaringan saraf tiruan, serta model modern seperti *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN). Keras juga bersifat modular, memungkinkan Anda dengan mudah menggabungkan lapisan dan model yang berbeda untuk membangun arsitektur yang kompleks dan disesuaikan untuk masalah tertentu.[41]

BAB 3

METODE PENELITIAN

3.1 Tempat Penelitian

Penelitian ini dilaksanakan dalam waktu 3 bulan bertempat di Kampus Fakultas Teknik Universitas Jenderal Soedirman, Kabupaten Purbalingga.

3.2 Alat dan Bahan

Selama penelitian, berikut adalah daftar alat dan bahan penunjang proses penelitian.

3.2.1 Hardware

1. Laptop Thinkpad L440
2. Redmi 9

3.2.2 Software

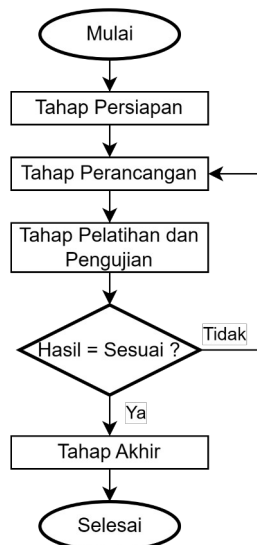
1. Kaggle Notebook
2. Visual Studio Code
3. Github
4. Google Drive

3.2.3 Dataset

Dataset yang terdiri dari 4 kategori yaitu *healthy*, *leaf blast*, *brownspot*, dan *hispa* yang didapatkan dari situs kaggle. *Rice Leafs dataset* – Shayan Riyaz dengan total data berjumlah 3355 data.

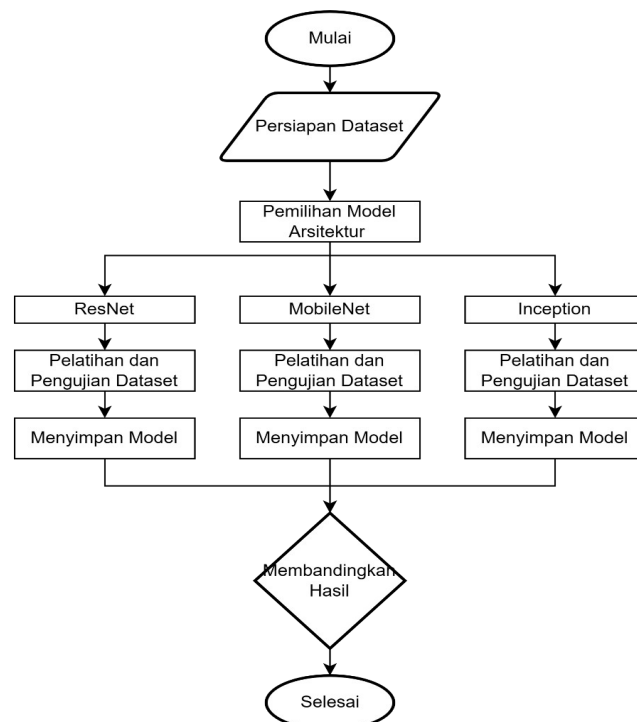
3.3 Alur dan Tahap Penelitian

Pada penelitian ini terdapat beberapa tahapan yaitu tahap persiapan, tahap perancangan, tahap pelatihan dan pengujian, serta tahap akhir. Pelaksanaan penelitian ini dapat digambarkan dalam bentuk diagram seperti pada gambar 3.1 berikut.



Gambar 3.1 Diagram Alur Penelitian

Sedangkan untuk diagram alur dari model yang dirancang pada penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Diagram Alur Perancangan Model

3.3.1 Studi Literatur

Pada tahap studi literatur mencari serta mengumpulkan referensi berupa jurnal, buku, artikel, maupun referensi lainnya yang memiliki keterkaitan dengan penelitian ini.

3.3.2 Tahap Perancangan

Pada tahap ini, hal pertama yang dilakukan adalah mendownload dataset dari kaggle ke local untuk dilakukan cleaning data augment data kemudian mengupload dataset yang sudah bersih ke kaggle kemudian mengimport dataset tersebut ke notebook kaggle. Selanjutnya membuat kode sumber untuk program CNN. Bahasa yang digunakan adalah bahasa python dengan Framework Keras dan Tensorflow serta dibuat menggunakan infrastruktur kaggle. Dataset dibagi menjadi dua, yaitu dataset pelatihan (train) dan dataset pengujian (test). pembagian dataset yaitu 80:20. Selanjutnya akan membuat model dari jaringan CNN yang sudah dipelajari sebelumnya, dalam penelitian ini, penulis menggunakan arsitektur MobileNet, Inception, dan Resnet untuk model jaringan CNN.

3.3.3 Tahap Pelatihan dan Pengujian

Pada tahap ini model akan dilatih dan diuji, apakah akurasi sesuai dengan yang diinginkan atau tidak. Apabila akurasi kurang sesuai maka penulis akan melakukan perbaikan dan pengkajian ulang. Akurasi dianggap tepat atau baik ketika nilai akurasi mendekati angka 1 dan nilai kesalahan atau loss mendekati angka 0. Pada tahap ini, beberapa model dibandingkan dengan akurasi yang paling baik kemudian model dikonversi atau disimpan ke bentuk .h5 untuk kemudian digunakan pada pembuatan aplikasi padicare.

3.3.4 Tahap Akhir

Tahap akhir pada penelitian ini dilakukan penyusunan laporan dengan judul “PERANCANGAN MODEL *DEEP LEARNING* DENGAN ARSITEKTUR CNN PADA *RICE DISEASES DATASET* UNTUK APLIKASI PADI CARE” dan memaparkan hasil penelitian dalam bentuk presentasi serta pembukuan laporan.

[illegible]

DAFTAR PUSTAKA

- [1] “Konsumsi Beras Penduduk RI Meningkat sejak Pandemi | Databoks.” Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://databoks.katadata.co.id/datapublish/2022/09/22/konsumsi-beras-penduduk-ri-meningkat-sejak-pandemi>
- [2] S. Suwarno, “Meningkatkan Produksi Padi Menuju Ketahanan Pangan yang Lestari,” *J. PANGAN*, vol. 19, no. 3, Art. no. 3, 2010, doi: 10.33964/jp.v19i3.150.
- [3] F. S. Wiange Mimanda, “PENGARUH DOSIS BOKASHI JERAMI PADI TERHADAP PERTUMBUHAN DAN HASIL PADI (*Oryza sativa* L.) METODE HAZTON DALAM SISTEM JARWO 2:1,” diploma, Universitas Andalas, 2022. Diakses: 29 November 2023. [Daring]. Tersedia pada: <http://scholar.unand.ac.id/101741/>
- [4] L. Hazarika dan K. C. Puzari, “Field efficacy of white muscardine fungus (*Beauveria bassiana*) on rice hispa (*Dicladyspa armigera*),” *Indian J. Agric. Sci.*, vol. 67, hlm. 463–465, Okt 1997.
- [5] D. B N dkk., “Understanding the Dynamics of Blast Resistance in Rice-Magnaporthe oryzae Interactions,” *J. Fungi — Open Access Mycol. J.*, Mei 2022, doi: 10.3390/jof8060584.
- [6] V. V. Krishnamurthi, T. Manoharan, dan K. Kumar, “Efficacy of certain botanicals against rice leaf folder and brownspot incidence,” vol. 23, hlm. 11–12, Agu 1999.
- [7] “Penggunaan Pestisida Dalam Perspektif Produksi dan Keamanan Pangan.” Diakses: 29 November 2023. [Daring]. Tersedia pada: <https://tanamanpangan.pertanian.go.id/detil-konten/iptek/16>
- [8] “What is Deep Learning? | IBM.” Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://www.ibm.com/topics/deep-learning>
- [9] “What Is a Convolutional Neural Network? A Beginner’s Tutorial for Machine Learning and Deep Learning,” freeCodeCamp.org. Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/>
- [10] “Transfer Learning Definition | DeepAI.” Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://deepai.org/machine-learning-glossary-and-terms/transfer-learning>
- [11] “ML | Introduction to Transfer Learning,” GeeksforGeeks. Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/>
- [12] “Module: tf.keras.applications.resnet50 | TensorFlow v2.13.0.” [Daring]. Tersedia pada: https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50?hl=en

- [13] “tf.keras.applications.resnet_v2.ResNet50V2 | TensorFlow v2.13.0.” [Daring]. Tersedia pada: https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet_v2/ResNet50V2?hl=en
- [14] “tf.keras.applications.vgg16.VGG16 | TensorFlow v2.13.0.” Diakses: 23 September 2023. [Daring]. Tersedia pada: https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg16/VGG16
- [15] “tf.keras.applications.inception_v3.InceptionV3 | TensorFlow v2.13.0.” Diakses: 23 September 2023. [Daring]. Tersedia pada: https://www.tensorflow.org/api_docs/python/tf/keras/applications/inception_v3/InceptionV3
- [16] “tf.keras.applications.mobilenet.MobileNet | TensorFlow v2.13.0.” [Daring]. Tersedia pada: https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet/MobileNet?hl=en
- [17] K. He, X. Zhang, S. Ren, dan J. Sun, “Identity Mappings in Deep Residual Networks,” *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, vol. 9908 LNCS, hlm. 630–645, Mar 2016, doi: 10.1007/978-3-319-46493-0_38.
- [18] K. He, X. Zhang, S. Ren, dan J. Sun, “Deep Residual Learning for Image Recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, hlm. 770–778, Des 2015, doi: 10.1109/CVPR.2016.90.
- [19] “Deep Learning menggunakan Transfer Learning -Python Code untuk ResNet50.” [Daring]. Tersedia pada: <https://ichi.pro/id/deep-learning-menggunakan-transfer-learning-python-code-untuk-resnet50-152617287822648>
- [20] Z. Qiang, L. He, dan F. Dai, “Identification of Plant Leaf Diseases Based on Inception V3 Transfer Learning and Fine-Tuning,” dalam *Smart City and Informatization*, G. Wang, A. El Saddik, X. Lai, G. Martinez Perez, dan K.-K. R. Choo, Ed., dalam Communications in Computer and Information Science. Singapore: Springer, 2019, hlm. 118–127. doi: 10.1007/978-981-15-1301-5_10.
- [21] A. Howard dkk., “Searching for MobileNetV3,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, hlm. 1314–1324, Mei 2019, doi: 10.1109/ICCV.2019.00140.
- [22] “ResNet-50: The Basics and a Quick Tutorial,” Datagen. Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://datagen.tech/guides/computer-vision/resnet-50/>
- [23] “Understanding ResNet50 architecture,” OpenGenus IQ: Computing Expertise & Legacy. Diakses: 23 September 2023. [Daring]. Tersedia pada: <https://iq.opengenus.org/resnet50-architecture/>
- [24] C. Szegedy, S. Ioffe, V. Vanhoucke, dan A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Art. no. 1, Feb 2017, doi: 10.1609/aaai.v31i1.11231.

- [25] H. Nguyen, H. Hoang Luong, L. Huynh, B. Le, N. Doan, dan D. Le, "An Improved MobileNet for Disease Detection on Tomato Leaves," *Adv. Technol. Innov.*, vol. 8, hlm. 192–209, Jul 2023, doi: 10.46604/aiti.2023.11568.
- [26] A. Afni, "PERANCANGAN MODEL DEEP LEARNING DENGAN ARSITEKTUR CNN PADA DATASET DROWSINESS UNTUK CAPSTONE PROJECT APLIKASI DRIVE SAFELY (DRIVE.LY)".
- [27] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, dan Galih Wasis Wicaksono, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 5, no. 6, hlm. 1216–1222, Des 2021, doi: 10.29207/resti.v5i6.3607.
- [28] E. Anggiratih, S. Siswanti, S. K. Octaviani, dan A. Sari, "Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 dengan Transfer Learning," *J. Ilm. SINUS*, vol. 19, no. 1, Art. no. 1, Jan 2021, doi: 10.30646/sinus.v19i1.526.
- [29] "Deep learning," *Wikipedia*. 26 November 2023. Diakses: 29 November 2023. [Daring]. Tersedia pada: https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=1186902970
- [30] "Deep Learning vs. Machine Learning: A Beginner's Guide," Coursera. Diakses: 29 November 2023. [Daring]. Tersedia pada: <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>
- [31] "Introduction to Convolution Neural Network," GeeksforGeeks. Diakses: 18 Desember 2023. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
- [32] P. Mahajan, "Understanding ResNet Architecture," Analytics Vidhya. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://medium.com/analytics-vidhya/understanding-resnet-architecture-869915cc2a98>
- [33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, dan Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," arXiv.org. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://arxiv.org/abs/1512.00567v3>
- [34] A. Sarkar, "Building MobileNet from Scratch Using TensorFlow," Medium. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://towardsdatascience.com/building-mobilenet-from-scratch-using-tensorflow-ad009c5dd42c>
- [35] A. G. Howard dkk., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv.org. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://arxiv.org/abs/1704.04861v1>
- [36] "Confusion Matrix in Machine Learning," GeeksforGeeks. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>
- [37] "Evaluation Metrics in Machine Learning," GeeksforGeeks. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/metrics-for-machine-learning-model/>

- [38] M. Gusarova, "What is Kaggle, and how can I use it as a beginner data scientist?," Medium. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://medium.com/@data.science.enthusiast/what-is-kaggle-com-why-should-i-use-it-as-a-beginner-data-scientist-how-do-i-get-started-with-it-7e218266fe6d>
- [39] "Applications for Python," Python.org. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://www.python.org/about/apps/>
- [40] "TensorFlow basics | TensorFlow Core." Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://www.tensorflow.org/guide/basics>
- [41] "Keras: The high-level API for TensorFlow | TensorFlow Core." Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://www.tensorflow.org/guide/keras>
- [42] "Identitas," Universitas Jenderal Soedirman. Diakses: 30 Januari 2018. [Daring]. Tersedia pada: <http://unsoed.ac.id/id/menu/identitas>
- [43] Badan Pengembangan dan Pembinaan Bahasa, "Hasil Pencarian - KBBI Daring," *Kamus Besar Bahasa Indonesia Daring*. Kementerian Pendidikan dan Kebudayaan Republik Indonesia, 2016. Diakses: 5 Januari 2017. [Daring]. Tersedia pada: <http://kbbi.kemdikbud.go.id/entri/tabel>
- [44] LibreOffice Help user WikiSysop, "Creating a Bibliography," Welcome to LibreOffice Help! Diakses: 5 Januari 2017. [Daring]. Tersedia pada: https://help.libreoffice.org/Writer/Creating_a_Bibliography
- [45] Center for History and New Media, "Zotero Quick Start Guide." [Daring]. Tersedia pada: http://zotero.org/support/quick_start_guide

LAMPIRAN

Lampiran 1. Contoh Lampiran 1

Ini adalah isi dari lampiran 1

Lampiran 2. Contoh Lampiran 2

Ini adalah isi dari lampiran 2

BIODATA PENULIS

Pas photo ukuran
3 x 4 cm
berjas

Biodata penulis berisi terkait dengan identitas penulis (nama, kontak email), riwayat akademis (pendidikan) penulis ditulis dari yang paling , skill, serta prestasi penulis.

A. Identitas

Nama :
NIM :
Tempat, tanggal lahir :
Alamat :
No. Telp. :
Alamat e-mail :

B. Riwayat Pendidikan Akademik

Periode	Jenjang	Institusi
2012 – 2016	S1	Teknik Elektro Universitas Jenderal Soedirman
2009 – 2012	SMA	SMAN xx Purwokerto
2006 – 2009	SMP	SMPN xx Sokaraja

C. Riwayat Pendidikan Non Formal (jika ada)

Tahun	Keahlian	Penyelenggara	Kota
2014	Bahasa Inggris Tingkat Mahir	Lembaga Kursus xxxxx	Purwokerto
2013	Kemanan Jaringan Mikrotik Tingkat Mahir	Lembaga xxxxxxxx	Jakarta

D. Prestasi

Tahun	Tingkat	Prestasi
2014	Nasional	Juara 1 lomba penulisan karya ilmiah, Yogyakarta
2013	Internasional	Medali emas olimpiade sains internasional, Dakka, India

E. Keahlian (tuliskan secara diskriptif)

Memiliki minat di bidang pengembangan perangkat tertanam. Mampu merancang sistem embedded berdasarkan mikro kontroler atmega, arduino dan ESP8266.

Terlibat secara aktif dalam kegiatan asistem Laboratorium Sistem Telekomunikasi dan Informasi sebagai asisten praktikum Algoritma dan Struktur Data, Jaringan Komputer, dan Dasar Pemrograman.