

**ANALISIS ARSITEKTUR DEEP LEARNING VGG UNTUK
KLASIFIKASI JENIS JAMUR**

SKRIPSI

RAHMAD SYUHADA

198160001



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 6/12/23

Access From (repository.uma.ac.id)6/12/23

**ANALISIS ARSITEKTUR DEEP LEARNING VGG UNTUK
KLASIFIKASI JENIS JAMUR**

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana di Fakultas Teknik
Universitas Medan Area



Oleh:

RAHMAD SYUHADA

198160001

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

MEDAN

2023

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

i

Document Accepted 6/12/23

Access From (repository.uma.ac.id)6/12/23

HALAMAN PENGESAHAN

Judul Skripsi : Analisis Arsitektur Deep Learning VGG Untuk Klasifikasi Jenis
Jamur
Nama : Rahmad Syuhada
NPM : 198160001
Fakultas : Teknik
Prodi : Teknik Informatika

Disetujui Oleh :

Pembimbing



Muhathir, S.T., M.Kom

Diketahui Oleh :

Dekan Fakultas Teknik

Ketua Prodi Teknik Informatika



Dr. Ngada Nanda S.Kom, M.Kom
NIDN : 0109058804



Rizka Nurroza S.Kom, M.Kom
NIDN : 0109038902

Tanggal Lulus : 26 September 2023

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian – bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi – sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.



HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, Saya bertanda tangan dibawah ini:

Nama : Rahmad Syuhada
NPM : 198160001
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

Analisis Arsitektur Deep Learning VGG Untuk Klasifikasi Jamur.

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (*Database*), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan

Pada Tanggal : 13 Agustus 2023

Yang Menyatakan



Rahmad Syuhada
NPM 198160001

ABSTRAK

Indonesia merupakan salah satu pusat keanekaragaman hayati, Salah satu keanekaragaman jenis tumbuhan adalah jamur. Jamur adalah tumbuhan yang sangat sederhana, memiliki inti, spora, tanpa klorofil, berbentuk sel atau filamen bercabang dengan dinding selulosa atau khitin atau keduanya, Identifikasi jamur masih sulit dilakukan karena banyaknya jenis jamur, kurangnya pengetahuan tentang jamur, dan kurangnya ahli dalam bidang jamur. Selain itu, sebagian besar jamur memiliki tingkat kesamaan yang tinggi dalam karakteristik tertentu, yang menyulitkan dalam mengidentifikasi jenis jamur secara visual oleh manusia yang tidak memiliki keahlian khusus. Oleh karena itu, penting untuk dapat mengklasifikasikan jenis jamur sehingga diharapkan masyarakat lebih paham tentang jenis masing-masing jamur. Penelitian ini menggunakan pendekatan transfer learning dengan arsitektur VGG-19 untuk menyediakan metode yang akurat dalam mengklasifikasi jenis jamur. Terdapat 18 skenario model yang di training dan diperoleh performa model terbaik menggunakan hyperparameter jumlah epoch 50, batch size 64, dan optimizer SGD mendapat akurasi 77.3% pada proses training. Setelah di uji menggunakan data testing dan di evaluasi menggunakan confusion matrix dan classification report, diperoleh score accuracy sebesar 70%.

Kata Kunci : Klasifikasi Jamur, Deep Learning, VGG-19

ABSTRACT

Indonesia is one of the centers of biodiversity. One of the diversity of plant species is mushrooms. Fungi are very simple plants, having a nucleus, spores, without chlorophyll, in the form of cells or branching filaments with cellulose or chitin walls or both. Identification of fungi is still difficult because of the many types of fungi, lack of knowledge about fungi, and lack of experts in the field of mushrooms. In addition, most fungi have a high degree of comfort in certain characteristics, which poses a challenge in visually identifying the type of mushroom by an unskilled human. Therefore, it is important to be able to classify the types of mushrooms so that it is hoped that people will understand more about the types of each mushroom. This study uses a transfer learning approach with the VGG-19 architecture to provide an accurate method for classifying fungal species. There were 18 model scenarios that were trained and obtained the best model performance using the hyperparameter number of epochs of 50, batch size of 64, and the SGD optimizer got 77.3% accuracy in the training process. After being tested using data testing and evaluated using the confusion matrix and classification report, a score accuracy of 70% was obtained.

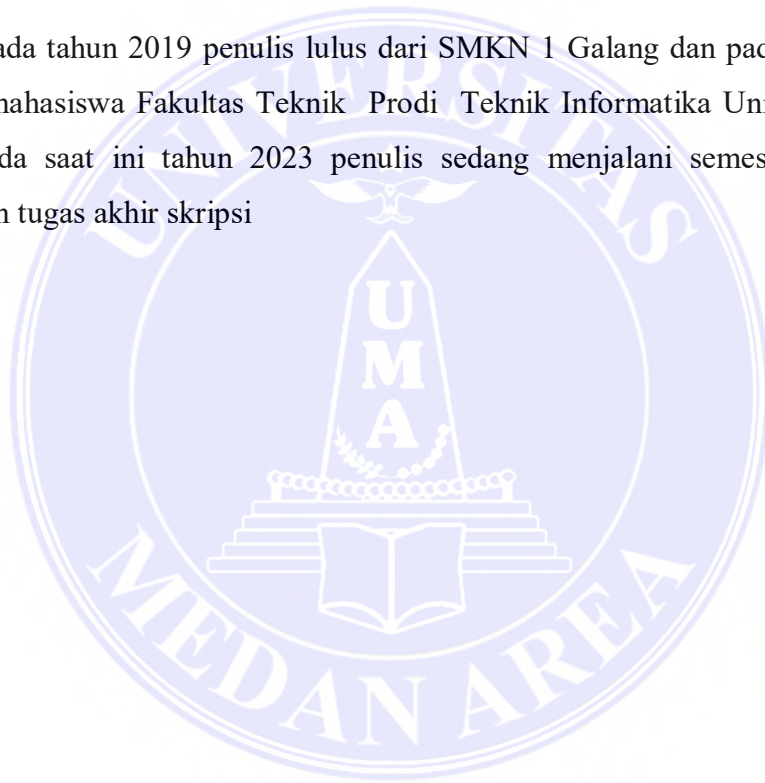
Keyword : *Mushroom classification, Deep Learning, VGG-19*

RIWAYAT HIDUP

Penulis lahir di Desa Kelapa Satu pada tanggal 17 Oktober 2000 dari Ayah Misno dan Ibu Sartika br.Manalu. Penulis adalah anak pertama dari 3 (tiga) bersaudara

Penulis pertama kali mengenyam Pendidikan dibangku SD 010 Siberuang pada tahun 2007-2013, meneruskan Pendidikan di Sekolah Menengah Pertama Negeri 1 Galang diselesaikan pada tahun 2013-2016, Meneruskan Pendidikan Sekolah Menengah Kejuruan Negeri 1 Galang pada tahun 2016-2019.

Pada tahun 2019 penulis lulus dari SMKN 1 Galang dan pada 2019 terdaftar sebagai mahasiswa Fakultas Teknik Prodi Teknik Informatika Universitas Medan Area. Pada saat ini tahun 2023 penulis sedang menjalani semester 8 (Delapan) menyusun tugas akhir skripsi



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Kuasa atas segala karuniaNya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian ini dengan judul “Analisis Arsitektur Deep Learning VGG Untuk Klasifikasi Jenis Jamur”.

Skripsi ini merupakan salah satu syarat untuk menyelesaikan pendidikan untuk mencapai gelar sarjana di Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area. Pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc. selaku Rektor Universitas Medan Area.
2. Bapak Dr. Rahmad Syah, S.Kom, M. Kom selaku Dekan Fakultas Teknik Universitas Medan Area.
3. Bapak Rizki Muliono, S.Kom, M.Kom selaku Kepala Program Studi Teknik Informatika Universitas Medan Area.
4. Bapak Muhathir,S.T,M.Kom, selaku Dosen pembimbing yang telah membantu penulis dari segi materi sehingga penulis dapat menyelesaikan skripsi ini.
5. Orang tua penulis yaitu Bapak Misno dan Ibu Sartika br.Manalu yang telah mendoakan tiada henti dan memberikan semangat serta membantu penulis dalam segi materi dan moril sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik baiknya.
6. Seluruh Dosen dan Staf Program Studi Teknik Informatika Universitas Medan Area.
7. Seluruh teman-teman yang sudah memberikan dukungannya selama penulisan proposal skripsi ini, khususnya teman-teman Teknik Informatika angkatan 2019.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu yang membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa penelitian ini masih memiliki kekurangan,oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi

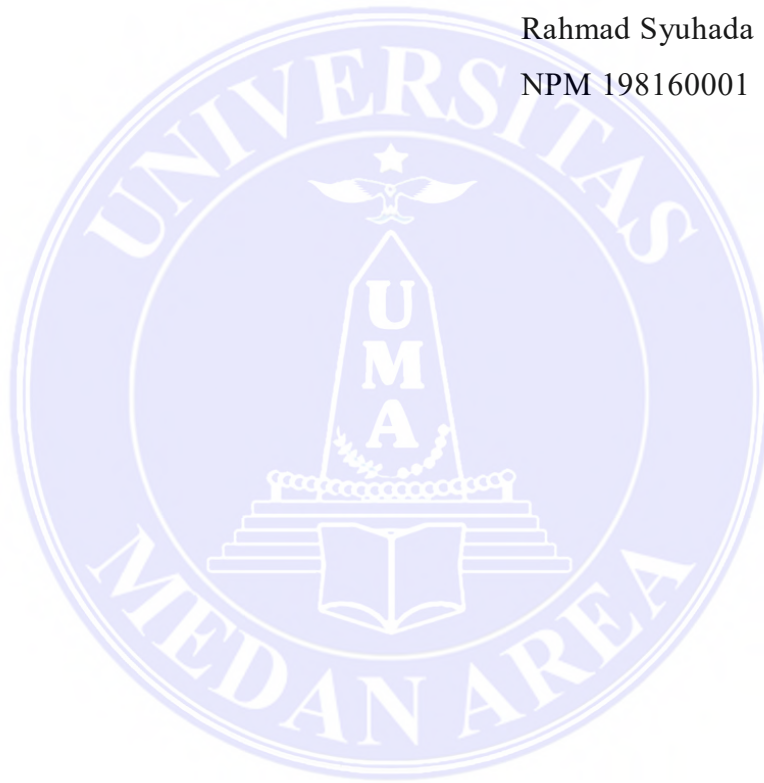
kesempurnaan penelitian ini. Penulis berharap tugas penelitian ini dapat bermanfaat baik kalangan pendidikan maupun masyarakat. Akhir kata penulis ucapkan terima kasih.

Medan, 13 November 2023

Penulis,

Rahmad Syuhada

NPM 198160001



DAFTAR ISI

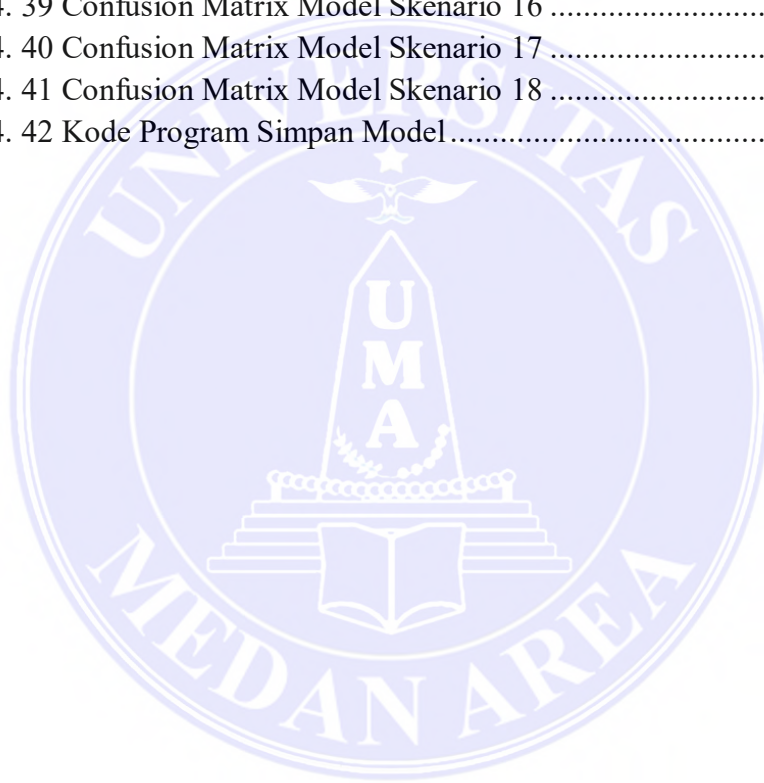
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN.....	iii
ABSTRAK	v
ABSTRACT	vi
RIWAYAT HIDUP	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I. PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan.....	7
BAB II. TINJAUAN PUSTAKA	8
2.1 Jamur	9
2.2 Deep Learning	9
2.3 Convolutional Neural Network.....	9
2.3.1 Convlotional Layer	11
2.3.2 Activation Function	12
2.3.3 Pooling Layer	13
2.3.4 Flatten Layer	14
2.3.5 Fully-Connected Layer	14
2.4 Arsitektur VGG-19.....	15
2.5 Confusion Matrix	16
2.6 Penelitian Terkait	18
BAB III. METODOLOGI PENELITIAN.....	21
3.1 Waktu dan Tempat Penelitian	21
3.2 Alat dan Bahan Penelitian.....	21
3.3 Jenis dan Sumber Data	22
3.4 Sampel Penelitian	22

3.5 Metode Analisis Data	23
3.6 Initialization Hyperparameter	24
3.7 Tahapan Penelitian	24
BAB IV. HASIL DAN PEMBAHASAN	Error! Bookmark not defined.
4.1 Hasil.....	Error! Bookmark not defined.
4.1.1 Preprocessing Dataset	28
4.1.2 Persiapan Dataset.....	28
4.1.3 Pembagian Dataset	31
4.1.4 Augmentasi Dataset.....	32
4.2 Pembahasan.....	35
4.2.1 Pemodelan Arsitektur VGG-19.....	35
4.2.2 Skenario Model	37
4.2.3 Hasil Training Model.....	39
4.2.4 Hasil Evaluasi Model.....	53
4.2.5 Hasil Pengujian Sistem	62
4.2.6 Penelitian Terdahulu	65
BAB V. KESIMPULAN DAN SARAN.....	67
5.1 Kesimpulan	67
5.2 Saran.....	67
Lampiran	72

DAFTAR GAMBAR

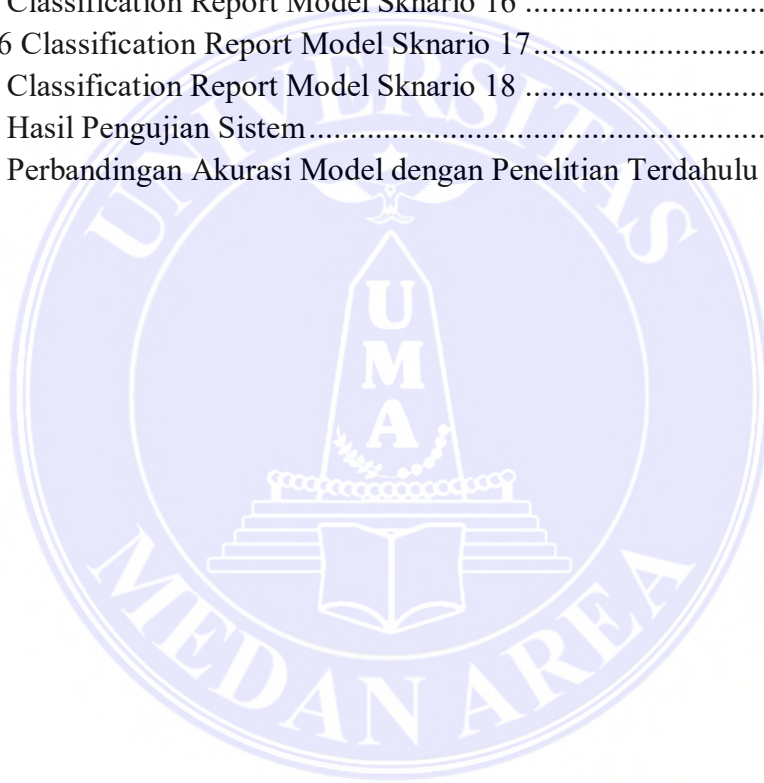
Gambar 2. 1 Arsitektur Convolutional Neural Network	10
Gambar 2. 2 Convolutional Layer	Error! Bookmark not defined.
Gambar 2. 3 Pooling Layer dengan Max Pooling dan Average Pooling	13
Gambar 2. 4 Global Average Pooling	14
Gambar 2. 5 Fully Connected Layer	15
Gambar 2. 6 Arsitektur VGG-19	16
Gambar 2. 7 Model Confusion Matrix	17
Gambar 3.1 Sampel Dataset	23
Gambar 3.2 Tahapan Penelitian	24
Gambar 4. 1 Sampel Dataset Kualitas Rendah	28
Gambar 4. 2 Proses Cropping Dataset	29
Gambar 4. 3 Proses Pelabelan Dataset	30
Gambar 4. 4 Sampel Hasil Pelabelan Dataset	31
Gambar 4. 5 Kode Program Augmentasi Dataset	33
Gambar 4. 6 Hasil Augmentasi dengan Rotation	33
Gambar 4. 7 Hasil Augmentasi dengan Width Shift	34
Gambar 4. 8 Hasil Augmentasi dengan Height Shift	34
Gambar 4. 9 Hasil Augmentasi dengan Shear	34
Gambar 4. 10 Hasil Augmentasi dengan Zoom	35
Gambar 4. 11 Hasil Augmentasi dengan Vertical Flip	35
Gambar 4. 12 Rancangan Arsitektur VGG-19	36
Gambar 4. 13 Kode Program untuk Base Model VGG-19	39
Gambar 4. 14 Kode Program untuk Create Model	39
Gambar 4. 15 Model Summary	40
Gambar 4. 16 Kode Program Compile dan Fitting Model Skenario 1	41
Gambar 4. 17 Proses Fitting Model Skenario 1	42
Gambar 4. 18 Grafik Accuracy dan Loss Model Skenario 1	43
Gambar 4. 19 Grafik Accuracy dan Loss Model Skenario 2	43
Gambar 4. 20 Grafik Accuracy dan Loss Model Skenario 3	44
Gambar 4. 21 Grafik Accuracy dan Loss Model Skenario 4	45
Gambar 4. 22 Grafik Accuracy dan Loss Model Skenario 5	45
Gambar 4. 23 Grafik Accuracy dan Loss Model Skenario 6	46
Gambar 4. 24 Grafik Accuracy dan Loss Model Skenario 7	46
Gambar 4. 25 Grafik Accuracy dan Loss Model Skenario 8	47
Gambar 4. 26 Grafik Accuracy dan Loss Model Skenario 9	48
Gambar 4. 27 Grafik Accuracy dan Loss Model Skenario 10	48
Gambar 4. 28 Grafik Accuracy dan Loss Model Skenario 11	49

Gambar 4. 29 Grafik Accuracy dan Loss Model Skenario 12	49
Gambar 4. 30 Grafik Accuracy dan Loss Model Skenario 13	50
Gambar 4. 31 Grafik Accuracy dan Loss Model Skenario 14	51
Gambar 4. 32 Grafik Accuracy dan Loss Model Skenario 15	51
Gambar 4. 33 Grafik Accuracy dan Loss Model Skenario 16	52
Gambar 4. 34 Grafik Accuracy dan Loss Model Skenario 17	52
Gambar 4. 35 Grafik Accuracy dan Loss Model Skenario 18	53
Gambar 4. 36 Grafik Perbandingan Validation Accuracy Model	54
Gambar 4. 37 Grafik Perbandingan Validation Loss Model.....	54
Gambar 4. 38 Confusion Matrix Model Skenario 15	56
Gambar 4. 39 Confusion Matrix Model Skenario 16	58
Gambar 4. 40 Confusion Matrix Model Skenario 17	59
Gambar 4. 41 Confusion Matrix Model Skenario 18	60
Gambar 4. 42 Kode Program Simpan Model.....	62



DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	18
Tabel 3.1 Deskripsi Kebutuhan Perangkat Keras	21
Tabel 3.2 Deskripsi Kebutuhan Perangkat Lunak	22
Tabel 3.3 Initialization Hyperparameter	24
Tabel 4. 1 Pembagian Dataset	32
Tabel 4. 2 Skenario Training Model.....	38
Tabel 4.3 Perbandingan Akurasi Skenario Model.....	54
Tabel 4.4 Classification Report Model Sknario 15	57
Tabel 4.5 Classification Report Model Sknario 16	58
Tabael 4.6 Classification Report Model Sknario 17	59
Tabel 4.7 Classification Report Model Sknario 18	61
Tabel 4.8 Hasil Pengujian Sistem.....	62
Tabel 4.9 Perbandingan Akurasi Model dengan Penelitian Terdahulu	65



BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu pusat keanekaragaman hayati dunia dan dianggap sebagai negara *megabiodiversity* (Angraini, 2018). Indonesia dikaruniai keanekaragaman hayati dan ekologi serta *organisme* dengan *endemisme* (keunikan) yang sangat tinggi dalam struktur geografis yang dapat dijadikan salah satu modal dasar pembangunan berkelanjutan (Suwarso, Paulus, & Widanirmala, 2019) . Keanekaragaman hayati berarti berbagai bentuk kehidupan yang terjadi di darat, di udara, dan di air dalam ruang dan waktu tertentu, baik berupa tumbuhan, hewan, maupun dalam bentuk makhluk hidup terkecil. seperti *mikroorganisme*. Salah satu keanekaragaman jenis tumbuhan adalah jamur. Jamur adalah tumbuhan yang sangat sederhana, memiliki inti, spora, tanpa klorofil, berbentuk sel atau filamen bercabang dengan dinding selulosa atau khitin atau keduanya, dan biasanya berkembang biak secara seksual dan aseksual (Suryani, 2020).

Jumlah jamur diseluruh dunia diperkirakan mencapai 1,5 juta, pada tahun 1997 ditemukan 70.000 spesies yang telah diidentifikasi, menyiratkan bahwa 1,43 juta masih belum terdeskripsikan (Hawksworth & Rossman, 1997). Jamur juga dibagi menjadi dua kelompok menurut ukurannya, yaitu jamur mikro, yaitu jamur yang strukturnya hanya dapat dilihat dengan mikroskop, dan jamur makro, yaitu jamur yang membentuk tubuh buah, yang dibagi lagi menjadi dua kelompok, yaitu jamur yang dapat dimakan, yaitu jamur yang bisa dimakan. ; misalnya *Pleurotus ostreatus* (jamur tiram), *Auricularia auricular* (jamur kuping) dan jamur lainnya yang beracun; misalnya *Amanita pallioides*, *Rusula emetik* dan lain-lain (Suryani, 2020).

Berdasarkan keunikan bentuk, tekstur, pola, ukuran jamur maka dalam penelitian ini mengujicobakan pengenalan jenis jamur berbasis komputerisasi dengan memanfaatkan metode deep learning. Berbagai macam metode deep learning seperti, *AlexNet*, *GoogleNet*, *ResNet18*, *ResNet50*, *Resnet101*, *InceptionV3*,

InceptionResNetV2, DenseNet20, XceptionNet, dan VGG. Arsitektur Visual Geometry Group (VGG) Merupakan hasil pengembangan dari *Alexnet* (Tanuwijaya & Roseanne, 2021). *VGG* diciptakan oleh Simonyan dan Andrew Zisserman di *Oxford University*. Model ini menjuarai kontes *ILSVRC* sebagai *1st Runner-up Image Classification* dan pemenang *Image Localization* pada tahun 2014 (Anggara, 2020). Arsitektur *VGG16* sendiri memiliki 13 layer lapisan konvolusi, dan 2 lapisan digunakan sebagai *fully_connected*, serta 1 lapisan klasifikasi. (Saputro, Mu'min, Lutfi, & Putri, 2022). Sedangkan model *VGG19* terdiri dari 19 layers yang terdiri dari 16 *convolutional layer*, 4 *max pooling layer*, 2 *fully connected layer*, dan 1 *softmax layer* (Bunyamin & Pangestu, 2018).

Penelitian tentang klasifikasi jenis jamur menjadi 2cena menarik dalam bidang pengenalan pola dan pengolahan citra, terutama karena peran penting jamur dalam ekosistem dan kehidupan manusia. Identifikasi jenis jamur dapat membantu dalam berbagai bidang seperti agrikultur dan kuliner. Dalam beberapa tahun terakhir, teknologi pengenalan pola berbasis *Deep Learning* telah menunjukkan prestasi yang signifikan dalam berbagai tugas pengenalan objek, termasuk klasifikasi citra. Selain *Deep Learning* terdapat juga metode lain yaitu *Machine Learning* yang dapat diterapkan untuk klasifikasi citra. Namun, performa *Deep Learning* lebih baik dibandingkan *Machine Learning*. Hal ini dibuktikan dari hasil penelitian yang dilakukan oleh (Amelia, Eosina, & Setiawan, 2018) dengan uji coba pada data penyakit kanker payudara. Penelitian menunjukkan bahwa hasil akurasi *Deep Learning* lebih tinggi daripada *Machine Learning*, sehingga metode *Deep Learning* yang akan diterapkan pada penelitian ini untuk klasifikasi jenis jamur.

Convolutional Neural Network (CNN) merupakan salah satu metode *Deep Learning* yang telah banyak di aplikasikan pada data citra. Salah satu arsitektur model *Deep Learning* menggunakan *CNN* adalah *Visual Geometry Group-19 (VGG-19)*. *VGG-19* merupakan perkembangan dari arsitektur *VGG-16* yang memiliki 19 lapisan mendalam sehingga kinerja yang dihasilkan menjadi lebih baik (Shinta, Jasril, Irsyad, Yanto, & Sanjaya, 2023). Kelebihan utama dari arsitektur *VGG-19* adalah

kemampuannya untuk mengekstraksi fitur-fitur kompleks dari citra dan memberikan representasi yang lebih baik daripada arsitektur sebelumnya. Dalam penelitian (Setiawan, 2019) membandingkan arsitektur *AlexNet*, *VGG-16*, *VGG-19*, *ResNet50*, *ResNet101*, *GoogLeNet*, *InceptionV3*, *InceptionResNetV2* dan *Squeezenet* dalam klasifikasi citra fundus. Hasil akurasi terbaik didapatkan oleh arsitektur *VGG-19* dengan nilai 89.3%. Dalam penelitian (Hindarto & Santoso, 2021) membandingkan arsitektur *DensetNet121*, *NasNetLarge*, *VGG-16* dan *VGG-19* untuk mengidentifikasi plat nomor kendaraan. Hasil akurasi terbaik didapatkan oleh arsitektur *VGG-19* dengan nilai 85.05%.

Penelitian lain dilakukan oleh (Dhaniswara, Kristian, & Setiawan, 2021) membandingkan arsitektur *VGG-19* dan Resnet untuk klasifikasi kematangan buah pisang. Hasil accuracy terbaik didapatkan oleh arsitektur *VGG-19* dengan nilai akurasi 100%. Dalam penelitian (Akbar, Aryani, & Ulum, 2022) membandingkan arsitektur *VGG-19* dan *AlexNet*, didapati hasil terbaik akurasi *VGG19* untuk mendeteksi citra banjir adalah 98.78%. Hasil kajian dari penelitian-penelitian terdahulu membuktikan arsitektur *VGG-19* memiliki performa yang sangat baik dalam melakukan tugas klasifikasi pada data citra. Hal ini menjadi rujukan dalam memilih *VGG-19* sebagai arsitektur model Deep Learning dalam melakukan klasifikasi jenis jamur pada penelitian ini.

Penelitian yang berkaitan dengan klasifikasi jenis jamur pada umumnya telah dilakukan oleh beberapa peneliti-peneliti sebelumnya. Diantara penelitian terdahulu seperti yang dilakukan oleh (Hanseliani & Adi, 2019) menggunakan metode Backpropagation untuk klasifikasi 6 jenis jamur. Hasil penelitian menunjukkan bahwa metode *Backpropagation* mampu mengklasifikasikan berbagai jenis jamur layak konsumsi dengan akurasi terbaik sebesar 97%. Dalam penelitian (Yohannes et al, 2021) menggunakan fitur *HOG* berbasis superpixel untuk klasifikasi 9 jenis jamur dengan metode *SVM*. Hasil pengujian yang didapat pada penelitian ini, superpixel dapat meningkatkan akurasi dari *SVM* pada dataset jamur dibandingkan tanpa superpixel dengan akurasi 83.29%.

Dalam penelitian (Sarimole & Diadi, 2022) menggunakan ekstraksi fitur GLCM dan KNN untuk klasifikasi 5 jenis jamur. Hasil penelitian setelah dilakukan 10 kali proses uji, tingkat akurasi tertinggi yang didapatkan sebesar 77%. Dalam penelitian (L.Chusna, Shalahudin, Riyanto, & Alexander, 2022) melakukan klasifikasi 5 jenis jamur layak konsumsi menggunakan algoritma Multiclass SVM dengan ekstraksi ciri orde pertama. Berdasarkan hasil pengujian diperoleh akurasi rata-rata sebesar 83%. Dalam penelitian (Rahmadhani & Marpaung, 2023) menggunakan metode CNN untuk klasifikasi 6 jenis jamur. Model pada penelitian menggunakan tiga convolution layer, tiga MaxPooling layer dan dua dropout layer. Pada tahap pengujian model mendapatkan akurasi sebesar 76%.

Berdasarkan pokok permasalahan serta hasil kajian dari penelitian terdahulu terkait dengan klasifikasi jenis jamur, maka perbedaan utama yang mendasari penelitian ini dengan penelitian sebelumnya terletak pada penerapan metode Deep Learning menggunakan CNN serta jumlah dataset dan jenis jamur yang diperbanyak untuk mengklasifikasi jamur. Dalam penelitian sebelumnya telah mencoba menggunakan berbagai metode untuk mengklasifikasi jenis jamur. Namun belum banyak peneliti yang menggunakan pendekatan transfer learning dengan arsitektur VGG-19 untuk melakukan pelatihan model.

Dengan latar belakang ini, penelitian tentang klasifikasi jenis jamur menggunakan pendekatan transfer learning dengan arsitektur VGG-19 menjadi sangat penting untuk menyediakan metode yang akurat untuk mengklasifikasi jenis jamur secara visual. Hal ini karena jumlah variasi bentuk dan ukuran jamur yang sangat beragam. Selain itu, beberapa spesies jamur memiliki kemiripan visual yang tinggi, menyebabkan kesulitan dalam membedakannya. Oleh karena itu, diperlukan dataset yang beragam yang mencakup berbagai jenis jamur untuk melatih model klasifikasi dengan baik. Hasil penelitian ini diharapkan dapat membantu dalam meningkatkan pemahaman tentang keragaman jenis jamur.

1.2 Perumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah yang telah diuraikan serta hasil dari penelitian sebelumnya, maka dapat dirumuskan pokok permasalahan yang akan dibahas dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana menerapkan arsitektur *Deep Learning VGG* dalam mengklasifikasi jenis jamur.
2. Bagaimana tingkat akurasi yang dihasilkan arsitektur *Deep Learning VGG* dalam mengklasifikasi jenis jamur.

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini lebih 5scena, maka diterapkan batasan-batasan masalah sehingga sesuai dengan tujuan yang ingin dicapai. Adapun batasan masalah dalam penelitian ini yaitu sebagai berikut:

1. Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari Kaggle dengan jumlah data 4500 *dataset* citra atau gambar jamur dengan format *.jpg/jpeg*.
2. *Dataset* terbagi menjadi 3 kategori yaitu data *training* dengan persentase 80%, data *validation* 15% dan sisanya 5 % sebagai data *testing*.
3. Hasil klasifikasi jenis jamur terbagi menjadi 9 kelas atau kategori, yaitu *agaricus*, *amanita*, *boletus*, *cortinarius*, *entoloma*, *hygrocybe*, *lactarius*, *russula*, dan *suillus*.
4. Terdapat 16 skenario *model* yang akan di training berdasarkan perbandingan parameter *batch size* (32 dan 64). Jumlah *epoch* (20 dan 50), *optimizer* (*Adam*, *RMSprop*, *SGD*, dan *Adadelta*) serta *learning rate* (0.01 dan 0.09). Sedangkan parameter dari *input shape* akan dijadikan statis pada setiap 5scenario model yang diuji.
5. Uji akurasi dilakukan menggunakan *confusion matrix* dan *classification report* untuk menghitung tingkat akurasi yang didapatkan oleh masing-masing skenario model, sehingga dapat ditentukan performa model terbaik yang selanjutnya akan di deploy kedalam aplikasi berbasis web.

7. Bahasa pemrograman yang digunakan untuk melakukan *training model* adalah *Python 3.10.11* menggunakan bantuan *Google Collaboratory*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah maka tujuan yang ingin dicapai dalam penelitian ini, yaitu sebagai berikut:

1. Membangun sebuah *model* yang dapat mengklasifikasi jenis jamur dengan menggunakan arsitektur *Deep Learning VGG-19*
2. Menganalisa performa dari tingkat akurasi *model* dari arsitektur *Deep Learning VGG-19* sehingga dapat menghasilkan klasifikasi jenis jamur secara akurat.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini dikategorikan berdasarkan penerapannya, yakni secara teoritis, praktis, dan akademis yaitu sebagai berikut:

1. Secara teoritis, penelitian ini akan memperkaya pengetahuan ilmiah tentang penggunaan arsitektur *VGG-19* dalam klasifikasi jenis jamur. Hasil penelitian ini dapat dijadikan sebagai acuan dalam pengembangan metode klasifikasi jamur menggunakan pendekatan *Deep Learning*.
2. Secara praktis, penelitian ini dapat membantu dalam pengembangan aplikasi atau sistem komputer yang dapat mengklasifikasi jenis jamur secara akurat. Hal ini akan mempermudah pengguna dalam mengidentifikasi jenis jamur yang ditemui.
3. Secara akademis, hasil penelitian ini dapat menambah bahan referensi untuk penelitian lain dibidang *Deep Learning* dengan menggunakan arsitektur *VGG-19*, khususnya pada Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan skripsi ini dikelompokkan menjadi lima bab, yaitu sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini dijelaskan terkait penyampaian masalah yang dikemas melalui latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan dalam penelitian.

BAB II : TINJAUAN PUSTAKA

Pada bab ini menguraikan rangkuman dasar teori yang dihimpun dari berbagai pustaka yang relevan dengan topik yang menjadi objek kajian dalam penelitian ini. Dasar teori yang dipakai dalam penelitian ini yaitu mengenai *Deep Learning*, *Convolutional Neural Network*, *Transfer Learning* dengan arsitektur *VGG-19* dan beberapa teori pendukung lainnya.

BAB III : METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan tentang tahapan-tahapan yang akan dilakukan dalam penelitian ini.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian serta pembahasan dari penelitian yang telah dilaksanakan. Dalam bab ini disajikan gambar, tabel, serta grafik dari hasil penelitian yang telah dilakukan.

BAB V : KESIMPULAN DAN SARAN

Bab ini merupakan kesimpulan dari keseluruhan pembahasan terutama hasil penelitian yang telah dijelaskan pada bab sebelumnya. Bab ini juga berisi saran untuk penelitian selanjutnya yang akan mengangkat tema yang sama dengan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Jamur

Jamur (fungi) berasal dari kata Yunani “fungi” yang artinya “berkembang biak”. Ciri umum jamur adalah hidup sebagai parasit atau saprofit, karena merupakan organisme tanpa klorofil (Kusumaningrum, 2018). Tubuh jamur terdiri dari benang-benang yang disebut hifa yang dapat membentuk jaringan bercabang (miselium) (Mahran, Hapsari, & Nugroho, 2020).

Jamur dibagi menjadi kingdom mereka sendiri, Kerajaan Jamur. Tubuh jamur mungkin berbeda dengan spesies jamur lainnya, yang ditunjukkan dengan perbedaan tudung (pileus), tangkai (batang), lamellae (insang), dan kelopak (borba). Perbedaan ukuran, warna dan bentuk jumbai dan batang merupakan ciri pembeda penting dari jenis jamur (Kusumaningrum, 2018).

Beberapa spesies jamur diketahui dapat dimakan bahkan mengandung zat yang bisa digunakan sebagai obat-obatan. Namun banyak juga jamur yang tidak dapat dimakan karena mengandung zat beracun (Hermawan & Wibowo, 2022). Jamur merupakan tanaman dari kingdom fungi dengan tubuh berdaging dan dapat dimakan, meski demikian terdapat beberapa spesies jamur yang tidak dapat dimakan dan memiliki ciri khas tersendiri (Hayami et al, 2022). Pada saat ini diperkirakan 140.000 spesies jamur diidentifikasi dan 2.000 diantaranya aman untuk dikonsumsi manusia. Dimana 138.000 species lain masih belum bisa ditentukan dapat dimakan atau tidak dan bahkan mengandung racun yang membahayakan tubuh (Aziz et al, 2022).

Dalam mengenali jamur yang bisa dikonsumsi dan yang tidak, bisa dikenali dengan memperhatikan ciri-ciri morfologi jamur seperti bentuk payung, warna, tekstur payung, dan ciri lain yang dapat dilihat (Haksoro & Setiawan, 2021). Namun, identifikasi jamur masih sulit dilakukan karena banyaknya jenis jamur, kurangnya pengetahuan tentang jamur, dan kurangnya ahli dalam bidang jamur. Selain itu, sebagian besar jamur memiliki tingkat kesamaan yang tinggi dalam karakteristik

tertentu, yang menyulitkan dalam mengidentifikasi jenis jamur secara visual oleh manusia yang tidak memiliki keahlian khusus.

2.2 Deep Learning

Selain Machine Learning terdapat teknik pengolahan citra lain, yaitu Deep Learning yang lebih baik dibandingkan Machine Learning (Shinta et al, 2023). Deep Learning merupakan bagian dari Machine Learning yang dapat memodelkan data kompleks seperti gambar maupun suara (Susanto et al, 2023). Perbedaan utama antara Machine Learning dengan Deep Learning terletak pada bagian feature extraction. Dalam Deep Learning, feature extraction sudah ada (Hindarto & Santoso, 2021) secara otomatis dan dapat melakukan ekstraksi pada fitur-fitur citra untuk melakukan klasifikasi, sedangkan pada Machine Learning perlu dibuat secara manual atau terpisah. Teknik Deep Learning memberikan hasil terbaik dalam pengenalan citra yakni metode Convolutional Neural Network (CNN) (Susanto et al, 2023). Oleh karena itu, Deep Learning sangat sesuai dijadikan sebagai solusi atas permasalahan yang berhubungan dengan klasifikasi citra.

CNN merupakan salah satu algoritma Deep Learning yang dapat diterapkan untuk klasifikasi citra. Dalam Deep Learning, metode CNN sangat baik dalam klasifikasi gambar dapat menghasilkan prediksi akurat dalam pengenalan gambar (Rahmadhani & Marpaung, 2023). Seperti halnya Deep Learning pada umumnya, metode CNN memiliki kelemahan pada proses pelatihan data, yaitu memakan waktu yang cukup lama dan dapat diatasi dengan menggunakan teknologi GPU pada perangkat komputasi (Susanto et al, 2023).

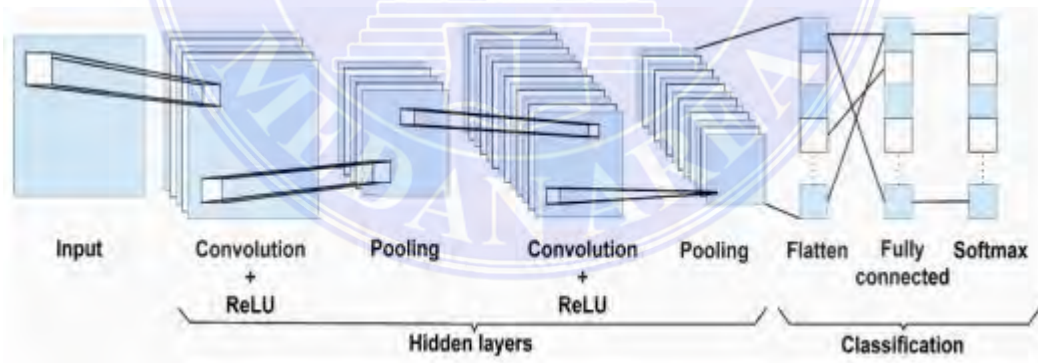
2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) dikenal juga sebagai ConvNet, merupakan pengembangan dari Multilayer Perceptron (MLP) yang digunakan untuk mengolah data 2D (dua dimensi) (Yohanes & Riva, 2022). CNN adalah kategori neural networks yang telah terbukti sangat efektif digunakan untuk pengenalan dan klasifikasi citra (Dhaniswara et al, 2021). CNN digunakan untuk mengklasifikasikan data yang terlabel dengan menggunakan metode supervised learning, yang mana cara

kerja dari supervised learning adalah terdapat data yang dilatih dan terdapat variabel yang ditargetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada (Kusumaningrum, 2018).

CNN saat ini dianggap sebagai model terbaik untuk menyelesaikan klasifikasi citra atau masalah yang berkaitan dengan klasifikasi citra (Sabri, 2022). CNN memiliki fungsi ekstraksi fitur. Suatu fungsi harus ditentukan untuk suatu proses atau tugas seperti klasifikasi, pengelompokan, atau regresi. Dalam pembelajaran mesin, adalah umum untuk melakukan ekstraksi fitur secara manual. Ini berarti bahwa fitur yang akan diekstraksi sudah ditentukan sebelumnya. Di sisi lain, CNN secara otomatis melakukan ekstraksi fitur dalam lapisan convolutional dan pooling dan juga mengaktifkan Rectified Linear Unit (ReLU). Selanjutnya, proses klasifikasi fitur dilakukan pada level konektivitas penuh dan aktivasi softmax (Setiawan, 2019).

CNN terdiri atas tiga lapis (layer) yaitu lapis masukan (input layer), lapis keluaran (output layer), dan beberapa lapis tersembunyi (hidden layers). Lapis tersembunyi (hidden layer) umumnya berisi convolutional layers, pooling layers, normalization layers, ReLU layer, fully connected layers, dan serta loss layer (Radana, 2022). Adapun arsitektur dari CNN dapat ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Arsitektur Convolutional Neural Network
Sumber : Ketwongsa et al, 2022

Secara umum tipe lapisan CNN pada Gambar 2.1 dibagi menjadi dua bagian, yaitu feature extraction layer (feature learning) dan classification layer. Pada bagian feature learning terdiri dari convolutional layer, fungsi aktivasi ReLU, dan pooling

layer. Sedangkan pada bagian classification layer terdiri dari flatten layer, fully connected layer, dan fungsi aktivasi Softmax.

2.3.1 Convolutional Layer

Convolutional layer merupakan lapisan inti CNN, pada lapisan ini sebagian besar proses komputasi dilakukan (Radana, 2022). Convolution layer melakukan operasi konvolusi pada output dari lapisan sebelumnya (Kusumaningrum, 2018). Tujuan utama konvolusi dalam kaitannya dengan ConvNet adalah untuk mengekstraksi fitur dari gambar yang dimasukkan (Radana, 2022).

Lapisan convolutional terdiri dari struktur dengan seperangkat filter berukuran tetap yang dapat menerapkan fungsi kompleks pada gambar yang disisipkan. Proses ini dilakukan dengan menggeser filter di atas gambar. Selama proses ini, setiap filter memiliki bobot dan nilai bias yang sama di seluruh gambar. Lapisan konvolusi menghasilkan gambar baru yang disebut peta fitur (feature map) (Radana, 2022). Operasi konvolusi pada citra dengan lebih dari satu dimensi, dapat digunakan rumus pada persamaan (2.1).

$$S_{(i,j)} = (I * K)_{(i,j)} = \sum_m \sum_n I_{(i-m,j-n)} K_{(m,n)} \tag{2.1}$$

Untuk lebih mudah memahami proses penghitungan konvolusi menggunakan filter 3 x 3 yang dilakukan dengan menggunakan persamaan (2.1) dapat diilustrasikan pada Gambar 2.2.



Gambar 2.2 Convolutional Layer

Sumber : Ketwongsa et al, 2022

Pada Gambar 2.2, citra berukuran 8x8 dikonvolusi dengan menggunakan kernel 3x3. Hasil keluaran dari proses konvolusi adalah citra baru dengan ukuran 6x6. Elemen pada citra hasil konvolusi dihitung dengan cara melakukan perkalian antara bobot kernel dengan nilai citra yang bersangkutan, dan jumlah dari perkalian tersebut dijadikan elemen pada citra hasil konvolusi. Untuk menjaga informasi tepi pada citra input, seringkali dilakukan penambahan zero padding pada sisi citra input.

2.3.2 Activation Function

Activation function atau fungsi aktivasi adalah merupakan node yang diletakkan di akhir atau di antara neural networks supaya hasil yang dikeluarkan lebih akurat lagi (Ayu et al, 2021). Fungsi aktivasi yang paling digunakan dalam metode CNN adalah fungsi aktivasi ReLU dan fungsi aktivasi Softmax.

1. Fungsi Rectified Linear Unit (ReLU)

ReLU merupakan fungsi aktivasi yang saat ini paling populer digunakan. ReLU pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila x merupakan bilangan negatif maka akan dirubah menjadi angka 0. Fungsi aktivasi ReLU (Anugerah, 2018) menggunakan persamaan (2.2).

$$f(x) = \max(0, x) \quad (2.2)$$

Fungsi aktivasi ReLU dibandingkan dengan fungsi Sigmoid atau fungsi aktivasi serupa, memungkinkan pelatihan yang lebih cepat dan efektif untuk arsitektur syaraf pada kumpulan data yang besar dan kompleks (Anugerah, 2018). ReLU memiliki keunggulan dalam mempercepat proses fitting stochastic gradient descent (SGD) dibandingkan dengan fungsi Sigmoid dan Tanh. Namun aktivasi ini juga memiliki kelemahan yaitu aktivasi ini bisa menjadi tricky dalam proses pelatihan dan menyebabkan unit mati. (Putri, 2020).

2. Fungsi aktivasi Softmax

Fungsi Softmax pada CNN biasanya digunakan pada fully connected layer untuk klasifikasi. Secara umum, fungsi ini akan menghitung probabilitas masing-masing kelas target di atas semua kelas sasaran yang mungkin ada. Kemudian probabilitas yang dihitung akan sangat membantu untuk menentukan

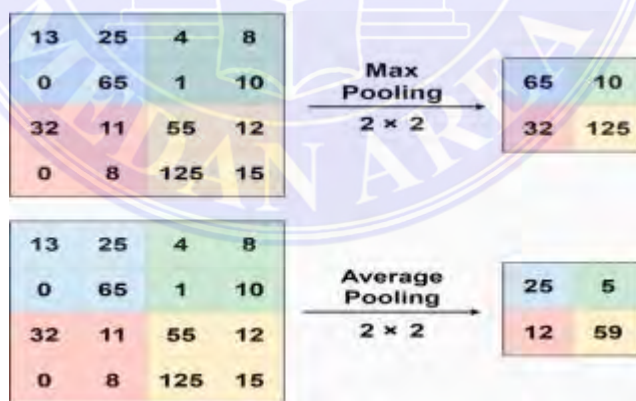
kelas target untuk input yang diberikan. Fungsi aktivasi Softmax (Anugerah, 2018) dengan menggunakan persamaan (2.3).

$$f(x) = \frac{e^{x_i}}{\sum_{j=0}^k e^{x_j}} \tag{2.3}$$

Keuntungan utama menggunakan Softmax adalah rentang probabilitas output. Rentang akan 0 sampai 1, dan jumlah semua probabilitas akan sama dengan satu. Jika fungsi softmax digunakan untuk model multi-klasifikasi, maka akan mengembalikan probabilitas masing-masing kelas dan kelas target akan memiliki probabilitas tinggi (Anugerah, 2018).

2.3.3 Pooling Layer

Pooling layer mengambil layer convolutional sebagai input. Proses pada pooling layer diterapkan ke feature maps yang telah melewati fungsi konvolusi dan aktivasi (Radana, 2022). Pooling layer berfungsi untuk menjaga ukuran data pada saat konvolusi, serta melakukan down-sampling, sehingga data dikembangkan menjadi bagian yang lebih kecil, mudah dikelola, dan mudah dalam mengatur overfitting (Yohannes et al, 2021). Adapun ilustrasi dari proses pooling layer dengan menggunakan max pooling dan average pooling dapat ditampilkan pada Gambar 2.3.



Gambar 2. 2 Pooling Layer dengan Max Pooling dan Average Pooling Sumber : Ketwongsa et al, 2022

Penggunaan pooling layer pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah convolution layer

dengan stride yang sama dengan pooling layer yang bersangkutan (Kusumaningrum, 2018).

2.3.4 Flatten Layer

Flatten layer merupakan langkah yang dilakukan sebelum masuk ke fully connected layer. Feature map dari tahap feature extractor masih berbentuk array multidimensi, sehingga perlu dilakukan proses flatten untuk mengubah ulang fitur (reshape feature map) menjadi vektor satu dimensi agar dapat digunakan sebagai input untuk tahap klasifikasi atau fully connected. Pada penelitian ini flatten layer akan diganti dengan Global Average Pooling. Konsep dari *Global Average Pooling* dapat ditunjukkan pada Gambar 2.4.



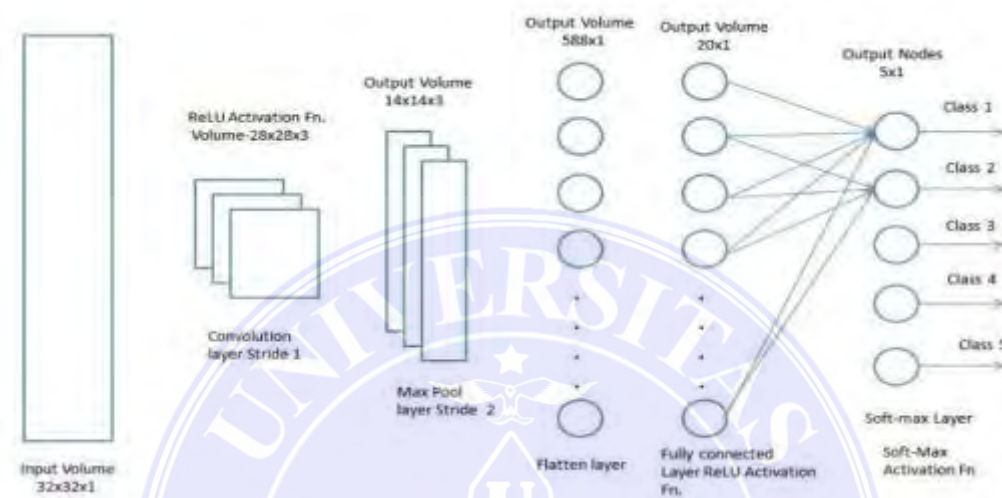
Gambar 2. 3 Global Average Pooling
Sumber : Syarif, 2021

Global Average Pooling merupakan operasi yang mereduksi suatu ukuran matriks menjadi matriks yang kecil. Contohnya matriks 4x4 diubah menjadi matriks 1x1 dengan cara mengambil nilai rata-rata pada setiap matriks tersebut seperti yang ditunjukkan oleh Gambar 2.4.

2.3.5 Fully-Connected Layer

Fully connected layer merupakan layer terakhir dari CNN yang biasa dipakai dalam penerapan Multilayer Perceptron (MLP) Layer ini bertujuan untuk melakukan tranformasi agar data dapat diklasifikasi (Yohannes & Rivan, 2022). Perbedaan antara fully connected layer dan convolution layer biasanya adalah neuron di

convolution layer terhubung hanya ke daerah tertentu pada input, sedangkan fully connected layer mempunyai neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda.



Gambar 2. 4 Fully Connected Layer
Sumber : Radana, 2022

Fully-Connected Layer terdiri dari input, hidden dan output layer. Disetiap hidden layer terdapat fungsi aktivasi, yang umum digunakan yaitu ReLU, dan begitu pula di output layer, juga terdapat fungsi aktivasi, dalam kasus klasifikasi (yang lebih dari 2 label/kelas/kategori) yang umum digunakan yaitu Softmax. Tujuan utama dari fully connected layer ialah mengolah data sehingga bisa dilakukan klasifikasi. output dari fully connected layer yaitu probabilitas terhadap kategori (jika menggunakan Softmax).

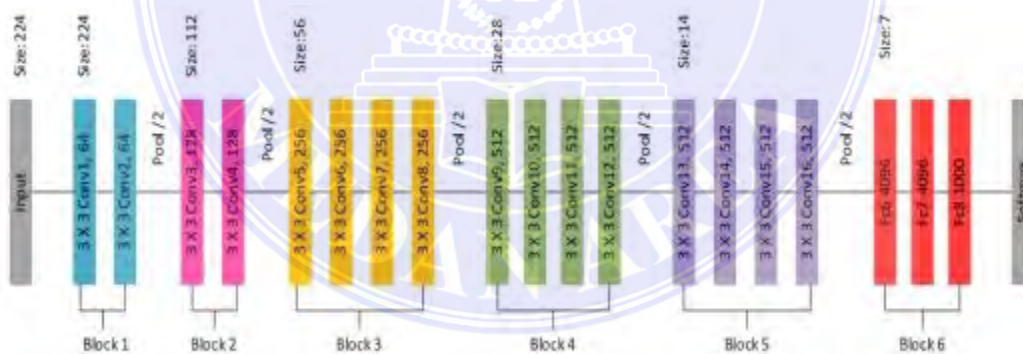
2.4 Arsitektur VGG-19

VGG-19 (Visual Geometry Group-19) adalah salah satu dari sekian banyak arsitektur Convolutional Neural Network (CNN) (Fendiawati & Rivan, 2023). VGG-19 adalah model CNN yang terkenal, dan telah berhasil diterapkan dalam beberapa klasifikasi gambar, pengenalan pola, dan pengenalan suara (Dhaniswara et al, 2021). VGG-19 dikemukakan oleh Simonyan dan Zisserman adalah model CNN yang terdiri

dari 19 lapisan dengan 16 convolution layers dan 3 fully connected untuk mengklasifikasikan gambar ke dalam 1000 kategori objek (Bansal et al, 2021).

VGG-19 merupakan perkembangan dari arsitektur VGG-16 yang memiliki 19 lapisan mendalam sehingga kinerja yang dihasilkan menjadi lebih baik. VGG-19 merupakan arsitektur CNN yang terdiri atas 16 lapisan convolutional layer beserta aktivasi ReLU dan 3 lapisan fully connected layer beserta aktivasi ReLU. Pada setiap lapisan terakhir convolutional terdapat lapisan max pooling untuk mengurangi ukuran spasial dari convolutional layer sebelumnya. Lapisan input menggunakan ukuran 224 x 224 piksel, dan lapisan output diganti dengan fungsi aktivasi softmax yang menunjukkan hasil klasifikasi (Shinta et al, 2023).

VGG-19 merupakan arsitektur yang pernah digunakan untuk melatih lebih dari 1 juta gambar yang didapatkan dari database ImageNet. Selain itu, arsitektur ini memiliki kernel yang berukuran 3x3 dan memiliki 5 blok dengan berbagai ukuran convolutional layer pada setiap blok, yang kemudian ditambahkan max pooling layer sebagai pemisah setiap blok (Marcella et al, 2022). Berikut ini merupakan arsitektur VGG-19 yang yang dapat ditampilkan pada Gambar 2.6.



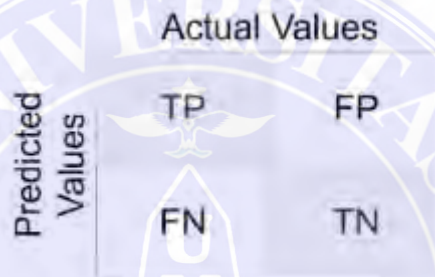
Gambar 2. 5 Arsitektur VGG-19
Sumber : Shinta et al, 2023

2.5 Confusion Matrix

Dalam menentukan baik tidaknya performa dari suatu model klasifikasi yang sudah dilatih (training) dapat dilihat dari pengukuran performa parameternya, yaitu accuracy, recall, precision, dan f1-score. Untuk menghitung faktor-faktor tersebut

diperlukan sebuah matriks yang biasa disebut dengan *confusion matrix*. *Confusion matrix* adalah suatu metode untuk menghitung performansi dari masalah klasifikasi (Rahmadhani & Marpaung, 2023).

Confusion matrix merupakan matriks berukuran N x N dengan N adalah jumlah kelas target yang akan dievaluasi (Shinta et al, 2023). Metode confusion matrix digunakan untuk menghitung *accuracy*, *recall*, *precision*, dan *f1-score* (Marcella et al, 2022). Cara kerja *Confusion matrix* yaitu melakukan perbandingan antara nilai prediksi dengan nilai sebenarnya (Shinta et al, 2023). *Model confusion matrix* untuk kasus klasifikasi dua kelas (binary classification) dapat dilihat pada Gambar 2.1.



Gambar 2. 6 Model Confusion Matrix

Sumber : Ketwongsa et al, 2022

Confusion matrix memiliki 4 istilah sebagai representasi hasil proses klasifikasi yaitu, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negatif* (FN) (Marcella et al, 2022). Berikut nilai untuk mengevaluasi model klasifikasi (Shinta et al, 2023), yaitu sebagai berikut:

1. *Accuracy*, merupakan *matrix* yang digunakan untuk menghitung perbandingan keseluruhan prediksi benar terhadap keseluruhan nilai kelas. Nilai *accuracy* dapat dihitung dengan menggunakan persamaan (2.4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

2. *Precision*, merupakan *matrix* yang digunakan untuk menghitung presisi atau data prediksi positif dari keseluruhan hasil yang diprediksi positif. Nilai *precision* dapat dihitung dengan menggunakan persamaan (2.5).

$$Precision = \frac{TP}{TP + FP} \quad (2.5)$$

3. *Recall*, merupakan *matrix* yang digunakan untuk menghitung jumlah prediksi pada kelas yang sama. Nilai *recall* dapat dihitung dengan menggunakan persamaan (2.6).

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

4. *F1-Score*, merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Rentang pembobotan *F1-score* adalah 0 sampai 1 dengan 0 adalah nilai terburuk dan 1 adalah nilai terbaik. Nilai *F1-score* dapat dihitung dengan menggunakan persamaan (2.7).

$$F1\ Score = 2 \left(\frac{Precision * Recall}{Precision + Recall} \right) \tag{2.7}$$

Keterangan:

TP : jumlah data aktual yang sebenarnya benar dan diprediksi benar

TN : jumlah data aktual yang sebenarnya salah dan diprediksi salah

FP : jumlah data aktual yang sebenarnya benar dan diprediksi salah

FN : jumlah data aktual yang sebenarnya salah dan diprediksi benar

2.6 Penelitian Terkait

Terkait penelitian yang dilakukan, maka referensi dari penelitian-penelitian sebelumnya penting dilakukan agar penelitian dengan tema yang sama akan semakin berkembang dengan kontribusi baru yang dikerjakan pada penelitian ini. Berikut ulasan mengenai hasil penelitian yang pernah dilakukan sebelumnya yang berkenaan dengan klasifikasi jamur sebagai objek penelitian.

Tabel 2. 1 Penelitian Terkait

No.	Nama Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian
1.	Ruth Hanseliani, 2019	Klasifikasi Berbagai Jenis Jamur Layak Konsumsi dengan Metode <i>Backpropagation</i>	Metode <i>Backpropagation</i> mampu mengklasifikasikan 6 jenis jamur layak konsumsi

			dengan akurasi terbaik sebesar 97%.
2.	Yohannes, Nur Rachmat dan Calvin Oliver Saputra, 2021	Penggunaan Fitur <i>HOG</i> Berbasis <i>Supapixel</i> Untuk Klasifikasi Jenis Jamur Dengan Metode <i>SVM</i>	Penerapan fitur <i>HOG</i> berbasis superpixel dapat meningkatkan akurasi metode <i>SVM</i> dalam klasifikasi 9 jenis jamur dibandingkan tanpa <i>supapixel</i> dengan akurasi 83.29%
3.	Yohannes, Daniel Udjulawa, Timoteus Ivan Sariyo, 2022	Klasifikasi Jenis Jamur Menggunakan <i>SVM</i> dengan Fitur <i>HSV</i> dan <i>HOG</i>	Metode <i>SVM</i> mampu mengklasifikasikan 9 jenis jamur menggunakan fitur <i>HSV</i> dan <i>HOG</i> dengan accuracy 82,69%
4.	Frencis Matheos Sarimole dan Randitia Ridad Diadi, 2022	Klasifikasi Jenis Jamur Menggunakan Ekstraksi Fitur <i>GLCM</i> dan <i>K-Nearest Neighbor</i> (KNN)	Metode <i>KNN</i> dengan ekstraksi fitur <i>GLCM</i> dapat mengklasifikasi 5 jenis jamur dengan akurasi 77%
5.	Nuke L. Chusna, Mohammad Imam Shalahudin, Umbar Riyanto, dan Allan Desi Alexander,	Klasifikasi Citra Jenis Tanaman Jamur Layak Konsumsi Menggunakan Algoritma <i>Multiclass Support Vector Machine</i>	Hasil pengujian diperoleh akurasi rata-rata sebesar 83% dalam klasifikasi 5 jenis jamur layak konsumsi menggunakan algoritma <i>Multiclass SVM</i> dengan akstraksi ciri orde pertama

	2022		
6.	Ummi Sri Rahmadhani dan Noveri Lysbetti Marpaung, 2023	Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode <i>CNN</i>	Model pada penelitian menggunakan tiga convolution layer, tiga MaxPooling layer dan dua dropout layer. Pada tahap pengujian model diperoleh akurasi sebesar 76% dalam klasifikasi 6 jenis jamur

Berdasarkan Tabel 2.1, setelah mempelajari beberapa hasil penelitian terdahulu yang berkaitan dengan metode dan objek yang digunakan pada penelitian ini, maka dapat diketahui bahwa perbedaan penelitian ini dengan penelitian sebelumnya terletak pada jumlah dataset serta kelas atau jenis jamur yang akan diklasifikasi serta metode yang diterapkan. Dari 5 penelitian sebelumnya, terdapat 4 penelitian yang menerapkan metode *SVM* dan *KNN*, 1 penelitian menggunakan metode *ANN* Backpropagation dan 1 lagi menggunakan metode *CNN* kustom. Sedangkan pada penelitian yang akan dilakukan menerapkan metode Deep Learning dengan pendekatan *transfer learning* arsitektur *VGG-19* dalam melakukan pelatihan model untuk klasifikasi jenis jamur.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan sejak dikeluarkannya izin penelitian pada bulan Mei 2023. Tempat yang digunakan dalam melakukan penelitian ini adalah pada laboratorium Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.

3.2 Alat dan Bahan Penelitian

Alat yang digunakan dalam penelitian ini adalah satu buah unit laptop yang dilengkapi dengan perangkat keras (hardware) dan perangkat lunak (software). Perangkat keras komputer yang digunakan adalah perangkat keras yang dapat mendukung perangkat lunak yang memiliki kemampuan atau tampilan grafis yang cukup baik. Adapun deskripsi spesifikasi kebutuhan perangkat keras (hardware) yang digunakan dalam penelitian ini ditunjukkan pada Tabel 3.1.

Tabel 3.1 Deskripsi Kebutuhan Perangkat Keras

No.	Nama Perangkat	Deskripsi
1.	Laptop	Asus VivoBook Max
2.	Processor	Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 1.99 GHz
3.	RAM	8.00 GB
4.	Hard Disk	1 TB

Adapun deskripsi spesifikasi kebutuhan perangkat lunak (software) yang digunakan dalam penelitian ini dapat disajikan pada Tabel 3.2.

Tabel 3.2 Deskripsi Kebutuhan Perangkat Lunak

No.	Nama	Deskripsi
1.	Windows 10 Pro 64-bit	Operating System
2.	Python 3.10.11 dan Google Colab Pro	Tools untuk membangun dan melatih (training) model
3.	Visual Studio Code	Tools untuk coding dan eksekusi program dari model yang akan di deploy kedalam aplikasi berbasis web
4.	Google Chrome	Web browser untuk menampilkan aplikasi
5.	Advanced Renamer	Tools untuk labeling atau rename dataset
6.	Microsoft Office 2010	Tools untuk membuat laporan penelitian

Bahan yang digunakan dalam penelitian ini adalah dataset berupa citra atau gambar jamur dari jenis agaricus, amanita, boletus, cortinarius, entoloma, hygrocyebe, lactarius, russula, dan suillus. Bahan pendukung lainnya yang digunakan dalam penelitian ini mencakup literatur dari berbagai sumber seperti buku, skripsi, jurnal, dan sumber-sumber ilmiah lainnya yang diperoleh dari berbagai situs website.

3.3 Jenis dan Sumber Data

Data yang akan digunakan dalam penelitian ini adalah merupakan jenis data sekunder yang diperoleh secara tidak langsung atau dari sumber yang sudah ada. Pada penelitian ini, data sekunder yang digunakan berupa citra atau gambar jamur diperoleh dari halaman website Kaggle (www.kaggle.com) yang diakses pada bulan Mei 2023. Dari website tersebut terdapat 9 kategori jenis jamur yang digunakan dalam penelitian ini, yaitu jenis agaricus, amanita, boletus, cortinarius, entoloma, hygrocyebe, lactarius, russula, dan suillus.

3.4 Sampel Penelitian

Populasi yang digunakan dalam penelitian ini adalah citra atau gambar dari 9 jenis jamur. Sedangkan sampel yang digunakan dalam penelitian ini berjumlah 4500

buah citra, terdiri dari 500 buah citra untuk setiap jenis jamur. Adapun sampel dari data yang digunakan dalam penelitian ini dapat ditampilkan pada Gambar 3.1.



Gambar 3.1 Sampel Dataset

Gambar 3.1 merupakan visualisasi dari sampel data yang akan digunakan dalam melakukan klasifikasi jenis jamur menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur Visual Geometry Group (VGG-19) yang merupakan salah satu metode dalam Deep Learning yang terkenal mampu mengklasifikasi data citra atau gambar dengan baik.

3.5 Metode Analisis Data

Metode analisis data dalam penelitian ini menggunakan metode CNN dengan menerapkan arsitektur VGG-19 untuk melakukan klasifikasi terhadap data citra atau gambar jamur. Penelitian ini dilakukan untuk membangun beberapa skenario model CNN dengan arsitektur VGG-19 untuk mengetahui model mana yang melakukan

pengklasifikasian dengan tingkat akurasi terbaik pada citra jamur. Proses analisis data dalam penelitian ini menggunakan tools Google Colab Pro dengan bantuan Keras dan Tensorflow sebagai library utama dalam membangun dan melatih model. Adapun tools pendukung lainnya yang digunakan dalam penelitian ini dapat dilihat pada bagian sub bab sebelumnya.

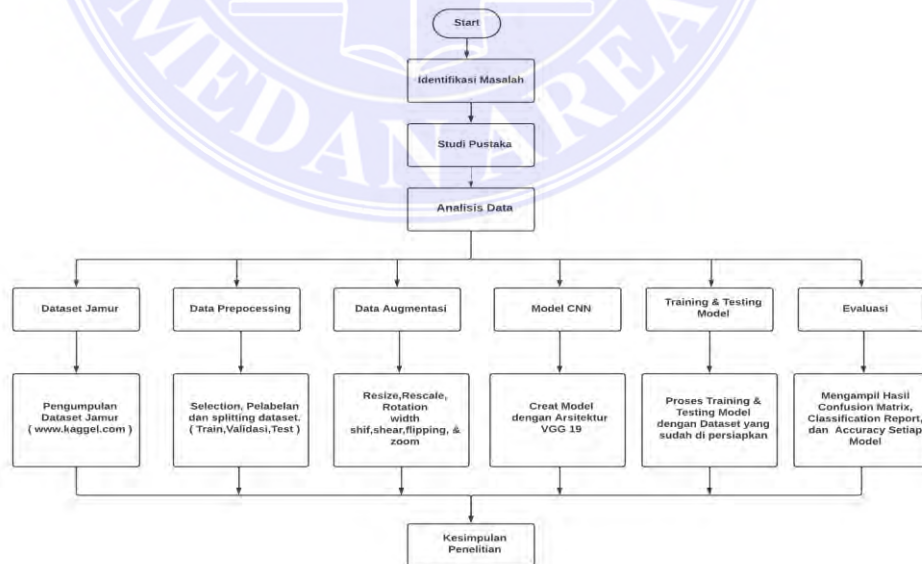
3.6 Initialization Hyperparameter

Tabel 3.3 Initialization Hyperparameter

Parameter	Value
Epoch	{ (20,50) }
Batch Size	{ (32),(64) }
Learning Rate	{ (0.001), (005) }
Optimizer	{ (Adaptive Moment Estimation),(Root Mean Square Propagation) }

3.7 Tahapan Penelitian

Tahapan atau langkah-langkah dalam penelitian ini digambarkan dalam bentuk diagram alir pada Gambar 3.2.



Gambar 3. 2 Tahapan Penelitian

Berdasarkan diagram alir penelitian pada Gambar 3.4 terdapat beberapa tahapan yang akan dilakukan dalam penelitian klasifikasi jenis jamur. Adapun penjelasan dari tiap tahapan yang dilakukan dapat diuraikan sebagai berikut:

1. Identifikasi dan Perumusan Masalah

Tahapan pertama yang dilakukan adalah mengidentifikasi pokok permasalahan yang akan diangkat dalam penelitian ini. Dalam hal ini, penelitian yang dilakukan didasarkan pada kurangnya pengetahuan tentang klasifikasi jamur berdasarkan jenisnya. Selanjutnya dilakukan perumusan terhadap masalah yang dihadapi untuk menentukan tujuan penelitian yang diharapkan mampu menjawab setiap permasalahan yang ada dari topik penelitian yang diangkat. Setelah itu, dibuat batasan masalah dan manfaat dari dilakukannya penelitian ini.

2. Studi Pustaka

Pada tahap ini dilakukan studi pustaka yang berhubungan dengan topik dan pokok permasalahan dalam penelitian yang dilakukan. Studi pustaka dilakukan dengan mencari berbagai sumber-sumber pendukung, seperti buku, skripsi, jurnal, dan sumber-sumber ilmiah lainnya yang diperoleh dari berbagai situs website. Dalam hal ini, studi pustaka meliputi jamur, Deep Learning, Convolutional Neural Network, arsitektur VGG-19 dan literatur terkait sebagai acuan dalam melakukan penelitian ini.

3. Analisis Data

Pada tahap selanjutnya dilakukan analisis data yang merupakan inti dari tahapan dalam proses klasifikasi jenis jamur. Dalam hal ini, terdapat beberapa langkah yang akan dilakukan, yaitu sebagai berikut:

a. Dataset Jamur

Pada tahap ini dilakukan pengumpulan dataset, yaitu berupa citra atau gambar jamur. Pengumpulan dataset diperoleh melalui situs Kaggle (www.kaggle.com) yang merupakan database yang tersedia untuk umum. Dataset terdiri dari 9 jenis jamur dengan jumlah dataset sebanyak 4500 buah citra (500 citra setiap jenis jamur).

b. Data Preprocessing

Dataset yang telah dikumpulkan selanjutnya akan melewati tahapan data preprocessing dengan melakukan selection, pelabelan, dan splitting dataset. Pada proses selection, dataset akan diseleksi terlebih dahulu, untuk citra yang berkualitas rendah dan terdapat noise akan dihapus. Setelah melakukan tahapan seleksi pada *dataset*, maka semua citra pada dataset akan dinamai ulang (*labeling*) sesuai dengan nama jenis jamur menggunakan pola yang telah ditentukan. Pada proses terakhir dari data preprocessing akan dilakukan splitting atau pembagian dataset, yaitu data train (data yang digunakan untuk proses training), data val (data yang digunakan untuk proses validasi), dan data test (data yang digunakan untuk proses testing atau pengujian).

c. Data Augmentation

Dataset yang telah melewati tahapan data preprocessing, selanjutnya dilakukan proses augmentasi yang bertujuan untuk meningkatkan keberagaman pada dataset. Augmentasi data dilakukan dengan cara resize, rescale, pemutaran (rotation), pergeseran pada sudut citra (width shift dan height shift), memiringkan posisi citra (shear), memperbesar (zoom) pada citra, dan pencerminan (flipping) secara horizontal.

d. Modeling CNN

Pada tahap ini dilakukan perancangan model CNN dengan arsitektur VGG-19 yang dimana menerapkan metode transfer learning dalam pembuatan model yang siap pakai (pre-trained model) untuk melakukan klasifikasi jenis jamur. Terdapat sejumlah skenario model yang akan di uji pada penelitian ini, dimana nantinya hasil dari masing-masing model akan di bandingkan dengan kondisi aktual. Dengan melihat keakurasian klasifikasi, maka model tersebut yang nantinya akan digunakan dalam klasifikasi jenis jamur.

e. Training dan Testing Model

Pada tahap ini dilakukan proses pembelajaran untuk melatih skenario model menggunakan data latih (data training). Pengujian model menggunakan perbandingan hyperparameter berupa batch size (32 dan 64), jumlah epoch

(20 dan 50), optimizer (Adam, RMSprop, SGD, dan Adadelta) serta learning rate (0.001 dan 0.005). Sedangkan parameter dari input shape (224,224,3) akan dijadikan statis pada setiap skenario model yang diuji.

f. Evaluation

Mengevaluasi kinerja model menggunakan data uji yang independen. Mengukur metrik evaluasi seperti akurasi, precision, recall, dan f1-score dengan menggunakan confusion matrix, untuk mengevaluasi seberapa baik model dalam mengklasifikasi jenis jamur. Model yang dihasilkan akan diuji menggunakan data testing untuk mengevaluasi hasilnya lalu memilih model terbaik dengan akurasi yang paling tinggi, dan tahap yang terakhir adalah men-deploy model kedalam aplikasi siap pakai.

4. Kesimpulan Penelitian

Tahap akhir adalah dengan membuat kesimpulan atas keseluruhan rangkaian penelitian yang telah dilakukan, dan kesimpulan hasil yang diperoleh sesuai dengan tujuan dan rumusan masalah yang telah ditentukan diawal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil analisis dalam melakukan klasifikasi jenis jamur dengan menggunakan arsitektur *VGG-19* adalah sebagai berikut:

1. Dari 18 skenario model yang di training, diperoleh 4 model dengan akurasi terbaik yaitu model skenario 15 menggunakan *hyperparameter* jumlah *epoch* 50, *batch size* 64, dan *optimizer SGD* mendapat akurasi 77.3%. Model skenario 16 menggunakan *hyperparameter* jumlah *epoch* 50, *batch size* 64, dan *optimizer Adadelta* mendapat akurasi 77.7%. Model skenario 17 menggunakan *hyperparameter* jumlah *epoch* 50, *batch size* 64, dan *optimizer Adadelta*, dan learning rate 0.001 mendapat akurasi 77.3%. Model skenario 18 menggunakan *hyperparameter* jumlah *epoch* 50, *batch size* 64, dan *optimizer Adadelta*, dan learning rate 0.005 mendapat akurasi 77.4%.
2. Dari 4 skenario model yang di uji menggunakan data testing dan di evaluasi menggunakan confusion matrix dan classification report, maka performa model yang paling optimal adalah model skenario 15 dengan score accuracy sebesar 70%.

5.2 Saran

Saran yang dapat penulis berikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menggunakan dataset yang diperluas dengan jumlah yang lebih banyak, sehingga dapat menghasilkan performa model yang lebih baik, atau bisa dengan menggunakan data primer untuk pengujiannya.
2. Penelitian selanjutnya diharapkan bisa menggunakan metode penilaian yang lain seperti K-Fold Cross Validation, sehingga bisa digunakan sebagai acuan penilaian performa model dari sudut pandang yang lain.

3. Penelitian selanjutnya dapat mengembangkan (*deploy*) *model* kedalam aplikasi berbasis *mobile* dan proses klasifikasi dapat dilakukan secara *real time*.



Daftar Pustaka

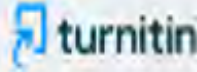
- Akbar, H., Aryani, D., & Ulum, M. B. (2022). Deteksi Banjir Area Perkotaan Berbasis Citra Digital dan Convolution Neural Network (VGG19).
- Al Aziz, M. R., Furqon, M. T., & Muflikhah, L. (2022). Klasifikasi Jamur Dapat Dimakan atau Beracun Menggunakan Naïve Bayes dan Seleksi Fitur berbasis Association Rule Mining. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Amelia, Y., Eosina, P., & Setiawan, F. A. (2018). Perbandingan Metode Deep Learning dan Machine Learning Untuk Klasifikasi (Uji Coba pada Data Penyakit Kanker Payudara). *Seminar Nasional Teknologi Informasi Universitas Ibn Khaldun Bogor* .
- Anggara, M. P. (2020). Transfer Learning Dengan Model Arsitektur VGG16 Dan Inception-v3 Untuk Klasifikasi Penyakit Daun Kentang. *skripsi*.
- Anggraini, W. (2018). KEANEKARAGAMAN HAYATI DALAM MENUNJANG PEREKONOMIAN MASYARAKAT KABUPATEN OKU TIMUR. *Aktual STIE Trisna Negara*.
- Angraini, W. (2018). Keanekaragaman Hayati Dalam Menunjang Perekonomian Masyarakat Kabupaten OKU Timur. *Aktual STIE Trisna Negara* .
- Anugerah, A. G. (2018). Klasifikasi Tingkat Keganasan Kanker Paru-Paru Pada Citra Computed Tomography (CT) Scan Menggunakan Metode Convolution Neural Network. *Institut Teknologi Sepuluh November*.
- Ayu, T., & Dwi, V. (2021). Pendiagnosa Daun Mangga Dengan Model Convolution Neural Network. *CESS (Journal of Computer Engineering System and Science)*.
- Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*.
- Bunyamin, H., & Pangestu, M. A. (2018). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model . *Teknik Informatika dan Sistem Informasi*.
- Darsilowati, Riadi, A. A., & Evanita. (2021). Klasifikasi Jenis Jamur Konsumsi Berbasis Android Menggunakan Metode K-Nearest Neighbors (KNN). *JAST : Journal of Applied Science and Technology*.

- Dhaniswara, E., Kristian, Y., & Setiawan, E. I. (2021). Detection of Banana and Its Ripeness Using Residual Neural Network. *JITE (Journal of Informatics and Telecommunication Engineering)*.
- Fendiawati, A., Rivan, M. A., & O. (2023). KLASIFIKASI AMERICAN SIGN LANGUAGE DENGAN METODE VGG-19. *MDP STUDENT CONFERENCE (MSC)*.
- Haksoro, E. I., & Setiawan, A. (2021). Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolution Neural Network. *Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*.
- Hanseliani, R., & Adi, C. K. (2019). Klasifikasi Berbagai Jenis Jamur Layak Konsumsi dengan Metode Backpropagation. *Media Informasi Analisa dan Sistem*.
- Hawksworth, D., & Rossman, A. (1997). where are all the undescribed fungi? *phytopathology*, 87:888-891.
- Hayami, R., Soni, & Gunawan, I. (2022). Klasifikasi Jamur Menggunakan Algoritma Naive Bayes. *Jurnal Computer Science and Information Technology (CoSciTech)*.
- Hermawan, A., Wibowo, A. P., & Hermawan, A. (2022). Implementasi Korelasi Untuk Seleksi Fitur pada Klasifikasi Jamur Beracun Menggunakan Jaringan Syaraf Tiruan. *INTEK*.
- Hindarto, D., & Santoso, H. (2021). Plat Nomor Kendaraan dengan Convolution Neural Network. *Jurnal Inovasi Informatika Universitas Pradita*.
- Ketwongsa, W., Boonlue, S., & Kokaew, U. (2022). A New Deep Learning Model for the Classification of Poisonous and Edible Mushrooms Based on Improved AlexNet Convolutional Neural Network.
- Kusumaningrum, T. F. (2018). Implementasi Convolution Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi di Indonesia Menggunakan Keras. *TUGAS AKHIR*.
- L.Chusna, N., Shalahudin, M. I., Riyanto, U., & Alexander, A. D. (2022). Klasifikasi Citra Jenis Tanaman Jamur Layak Konsumsi Menggunakan Algoritma Multiclass Support Vector Machine. *Building of Informatics, Technology and Science (BITS)*.
- Mahrani, A. A., Hapsari, R. K., & Nugroho, H. (2020). Penerapan Naive Bayes Gaussian Pada Klasifikasi Jenis Jamur Berdasarkan Ciri Statistik Orde Pertama. *Jurnal Ilmiah Nero*.

- Marcella, D., Yohannes, & Devella, S. (2022). Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19 . *Jurnal Algoritme*.
- Putri, N. O. (2020). Implementasi Metode CNN Dalam Klasifikasi Gamber Jamur Pada Analisis Image Processing . *TUGAS AKHIR* .
- Radana , J. D. (2022). Implementasi Deep Learning Menggunakan Algoritma Convolution Neural Network (CNN) Untuk Image Classification Varietas Tanaman Tembakau . *SKRIPSI*.
- Rahmadhani, U. S., & Marpaung, N. L. (2023). Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN. *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*.
- Sabri, A. (2022). Analisis Perbandingan Model Arsitektur CNN Dalam Pendeteksian Corona Virus Disease Menggunakan Citra X-ray Paru-Paru. *SKRIPSI*.
- Saputro, A., Mu'min, S., Lutfi, M., & Putri, H. (2022). DEEP TRANSFER LEARNING DENGAN MODEL ARSITEKTUR VGG16 UNTUK KLASIFIKASI JENIS VARIETAS TANAMAN LENGKENG BERDASARKAN CITRA DAUN. *Jurnal Mahasiswa Teknik Informatika* .
- Sarimole , F. M., & Diadi, R. R. (2022). Klasifikasi Jenis Jamur Menggunakan Ekraksi Fitur GLCM Dan K-Nearest Neighbor (KNN). *JINTEKS (Jurnal Informatika Teknologi dan Sains)*.
- Setiawan, W. (2019). PERBANDINGAN ARSITEKTUR CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI FUNDUS. *Jurnal Semantec*.
- Shinta, R., Jasril , Irsyad , M., Yanto, F., & Sanjaya, S. (2023). Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19. *Jurnal Sains dan Informatika* .
- Suryani, Y. (2020). *MIKOLOGI*. Bandung.
- Susanto, L. A., Nilogiri, A., & Handayani, L. (2023). Klasifikasi Citra Lesi Kulit Serupa Virus Monkeypox Menggunakan VGG-19 Convolutional Neural Network. *Jurnal Sistem dan Teknologi Informasi Indonesia*.
- Suwarso, E. (2019). KAJIAN DATABASE KEANEKARAGAMAN HAYATI KOTA SEMARANG. *RIPTEK*.

- Suwarso, E., Paulus, D. R., & Widanirmala, M. (2019). KAJIAN DATABASE KEANEKARAGAMAN HAYATI KOTA SEMARANG. *RIPTEK*.
- Syarif, A. K. (2021). Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan Metode Deep Learning Dengan Library Tensorflow Lite . *Skripsi*.
- Tanuwijaya, E., & Roseanne, A. (2021). Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia. *Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*.
- Yohannes , R., & Al Rivan4, M. E. (2026). Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM. *Jurnal Algoritme*.
- Yohannes , Rachmat, N., & Saputra, C. O. (2021). Penggunaan Fitur HOG Berbasis Superpixel Untuk Klasifikasi Jenis Jamur Dengan Metode SVM. *Jusikom :Jurnal Sistem Komputer Musirawas*.
- Yohannes, Udjulawa, D., & Sariyo, T. I. (2022). Klasifikasi Jenis Jamur Menggunakan SVM dengan Fitur HSV dan HOG. *Jurnal Pengkajian dan Penerapan Teknik Informatika* .
- Zubair , A., & Muslikh, A. R. (2018). Identifikasi Jamur Menggunakan Metode K-Nearest Neighbor Dengan Ekstraksi Ciri Morfologi.

Lampiran Turnitin



Similarity Report ID: oid:29477:44901107

PAPER NAME	AUTHOR
Rahmad Syuhada (198160001) oke - turnitin.docx	Rahmat Syuhada

WORD COUNT	CHARACTER COUNT
11622 Words	71306 Characters
PAGE COUNT	FILE SIZE
86 Pages	13.5MB
SUBMISSION DATE	REPORT DATE
Oct 16, 2023 12:10 AM GMT+7	Oct 16, 2023 12:11 AM GMT+7

22% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.

- 19% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 12% Submitted Works database

Excluded from Similarity Report

- Small Matches (Less than 10 words)

Lampiran SK Pembimbing

**UNIVERSITAS MEDAN AREA**
FAKULTAS TEKNIK

Kampus I : Jalan Kualanaram 1 Medan SumateraJalan PDSI Nomor 1 ☎ (061) 7366078, 7360168, 7364348, 7366781, Fax:(061) 7360008 Medan 20223
Kampus II : Jalan Seiutuadi Nomor 79 / Jalan Sei Serayu Nomor 75 A ☎ (061) 8225602, Fax, (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medan@uma.ac.id

Nomor : 66/FT.6/01.10/II/2023 2 Februari 2023
Lamp : -
Hal : Pembimbing Tugas Akhir

Yth. Pembimbing Tugas Akhir
Muhathir, ST, M. Kom
di
Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas

Nama : Rahmad Syuhada
N P M : 198160001
Jurusan : Teknik Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

Muhathir, ST, M. Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

"Analisis Arsitektur *Deep Learning* VGG untuk Klasifikasi Jenis Jamur".

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.


Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,



Syah, S. Kom, M.Kom

Lampiran Surat Pengantar Riset

 **UNIVERSITAS MEDAN AREA**
FAKULTAS TEKNIK

Kampus 1 : Jalan Pektan Nomor 1 Medan Estate/Jalan PGG Nomor 1 ☎ (061) 7360178, 7360160, 7364348, 7366781, Fax (061) 7366998 Medan 20223
Kampus 2 : Jalan Doksibus Nomor 70 / Jalan Gad Serayu Nomor 70 A, ☎ (061) 8229602, Fax (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id • E-mail: uriv_medan@uma.ac.id

Nomor : 253 /FT.6/01.10/III/2023
Lamp : 4
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

31 Maret 2023

Yth. Pimpinan CV. Sae Akademi Digital
Komplek Puri Zahara II F.12, Jln. Rinte Raya
Di
Medan

Dengan hormat,
Kami mohon kesediaan Bapak/Ibu berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :


NO	NAMA	NPM	PRODI
1	Rahmad Syuhada	198160001	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir pada perusahaan/Instansi yang Bapak/Ibu Pimpin.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :


Analisis Arsitektur Deep Learning VGG untuk Klasifikasi Jenis Jamur

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.


Dr. Raihan Syah, S. Kom, M. Kom

Tembusan :
1. Ka. BAMA
2. Mahasiswa
3. File

Lampiran Surat Selesai Riset

 **SAE DIGITAL AKADEMI**
Pusat Prakerin Bidang Digital & Entrepreneur
Alamat: Komplek Puri Zahara 2 Blok M No.6, Jalan Dunga Rote Raya
Simpang Selayang, Kec. Medan Tunjung, Kota Medan, Sumatera Utara 20133
www.tempatpkimedan.com | Instagram @tempatpkimedan | 0813-7017-7719

Nomor : 050/SAE/PKL/V/2023
Hal : Surat Balasan Penelitian dan Pengambilan Data Tugas Akhir

Kepada Yth :
Bapak/Ibu
Dekan,
Dr. Rahmad Syah, S.Kom, M.Kom


Assalamu'alaikum warahmatullahi wabarakatuh
Dengan Hormat, berdasarkan Surat Nomor : 253/FT.6/01.10/11/20223 Tanggal 31 Maret 2023 Perihal Penelitian Dan Pengambilan Data Tugas Akhir :

No	Nama	NPM	Prodi
1	Rahmad Syuhada	198160001	Teknik Informatika

Benar telah melaksanakan dan menyelesaikan Penelitian dan Pengambilan Data dengan judul :
" Analisis Arsitektur Deep Learning VGG untuk Klasifikasi Jenis Jamur " di SAE Digital Akademi Mitra UMKM Jamur Tiram Hartopo.

Demikian Surat Keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Wassalamu'alaikum
Medan, 15 Mei 2023


(Sugianto, S.T., M.Kom)

Lampiran Source Code

Source Code Google Colaboratory

```
# import Library untuk data handling

import os

import numpy as np #linear algebra

import pandas as pd #data processing

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

# import Library Deep learning

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras import Model

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, GlobalAveragePooling2D, MaxPooling2D, Flatten, Dense, Activation, Dropout

from tensorflow.keras.optimizers import Adam, RMSprop, SGD, Adadelta

from tensorflow.keras.models import load_model

from tensorflow.keras.metrics import categorical_crossentropy

from tensorflow.keras.preprocessing import image_dataset_from_directory

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import regularizers

from tensorflow.keras.callbacks import EarlyStopping
```



```
from tensorflow.keras import metrics

import gc # Gabage collector for cleaning deleted data from memory (in case : Clear list
that are useless)

# Ignore Warnings

import warnings

warnings.filterwarnings("ignore")

print ('modules loaded')
```



Praproses Dataset

```
# Mount Google Drive

from google.colab import drive

drive.mount('/content/drive/')

# persiapan direktori dataset (data training, validation, dan testing)

train_dir = '/content/drive/MyDrive/My Dataset/mushrooms/train'

val_dir = '/content/drive/MyDrive/My Dataset/mushrooms/val'

test_dir = '/content/drive/MyDrive/My Dataset/mushrooms/test'

# menampilkan jumlah dataset

data_train = image_dataset_from_directory(train_dir)

data_val = image_dataset_from_directory(val_dir)

data_test = image_dataset_from_directory(test_dir)

# Training & Validation generator + Augmentation

train_datagen = ImageDataGenerator(rescale=1./255,          # Scaling pixel value
                                   rotation_range=20,      # Rotate image from 0 to 20 degrees
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,         # Zoom in or zoom out range
                                   horizontal_flip=True,    # Flip image horizontally
                                   fill_mode='nearest')

train_generator = train_datagen.flow_from_directory(train_dir,
                                                    target_size=(224,224),
                                                    color_mode='rgb',
```

```
        shuffle=True,  
        class_mode='categorical')  
  
val_generator = train_datagen.flow_from_directory(val_dir,  
        target_size=(224,224),  
        color_mode='rgb',  
        shuffle=True,  
        class_mode='categorical')  
  
# Test Generator - Augmentation  
test_datagen = ImageDataGenerator(rescale=1./255)  
test_generator = test_datagen.flow_from_directory(test_dir,  
        target_size=(224,224),  
        color_mode='rgb',  
        batch_size=32,  
        # shuffle=True,  
        class_mode='categorical')
```

Transfer Learning VGG-19

```
# base mode VGG-19

from tensorflow.keras.applications.vgg19 import VGG19

base_model_vgg19 = VGG19(weights='imagenet', include_top=False,
input_shape=(224,224, 3))

# add base model VGG-19

x = base_model_vgg19.output

# add global average pooling layer
x = GlobalAveragePooling2D()(x)

# add a fully-connected layer
x = Dense(1024, activation='relu')(x)

# and a logistic layer (9 classes)
predictions = Dense(9, activation='softmax')(x)

# this is the model we will train
model_vgg19 = Model(inputs=base_model_vgg19.input, outputs=predictions)

# first: train only the top layers (which were randomly initialized)
# i.e. freeze all convolutional InceptionV3 layers
for layer in base_model_vgg19.layers:
    layer.trainable = False

model_vgg19.summary()

Skenario 18 (input_shape=224,224; batch_size=64; epochs=50; optimizer=Adadelta;
learning_rate=0.005)

model_skenario18 = model_vgg19

# compile the model (should be done *after* setting layers to non-trainable)
```

```
model_skenario18.compile(optimizer=Adadelta(learning_rate=0.005),
loss='categorical_crossentropy', metrics=[metrics.AUC(name='auc'), 'accuracy'])

es_callback=EarlyStopping(monitor='val_auc', mode='max', patience=8, verbose=1,
min_delta=0.0001, restore_best_weights=True)

# train the model on the new data for a few epochs

hist_model_skenario18 = model_skenario18.fit(train_generator,

steps_per_epoch=train_generator.samples//64, # angka 64
merupakan nilai batch_size

epochs=50,
validation_data=val_generator,
validation_steps=val_generator.samples//64,
batch_size=64,
callbacks=[es_callback])

# menampilkan grafik akurasi model

acc = hist_model_skenario18.history['accuracy']

val_acc = hist_model_skenario18.history['val_accuracy']

loss = hist_model_skenario18.history['loss']

val_loss = hist_model_skenario18.history['val_loss']

epochs_range = range(len(acc))

plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)

plt.plot(epochs_range, acc, "ro-", label='Training Accuracy')

plt.plot(epochs_range, val_acc, "bo-", label='Validation Accuracy')

plt.title('Training and Validation Accuracy Model Skenario 18')
```



```
plt.xlabel("Epoch")  
plt.ylabel("Accuracy")  
plt.legend(loc='lower right')  
plt.subplot(2, 2, 2)  
plt.plot(epochs_range, loss, "ro-", label='Training Loss')  
plt.plot(epochs_range, val_loss, "go-", label='Validation Loss')  
plt.title('Training and Validation Loss Model Model Skenario 18')  
plt.xlabel("Epoch")  
plt.ylabel("Loss")  
plt.legend(loc='upper right')  
plt.show()
```



Evaluasi Model (Confusion Matrix & Classification Report)

```
batch_size=64

target_size=(224,224)

test_path = '/content/drive/MyDrive/My Dataset/mushrooms/test'

test_generator = test_datagen.flow_from_directory(

    test_path,

    target_size=target_size,

    batch_size=batch_size,

    class_mode=None,

    shuffle=False)

test_generator.reset()

# Calling the saved model for making predictions

model_klasifikasi = load_model("/content/drive/MyDrive/My Model/vgg19-batch64-epochs50-adadelata-lr0005.h5")

pred = model_klasifikasi.predict(test_generator, verbose=1)

predicted_class_indices=np.argmax(pred,axis=1)

labels = (test_generator.class_indices)

labels = dict((v,k) for k,v in labels.items())

predictions = [labels[k] for k in predicted_class_indices]

filenames = test_generator.filenames

results = pd.DataFrame({"Filename":filenames,

    "Predictions":predictions})

# create a function for visualizing model performance

def PerformanceReports(conf_matrix, class_report, labels):
```

```
ax = plt.figure(figsize=(15, 10))  
  
# ax = plt.subplot(2, 2, 1)  
  
ax = plt.subplot()  
  
sns.heatmap(conf_matrix, cmap="crest_r", annot=True, fmt='.4g', linewidths=2,  
linecolor='white', cbar=True, ax=ax, annot_kws={"size":20})  
  
#labels, title and ticks  
  
ax.set_xlabel('Predicted labels', fontsize=12, color="darkblue", labelpad=24)  
ax.set_ylabel('True labels', fontsize=12, color="darkblue", labelpad=24)  
ax.set_title('Confusion Matrix Model Skenario 4', fontsize=12, pad=24)  
ax.xaxis.set_ticklabels(labels)  
ax.yaxis.set_ticklabels(labels)  
plt.show()  
  
ax = plt.figure(figsize=(15, 10))  
# ax= plt.subplot(2, 2, 2)  
ax = plt.subplot()  
sns.heatmap(pd.DataFrame(class_report).iloc[:-1, :].T,  
            annot=True,ax=ax)  
  
ax.set_title('Classification Report Model Skenario 15', fontsize=12, pad=24)  
plt.show()  
  
labels = ['agaricus', 'amanita', 'boletus', 'cortinarius', 'entoloma', 'hygrocybe',  
'lactarius', 'russula', 'suillus']  
  
test_labels = [fn.split('/')[0] for fn in filenames]  
  
cm = confusion_matrix(test_labels, predictions)
```

```
print(cm)

cr = classification_report(test_labels, predictions)

class_report = classification_report(test_labels,
                                    predictions,
                                    target_names=labels,
                                    output_dict=True)

print(cr)

PerformanceReports(cm, class_report, labels)
```



Source Code Tampilan Web

Sukses.html

```
<!DOCTYPE html>

<html lang="eng">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet" href={{url_for('static', filename = 'bootstrap-5.3.0/css/bootstrap.min.css')}}>

    <!-- Bootstrap Font Icon CSS -->

    <link rel="stylesheet" href={{url_for('static', filename = 'bootstrap-icons-1.10.5/font/bootstrap-icons.min.css')}}>

    <title>Hasil Klasifikasi</title>

  </head>

  <body>

    <nav class="navbar bg-body-tertiary">

      <div class="container-fluid">

        <a class="navbar-brand" href="#">

           Mushroom Classification

        </a>

        <a href="/" class="btn btn-success">

          <i class="bi bi-arrow-left-square"></i> Back to Home

        </a>

      </div>

    </nav>

  </body>

</html>
```



```
</div>

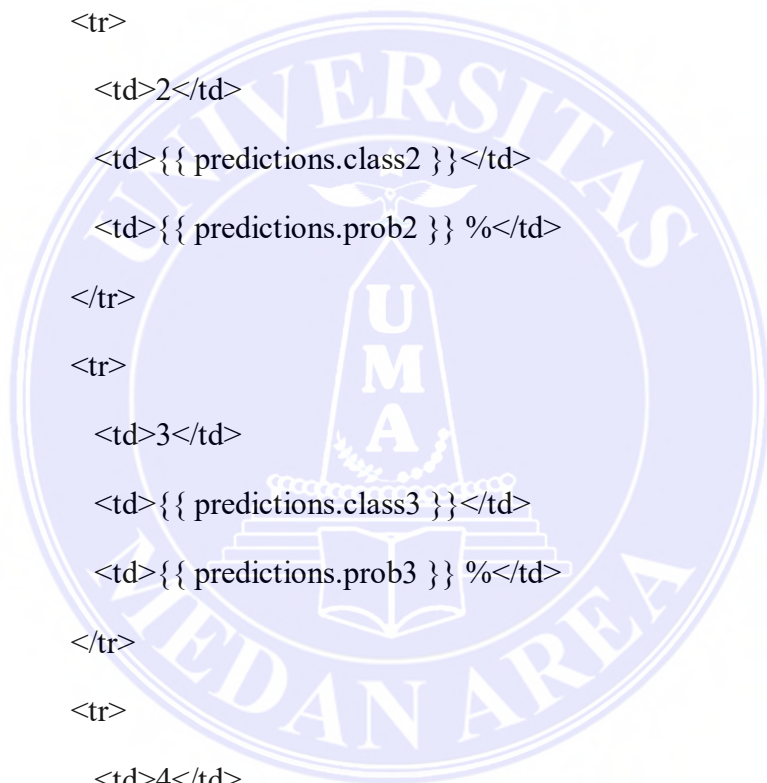
</nav>

<div class="container-fluid pt-3">
  <div class="row">
    <div class="col-md-6">
      <div class="card">
        <div class="card-header">Citra Input</div>
        <div class="card-body">
          <img class="img-thumbnail rounded mx-auto d-block"
src={{url_for('static', filename='images/'+ img)}} >
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <div class="card">
        <div class="card-header">Hasil Klasifikasi</div>
        <div class="card-body">
          <table class="table table-striped table-hover">
            <tr>
              <th>Rank</th>
              <th>Class (Jenis Jamur)</th>
              <th>Probability</th>
            </tr>
```

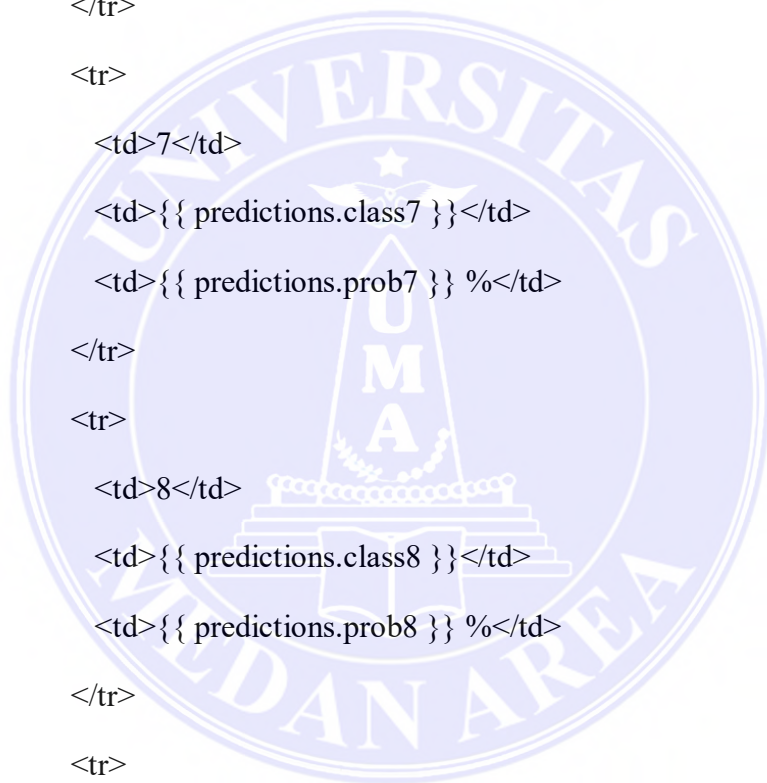
```

<tr class="table-success">
  <td>1</td>
  <td><span class="badge text-bg-success">{{ predictions.class1
}}</span></td>
  <td><span class="badge text-bg-success">{{ predictions.prob1 }}
%</span></td>
</tr>
<tr>
  <td>2</td>
  <td>{{ predictions.class2 }}</td>
  <td>{{ predictions.prob2 }} %</td>
</tr>
<tr>
  <td>3</td>
  <td>{{ predictions.class3 }}</td>
  <td>{{ predictions.prob3 }} %</td>
</tr>
<tr>
  <td>4</td>
  <td>{{ predictions.class4 }}</td>
  <td>{{ predictions.prob4 }} %</td>
</tr>
<tr>
  <td>5</td>
  <td>{{ predictions.class5 }}</td>

```



```
<td>{{ predictions.prob5 }} %</td>
</tr>
<tr>
<td>6</td>
<td>{{ predictions.class6 }}</td>
<td>{{ predictions.prob6 }} %</td>
</tr>
<tr>
<td>7</td>
<td>{{ predictions.class7 }}</td>
<td>{{ predictions.prob7 }} %</td>
</tr>
<tr>
<td>8</td>
<td>{{ predictions.class8 }}</td>
<td>{{ predictions.prob8 }} %</td>
</tr>
<tr>
<td>9</td>
<td>{{ predictions.class9 }}</td>
<td>{{ predictions.prob9 }} %</td>
</tr>
</table>
</div>
```



```
</div>  
</div>  
</div>  
</div>  
  
</body>  
</html>
```



Indeks.html

```
<!DOCTYPE html>

<html lang="eng">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet" href={{url_for('static', filename = 'bootstrap-5.3.0/css/bootstrap.min.css')}}>

    <!-- Bootstrap Font Icon CSS -->

    <link rel="stylesheet" href={{url_for('static', filename = 'bootstrap-icons-1.10.5/font/bootstrap-icons.min.css')}}>

    <title>Klasifikasi Jenis Jamur</title>

  </head>

  <body>

    <nav class="navbar bg-body-tertiary">

      <div class="container-fluid">

        <a class="navbar-brand" href="#">

           Mushroom Classification

        </a>

        <a class="btn btn-outline-success d-flex" href="#">Rahmad Syuhada</a>

      </div>

    </nav>
```



```
<div class="alert alert-success" role="alert">
    Silahkan upload file citra jamur (Agaricus, Amanita, Boletus, Cortinarius,
    Entoloma, Hygrocybe, Lactarius, Russula, Suillus),
    lalu pilih tombol <b>Klasifikasi</b>
</div>
<div class="container-fluid pt-3">
    <form action="/success" method="post", enctype="multipart/form-data">
        <div class="input-group mb-3">
            <input class="form-control" type="file", name = "file"/>
        </div>
        <div class="input-group mb-3 pt-3">
            <button class="btn btn-success">
                <i class="bi bi-gear-wide-connected"></i> Klasifikasi</button>
            </div>
        </form>
    </div>
<!-- <div class="container-fluid pt-3">
    <div class="row">
        <div class="col-md-5">
            <div class="card">
                <div class="card-header">Top 3 Accuracy & Loss Model</div>
                <div class="card-body">
                    
                </div>
            </div>
        </div>
    </div>
</div>
```

```
        

    </div>

</div>

</div>

<div class="col-md-4">

    <div class="card">

        <div class="card-header">Confussion Matrix & Classification
Report</div>

        <div class="card-body">

        </div>

    </div>

</div>

<div class="col-md-3">

    <div class="card">

        <div class="card-header">Sampel Dataset (Jenis Jamur)</div>

        <div class="card-body">

            
```

```














</div>

</div>

</div>

</div>

</div> -->

</body>

</html>
```