

**KLASIFIKASI HAMA SERANGGA PADA PERKEBUNAN
KELAPA SAWIT**

SKRIPSI

OLEH:

BAMBANG PRATAMA

198160039



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2024

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 11/7/24

Access From (repository.uma.ac.id)11/7/24

**ANALISIS KINERJA ARSITEKTUR GOOGLINET UNTUK
KLASIFIKASI HAMA SERANGGA PADA PERKEBUNAN
KELAPA SAWIT**

SKRIPSI

OLEH:

BAMBANG PRATAMA

198160039



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2024

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 11/7/24

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

Access From (repository.uma.ac.id)11/7/24

**ANALISIS KINERJA ARSITEKTUR GOOGLINET UNTUK
KLASIFIKASI HAMA SERANGGA PADA PERKEBUNAN
KELAPA SAWIT**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh

Gelar Sarjana (S1) di Fakultas Teknik

Universitas Medan Area

OLEH :

BAMBANG PRATAMA

198160039

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2024

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

LEMBAR PENGESAHAN

Judul Skripsi : Analisis Kinerja Arsitektur GoogleNet Untuk Klasifikasi
Hama Serangga Pada Perkebunan Kelapa Sawit


Nama : Bambang Pratama

NPM : 198160039

Fakultas : Teknik

Program Studi : Teknik Informatika

Disetujui Oleh
Komisi Pembimbing


Rizki Muliono, S.Kom, M.Kom

Diketahui Oleh:

Dekan Fakultas Teknik

Ketua Prodi Teknik Informatika



Dr. Edy Supratno, S.T, M.T.
NIDN : 0102027402



Rizki Muliono, S.Kom, M.Kom
NIDN : 0109038902

Tanggal Lulus: 28 Maret 2024

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.

Medan, 28 Maret 2024



Bambang Pratama
198160039

HALAMAN PERNYATAAN PERSETUJUAN

PUBLIKASISKRIPSI UNTUK KEPENTINGANAKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawahini:

Nama : Bambang Pratama

NPM : 198160039

ProgramStudi : Teknik Infomatika

Fakultas : Teknik

JenisKarya : Skripsi

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

Analisis Kinerja Arsitektur GoogleNet Untuk Klasifikasi Hama Serangga Pada Perkebunan Kelapa Sawit.

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Medan Area berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir/skripsi/tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Di buat di: Medan
Pada tanggal: 28 Maret 2024
Yang menyatakan



Bambang Pratama
NPM 198160039

ABSTRAK

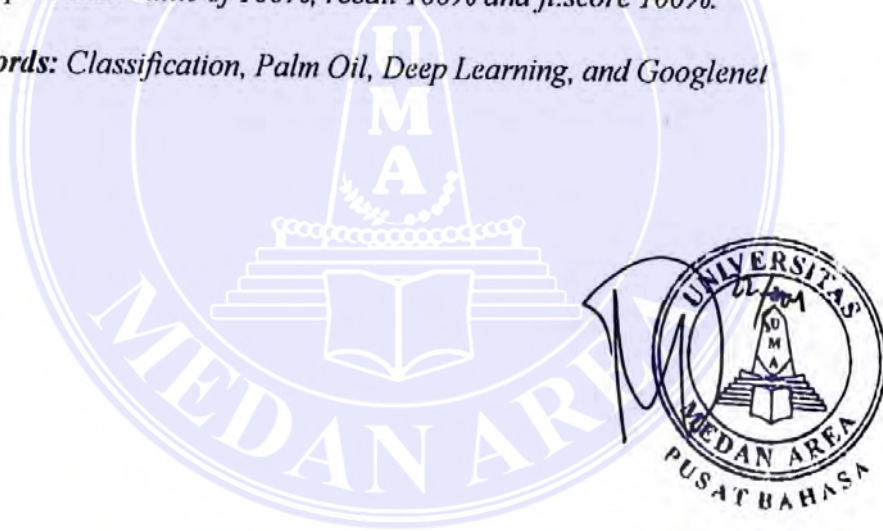
Indonesia merupakan negara penghasil utama minyak kelapa sawit yang memproduksi lebih dari 44%. Kelapa sawit sendiri merupakan sumber utama minyak *nabati* yang memberikan kontribusi besar minyak *nabati* dalam perdagangan dunia. Dalam aktivitas budidaya kelapa sawit terdapat penurunan hasil produksi yang dialami para petani maupun perusahaan yang disebabkan oleh salah satunya yaitu hama pada tanaman kelapa sawit. Terdapat banyak hama pada tanaman kelapa sawit salah satunya hama UPDKS (Ulat Pemakan Dan Kelapa Sawit) seperti ulat api *setora nitens*, ulat kantong *metisa plana* dan ulat api *setothosea asigna*. Identifikasi dan klasifikasi hama tanaman kelapa sawit secara manual oleh para petani maupun perusahaan membutuhkan waktu yang cukup tinggi. Oleh karena itu perlu dilakukan analisis kinerja arsitektur *GoogleNet* dalam konteks klasifikasi hama pada tanaman kelapa sawit. Analisis ini diharapkan dapat memberikan pemahaman lebih dalam tentang keunggulan dan keterbatasan dari kinerja arsitektur *GoogleNet* dalam mengenali dan mengklasifikasikan hama pada tanaman kelapa sawit. Penerapan metode *Convolutional Neural Network* dengan arsitektur *GoogleNet* menghasilkan performa yang baik dengan jumlah dataset sebanyak 3076 gambar, yang terdiri dalam 3 *class*. Pada skenario model yang menggunakan *hyperparameter* dengan jumlah *epoch* 25, *batchsize* 32, *optimizer* *RMSprop*, *learning rate* 0.001 didapatkan hasil akurasi sebesar 100% dengan nilai persisi (*Precision*) sebesar 100%, *recall* 100% dan *f1.score* 100%.

Kata Kunci : Klasifikasi, Kelapa Sawit, Deep Learning, dan GoogleNet

ABSTRACT

Indonesia is a major producer of palm oil, producing more than 44%. Palm oil itself is the main source of vegetable oil which contributes greatly to vegetable oil in world trade. In oil palm cultivation activities, there is a decrease in production experienced by farmers and companies caused by one of them, namely pests on oil palm plants. There are many pests on oil palm plants, one of which is CTAOP (Caterpillar Eater and Oil Palm) pests such as *Setora nitens* fire caterpillar, *Metisa plana* bag caterpillar and *Setothosea asigna* fire caterpillar. Manual identification and classification of oil palm plant pests by farmers and companies requires a high amount of time. Therefore, it was necessary to analyse the performance of GoogleNet architecture in the context of pest classification in oil palm plants. This analysis was expected to provide a deeper understanding of the advantages and limitations of GoogleNet architecture performance in recognising and classifying pests in oil palm plants. The application of the Convolutional Neural Network method with GoogleNet architecture produced good performance with a total dataset of 3076 images, which were divided into 3 classes. In the model scenario that used hyperparameters with the number of epochs 25, batch size 32, RMSprop optimiser, learning rate 0.001, the accuracy result was 100% with a precision value of 100%, recall 100% and fl.score 100%.

Keywords: Classification, Palm Oil, Deep Learning, and Googlenet



RIWAYAT HIDUP

Penulis lahir di Desa Meranti Paham, Kecamatan Panai Hulu, Kabupaten Labuhanbatu, Provinsi Sumatera Utara pada tanggal 13 Agustus 2001 dari Bapak Sugeng Hariyanto S.Kom, dan Ibu Ngatinem. Penulis merupakan anak pertama dari 3 (tiga) bersaudara yakni Keyla Winna Fitri selaku anak kedua dan Ranggi Sahputra selaku anak ketiga.

Penulis pertama kali mengenyam Pendidikan dibangku Sekolah Dasar (SD) Negeri 116248 Dusun IV Desa Meranti Paham, Kec. Panai Hulu, Kab. Labuhanbatu, pada tahun 2007-2013, kemudian meneruskan Pendidikan di Madrasah Tsanawiyah (MTs) Sawasta As-Shiddiq Dusun VI Desa Meranti Paham, Kec. Panai Hulu, Kab. Labuhanbatu, diselesaikan pada tahun 2013-2016, Meneruskan Pendidikan Sekolah Menengah Kejuruan (SMK) Negeri 1 Dusun VII Desa Meranti Paham, Kec. Panai Hulu, Kab. Labuhanbatu pada tahun 2016-2019.

Pada tahun 2019 penulis lulus dari SMKN 1 Panai Hulu dan pada 2019 terdaftar sebagai mahasiswa Fakultas Teknik Prodi Teknik Informatika Universitas Medan Area. Pada saat ini tahun 2024 penulis sedang menjalani semester 9 (Sembilan) menyusun tugas akhir skripsi dengan judul "Analisis Kinerja Arsitektur *GoogleNet* Untuk Klasifikasi Hama Serangga Pada Perkebunan Kelapa Sawit." dengan dosen pembimbing Bapak Rizki Muliono, S.Kom, M.Kom.

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 ialah *Deep Learning* dengan judul “Analisis Kinerja Arsitektur *Googlenet* Untuk Klasifikasi Hama Serangga Pada Perkebunan Kelapa Sawit” dengan dosen pembimbing Bapak Rizki Muliono, S.Kom, M.Kom.

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. Keluarga tercinta penulis yaitu ayah saya Sugeng Hariyanto S.Kom, ibu penulis Ngatinem, dan kedua adik kandung penulis Keyla Winna Fitri dan Ranggi Sahputra yang telah memberikan semangat, dukungan, serta motivasi, dan membantu penulis dalam segi materi dan moril sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik baiknya. Dan selalu berdo'a kepada Allah SWT untuk kesuksesan anaknya kedepan.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc. selaku Rektor Universitas Medan Area.
3. Bapak Dr.Eng., Supriatno, S.T, M.T.selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Bapak Rizki Muliono, S.Kom, M.Kom. selaku Kepala Program Studi Teknik Informatika Universitas Medan Area sekaligus dosen pembimbing yang telah banyak membantu memberi masukan serta arahan kepada penulis dalam

menyelesaikan skripsi ini. Dan selalu memberikan semangat serta motivasi dan ilmu kepada penulis selama masa perkuliahan sampai pada ke tahap pengerjaan skripsi ini.

5. Bapak Muhathir, S.T, M.Kom, selaku Dosen Mata kuliah yang telah membantu penulis dari segi materi, moril dan banyak membantu memberi masukan serta arahan kepada penulis dalam menyelesaikan skripsi ini sehingga penulis dapat menyelesaikan skripsi ini.
6. Seluruh Dosen dan Staf Program Studi Teknik Informatika Universitas Medan Area.
7. Teman dekat penulis, Sury Anggraini, Muhammad Fikri Khoir, Irfan Jamaluddin, Andre Purnawan, Dedi Syahputra, Luky Arso, dan Agung Prastio, yang selalu memberikan semangat serta motivasi selama penulisan skripsi ini.
8. Seluruh teman-teman yang sudah memberikan dukungannya selama penulisan proposal skripsi ini, khususnya teman-teman Teknik Informatika yaitu Dian Ifantiska, Yolanda Kitty Sastria, Faizal Razaq, Iqbal Fhilla Bukit, Muhammad Rifqi, dan Seluruh teman seperjuangan stambuk 19 teknik informatika tidak dapat disebutkan satu persatu yang membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa tugas akhir/skripsi/tesis ini masih memiliki kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan tugas akhir ini. Penulis berharap tugas

akhirini dapat bermanfaat baik untuk kalangan pendidikan maupun masyarakat.

Akhir kata penulis ucapkan terima kasih.

Medan, 28 maret 2024



Bambang Pratama
198160039



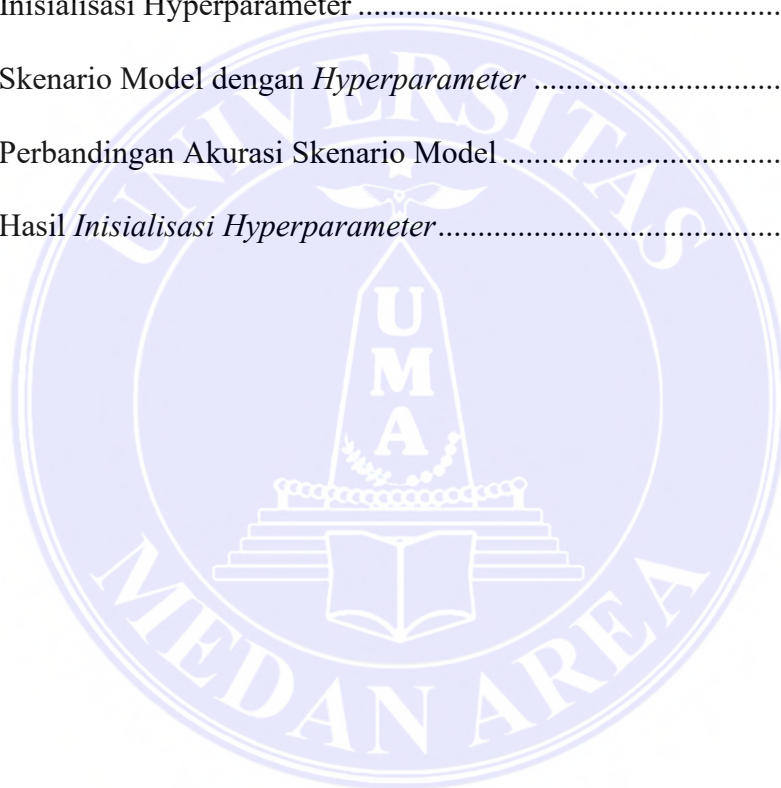
DAFTAR ISI

LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	iv
ABSTRAK	v
ABSTRAK	vi
RIWAYAT HIDUP	vii
KATA PENGANTAR	viii
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	5
1.3. Tujuan.....	6
1.4. Batasan Masalah.....	6
1.5. Manfaat Penelitian.....	7
1.6. Sistematika Penulisan.....	7
BAB II TINJAUAN PUSTAKA	10
1.1. <i>Convolutional Neural Network (CNN)</i>	10
1.2. <i>GoogleNet</i>	14
1.3. <i>Tools</i> Pendukung	16
1.3.1. <i>Python</i>	16
1.3.2. <i>Google Colaboratory</i>	19
1.4. Tanaman Kelapa Sawit.....	20
1.5. Hama Serangga Pada Kelapa Sawit	21
2.4.1. Ulat Api	21
2.4.2. Ulat Kantong.....	22
1.6. Penelitian Terkait.....	22

BAB III METODE PENELITIAN	25
3.1. Langkah Penelitian	25
3.2. Teknik Pengumpulan Data	26
3.3. Pra-Pemrosesan	27
3.3.1 Cropping data gambar.....	28
3.3.2 Resizing Data.....	29
3.3.3 <i>Augmentasi</i> Dataset	30
3.3.4 Data Training.....	33
3.3.5 Data Testing.....	33
3.4. Pemodelan <i>GoogleNet</i>	33
3.4.1. Perhitungan Manual CNN dalam Mengolah Gambar	34
3.5. Evaluasi Model.....	43
3.5.1. Confusion Matrix.....	43
BAB IV HASIL DAN PEMBAHASAN.....	45
4.1. Hasil.....	45
4.2. Pengumplan Data.....	45
4.2.1. <i>Inisialisasi Hyperparameter</i>	47
4.2.2. Skenario Model.....	48
4.2.3. <i>Implementasi</i>	49
4.3. Hasil Evaluasi Model	55
4.4. Pembahasan	60
BAB V KESIMPULAN DAN SARAN	62
5.1. Kesimpulan.....	62
5.2. Saran	63
DAFTAR PUSTAKA	64
LAMPIRAN	70

DAFTAR TABEL

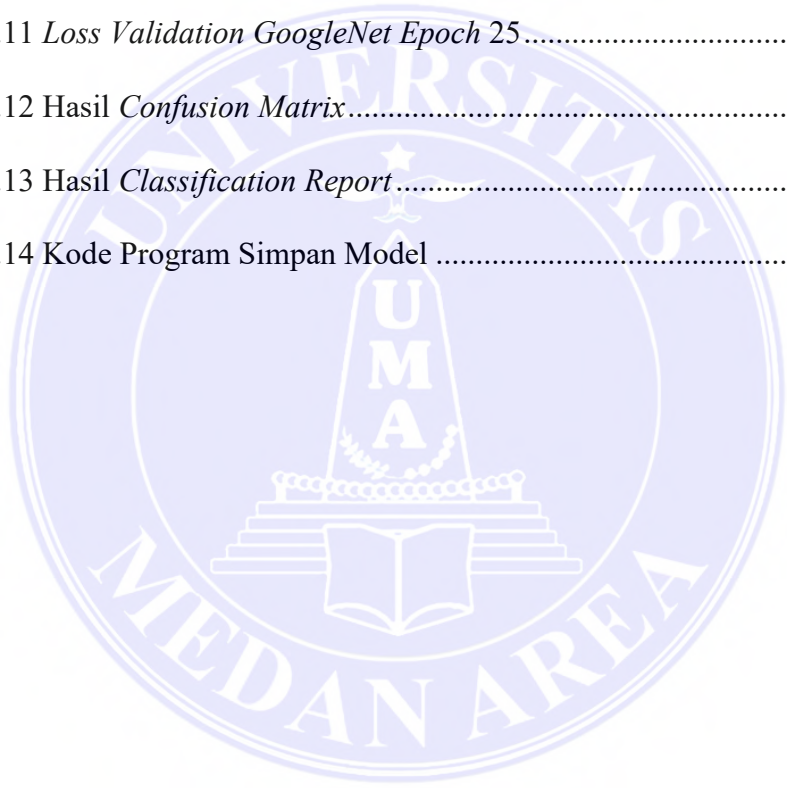
Tabel 2.1 Penelitian Terkait.....	22
Tabel 3.1 Deskripsi Citra dan Jumlah Data Citra	26
Tabel 3.2 <i>confusion matrix</i>	42
Tabel 3.3 <i>Confusion Matrix</i>	43
Tabel 4.1 Pembagian Data Training dan Testing	46
Tabel 4.2 Inisialisasi Hyperparameter	47
Tabel 4.3 Skenario Model dengan <i>Hyperparameter</i>	48
Tabel 4.4 Perbandingan Akurasi Skenario Model.....	55
Tabel 4.5 Hasil <i>Inisialisasi Hyperparameter</i>	56



DAFTAR GAMBAR

Gambar 2.1 <i>Convolutinal Neural Network</i> (Pulung, Indah & Rudy, 2020).....	10
Gambar 2.2 Proses Konvolusi dengan 2 filter untuk menghasilkan Feature <i>Map/Activation map</i> (Nugroho, dkk., 2020).....	11
Gambar 2.3 Contoh Diagram <i>Max Pooling Layer</i> (Santoso & Ariyanto, 2018).....	12
Gambar 2.4 Fungsi <i>Akrivitas ReLU</i> (Maysanjaya, 2020)	13
Gambar 2.5 Arsitektur <i>GoogleNet</i>	15
Gambar 2.6 Tampilan <i>Google Colab</i>	19
Gambar 3.1 Diagram Langkah Penelitian.....	25
Gambar 3.2 Proses <i>Cropping</i> setiap class dengan <i>MicrosoftOfficePictureManager</i> ..	28
Gambar 3.3 Proses <i>Cropping</i> Sesudah Dan Sebelum	29
Gambar 3.4 Simulasi <i>ResizeImage</i> Sesudah Dan Sebelum	29
Gambar 3.5 Proses Pelabelan Dataset.....	30
Gambar 3.6 Hasil <i>Augmentasi</i> Dataset Pada Ulat Api <i>SetoraNitens</i>	31
Gambar 3.7 <i>Augmentasi</i> Dataset Pada Ulat Kantong <i>Metisaplana</i>	32
Gambar 3.8 Hasil <i>Augmentasi</i> Dataset Pada Ulat Api <i>SetothoseaAsigna</i>	32
Gambar 3.9 Modul <i>Inception</i> pada <i>GoogleNet</i>	34
Gambar 3.10 Ulat api <i>Setora Nitens</i>	34
Gambar 3.11 Pixel citra dengan 3 channel	35
Gambar 3.12 <i>fungsi aktivasi ReLu</i>	39
Gambar 3.13 <i>Fully Connected Layer</i>	40
Gambar 4.1 Sampel Dataset Ulat Api (<i>SetoraNitens</i>)	46
Gambar 4.2 Sampel Dataset Ulat Kantong (<i>Metisa Plana</i>).....	46
Gambar 4.3 Dataset Ulat Api (<i>Setothosea Asigna</i>).....	46

Gambar 4.4 <i>Training Accuracy GoogleNet Epoch 15</i>	50
Gambar 4.5 <i>Loss Accuracy GoggleNet Epoch 15</i>	50
Gambar 4.6 <i>Accuracy Validation GoogleNet Epoch 15</i>	51
Gambar 4.7 <i>Loss Validation GoogleNet Epoch 15</i>	51
Gambar 4.8 <i>Training Accuracy GoogleNet Epoch 25</i>	52
Gambar 4.9 <i>Training Accuracy GoogleNet Epoch 25</i>	53
Gambar 4.10 <i>Accuracy Validation GoogleNet Epoch 25</i>	54
Gambar 4.11 <i>Loss Validation GoogleNet Epoch 25</i>	54
Gambar 4.12 Hasil <i>Confusion Matrix</i>	58
Gambar 4.13 Hasil <i>Classification Report</i>	58
Gambar 4.14 Kode Program Simpan Model	59



BAB I

PENDAHULUAN

1.1. Latar Belakang

Kelapa sawit (*Elaeis Guinensis Jacq*) merupakan tumbuhan tropis golongan palma yang termasuk tanaman tahunan dan habitat aslinya adalah daerah semak belukar(Nanda dkk., 2022). Tanaman kelapa sawit juga merupakan salah satu tumbuhan perkebunan yang memiliki prospek industri yang baik dipasar lokal maupun pasar dunia(Oktafanda, 2022). Tanaman kelapa sawit sendiri berasal dari amerika selatan dan afrika(Raffar dkk., 2022). Di Indonesia sendiri tanaman kelapa sawit pertama kali diperkenalkanoleh pemerintahan kolonial belanda pada tahun 1848 yang ditanam di daerah kebun raya bogor (Sartika, 2022). Tumbuhan kelapa sawit biasanya tumbuh di lahan dengan kontur tanah gambut, tanah humus dan lainnya(Lubis, 2020). Kelapa sawit dapat tumbuh dengan tinggi mencapai 0-25 Meter, dan Kelapa sawit memiliki masa produksi yaitu mencapai 25 tahun. Perkebunan kelapa sawit di indonesia saat ini sudah semakin berkembang dan menjadikan indonesia sebagai penghasil utama minyak kelapa sawit yang memproduksi lebih dari 44% minyak kelapa sawit dunia (Widians & Rizkyani, 2020). Selain itu perkebunan kelapa sawit hampir ada di seluruh bagian kepulauan Nusantara salah satunya pada Provinsi Kalimantan Timur.Potensi produksi kelapa sawit di Kalimantan Timur sangat besar dan selalu mengalami peningkatan produksi dalam tahun ke tahun. Namun,mengapa sering terdengar dari beberapa para petani ataupun perusahaan besar perkebunan kelapa sawit mengalami penurunan hasil produksi.

Ternyata dalam aktivitas budidaya tanaman kelapa sawit, setiap perusahaan ataupun para petani tidak terlepas dari faktor-faktor yang menghambat perkembangan, pertumbuhan, dan massa produksi tanaman sawit, seperti salah prosedur dalam pemeliharaan, pemupukan, dan pengendalian serangan hama. Karena serangan dari hama serangga pada tanaman kelapa sawit dapat menyebabkan penurunan produksi bahkan kematian pada tanaman (Turnip, 2021). Hama sendiri merupakan salah satu organisme pengganggu segala tanaman apapun itu yang menjadi faktor penting yang harus diperhatikan dalam perkebunan kelapa sawit. Hama yang sering menyerang tanaman perkebunan kelapa sawit ada banyak macam dan jenisnya namun peneliti hanya memilih salah satunya yaitu serang dari hama-hama Ulat Pemakan Daun Kelapa Sawit (UPDKS) yaitu ulat api dan ulat kantung yang dapat mengakibatkan kerusakan bahkan kehilangan daun pada kelapa sawit dan dampak dari serangan hama ulat api, dan ulat kantung dapat merugikan para petani perkebunan kelapa sawit dan juga dapat menurunkan produksi kelapa sawit (Agustina, 2021). Pada umumnya hama UPDKS ada banyak dan berbagai macam jenisnya, namun terdapat 4 jenis ulat api yang sering menyerang kelapa sawit di antaranya: *Setothosea asigna*, *Setoranitens*, *Darnatrima*, dan *Parasa Lepida*. (Hari dkk., 2020), (Arumugam dkk., 2019), sedangkan untuk ulat kantung terdapat 4 jenis yang kerap kali menyerang kelapa sawit diantaranya: *Mahasena Corbetti*, *Metisa Plana*, *Pteroma Pendula*, dan *Clania Tertia* (Dibisono dkk., 2021), (Saragih & Afrianti, 2021), (S. N. Ahmad dkk., 2020), (Setyawan dkk., 2021). Pada umumnya ulat api merupakan hama utama pada tanaman kelapa sawit dimana serangan oleh hama ini dapat kehilangan daun mencapai

70-90 % , (Pratama, 2021). Sedangkan ulat kantong akibat kerusakan dari hama ini dapat menimbulkan penyusutan produksi sampai 40%,(Yuliani dkk., 2019).

Penelitian klasifikasi terkait hama telah banyak diteliti sebelumnya pada peneliti-peneliti terdahulu seperti yang disampaikan oleh (Widians & Rizkyani, 2020), dengan menggunakan metode *certainty factor* mendapatkan hasil yaitu sebesar 88,8%. Selanjutnya oleh (Rahman dkk., 2020) dengan menggunakan arsitektur CNN seperti VGG16 dan *InceptionV3* mencapai akurasi 93,33%. Dan arsitektur *AlexNet* juga digunakan oleh (Dawei.dkk.,2019), untuk mengidentifikasi 10 jenis OPT (organisme pengganggu tanaman) dengan akurasi 93,84%. Hal yang sama juga dileliti oleh (Liu dkk., 2016) yang menggunakan *AlexNet* dan didapatkan *mean Accuracy Precision* (mAP) sebesar 0,951. Dan selanjutnya juga telah disampaikan oleh (M. N. Ahmad dkk., 2021), mengidentifikasi ulat kantong hidup dan mati *Metisa plana* (pertama sampai ketiga tahap instar) menggunakan teknik *motion tracking* pada pelepah kelapa sawit, dengan akurasi tinggi masing-masing sebesar 87,5% dan 78,8%.

Berdasarkan jenis dari masing-masing ulat api dan ulat kantong dengan keunikan bentuk dan teksturnya, maka dalam penelitian ini mencoba untuk mengklasifikasi jenis hama dengan memanfaatkan teknologi komputasi *DeepLearning*. Adapun berbagai macam arsitektur *Deep Learning* yang berkembang saat ini diantaranya *AlexNet*(Nurona Cahya dkk., 2021), *VisualGeometryGroup* (VGG) 16(AGUSTINA dkk., 2022), *VGG19*(Susanto dkk., 2023), *Residual Network (ResNet)*50 (Nashrullah dkk., 2020), *ResNet101* (Ghosal dkk., 2019), *GoogLeNet* (Yuesheng dkk., 2021), *Inception-V3* (Ramaneswaran dkk., 2021), *Inception-ResNet-V2* (Dominic dkk., 2021), dan

Squeezenet (Angga Irawan dkk., 2021). Namun dari kesekian banyak arsitektur *Deep Learning* peneliti memilih salah satunya yaitu Arsitektur *GoogleNet*.

GoogleNet adalah salah satu arsitektur dari CNN (*Convolutional Neural Network*) yang diperkenalkan oleh *google* sejak tahun 2014. Pada tahun 2014 juga arsitektur ini memiliki tingkat akurasi yang tinggi dan menjadi pemenang pada kompetisi ILSVRC (*ImageNet Large-Scale Visual Recognition Challenge*) yang mendapatkan predikat sebagai arsitektur kinerja paling baik (Arif Faizin dkk., 2022). ILSVRC adalah suatu kompetisi tahunan yang melombakan klasifikasi gambar dengan berbagai modifikasi CNN (Farhan dkk., 2021). *GoogleNet* juga memiliki penyatuan rata-rata global yang memiliki metode untuk mengurangi pemrosesan komputasi dan meningkatkan akurasi (Nisa dkk., 2020). Penelitian yang membahas arsitektur *GoogleNet* sendiri juga telah banyak diteliti pada penelitian sebelumnya seperti yang telah disampaikan oleh (Patani dkk., 2021), yaitu mendeteksi penyakit daun menggunakan *GoogleNet* mendapatkan akurasi yang sangat tinggi yaitu 99,35 %. Selanjutnya juga telah disampaikan oleh (Mustamin dkk., 2021), yang mengklasifikasi kualitas dari kayu dengan menggunakan arsitektur dari CNN yaitu *GoggleNet* yang mendapatkan hasil dengan rata-rata akurasi 84,89%. Berikutnya juga disampaikan oleh (Arsitektur dkk., 2022), dengan menggunakan arsitektur *GoogleNet* dalam mengklasifikasi penyakit daun kelapa sawit yang mendapatkan hasil akurasi yang optimal yaitu sebesar 93,22%. Dan disampaikan juga oleh (Salma dkk., 2021), yang mendeteksi *retinopati* diabetik melalui citra fundus menggunakan arsitektur *GoogleNet* yang mendapatkan akurasi yang baik dengan hasil rata rata akurasi sebesar 88%.

Berdasarkan hasil penelitian terdahulu untuk arsitektur *GoogleNet* yang membahas tentang kasus hama serangga kelapa sawit belum pernah diteliti, namun pada penelitian (Arsitektur dkk., 2022) hanya membahas tentang penyakit pada daun kelapa sawit dengan membandingkan arsitektur *GoogleNet* dan *Xception* yang mendapatkan akurasi yang tinggi dengan akurasi sebesar 99,72% untuk *Xception* dan 93,22% untuk *GoogleNet*. Oleh karena itu, peneliti mencoba untuk menganalisis kinerja arsitektur *GoogleNet* untuk klasifikasi hama serangga pada perkebunan kelapa sawit berdasarkan kasifikasi data citra. Dengan demikian, melalui penelitian yang peneliti lakukan dapat dikenali apa saja hama serangga yang terdapat pada perkebunan kelapa sawit, dan peneliti juga diharapkan mampu menunjukkan hasil yang baik dan mendapatkan akurasi yang tinggi dari pada penelitian terdahulu pada arsitektur *GoogleNet*.

1.2. Rumusan Masalah

Berdasarkan dari latar belakang diatas, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana menerapkan metode *Convolutional Neural Network* dengan arsitektur *GoogleNet* dalam mengklasifikasi hama serangga pada perkebunan kelapa sawit berdasarkan klasifikasi data citra?
2. Bagaimana tingkat akurasi yang dihasilkan metode *Convolutional Neural Network* dengan arsitektur *GoogleNe* tdalam penerapannya pada proes klasifikasi hama serangga pada perkebunan kelapa sawit berdasarkan klasifikasi data citra?

1.3. Tujuan

Berdasarkan perumusan masalah diatas terdapat tujuan dari penelitian ini yaitu untuk menganalisis kinerja arsitektur *GoogleNet* serta mampu menunjukkan apakah arsitektur *GoogleNe* tdalam klasifikasi hama pada tanaman perkebunan kelapa sawit berdasarkan data citra mendapatkan akurasi yang tinggi.

1.4. Batasan Masalah

Adapun yang menjadi batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Proses klasifikasi citra tersebut menggunakan arsitektur *GoogleNet*.
2. Pada Penelitian ini, Citra yang diinput adalah citra dari berbagai jenis masing-masing hama serangga ulat api dan hama serangga ulat kantong saja yang terdapat pada perkebunan tersebut.
3. Format gambar yang digunakan adalah JPG.
4. Jarak pengambilan citra tidak lebih dari 15 cm
5. Pengambilan sampel/gambar dilakukan di Perkebunan Kelapa Sawit Afd IV Kebun Ajamu I PTPN IV, Kecamatan Panai hulu, Kabupaten Labuhanbatu, Provinsi Sumatera Utara.
6. Jumlah dataset yang dipakai pada penelitian ini adalah 4097 data citra, dengan masing-masing *class* nya
7. Data yang digunakan dibagi menjadi 2 data yaitu data *Training* 80% dan data *Testing* 20%

1.5. Manfaat Penelitian

Penelitian tentang analisis kinerja arsitektur *GoogleNet* untuk klasifikasi hama serangga pada perkebunan kelapa sawit diharapkan dapat bermanfaat baik secara teoritis maupun secara praktis.

1. Secara Teoritis

- a. Untuk Pengembangan ilmu pengetahuan, terutama bagi dunia Teknik Informatika dan ilmu Komputer, serta dalam ilmu kedokteran maupun psikologi.
- b. Menemukan cara analisis kinerja arsitektur *GoogleNet* untuk klasifikasi hama serangga pada perkebunan kelapa sawit.

2. Secara Praktis

a. Bagi Peneliti

- 1) Memahami cara kerja arsitektur *GoogleNet* dan wujud analisis kerjanya.
- 2) Sebagai portopolio untuk peneliti yang akan berguna dimasa mendatang.

b. Bagi Universitas

- 1) Sebagai bahan pembelajaran untuk penelitian mendatang.
- 2) Sebagai bahan evaluasi bagi universitas dalam mengembangkan pengetahuan berkaitan pengolahan citra.

1.6. Sistematika Penulisan

Dalam sistematika penulisan skripsi ini dikelompokkan menjadi 5 sub-bab, antara lain :

BAB I : PENDAHULUAN

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

BAB II : TINJAUAN PUSTAKA

Pada bab ini meguraikan rangkuman dasar teori yang dihimpun dari berbagai pustaka yang relevan dengan topik yang menjadi objek kajian. Dasaryeori yang dipakai dalam penelitian ini yaitu mengenai *Deep Learning, Convolutional Neural Network, Arsitektur GoogleNet* dan beberapa teori pendukung lainnya

BAB III : METODE PENELITIAN

Dalam bab ini menjelaskan tentang tahapan-tahapan tang akan dilakukan dalam penelitian ini. Setiap rencana dari tahapan penelitian dideskripsikan secara rinci.

BAB IV :HASIL DAN PEMBAHSAN

Bab ini pula berisi hasil dari penelitian serta pemabahasan dari peneitian yang telah dilaksanakan. Dalam bab ini juga disajikan beberapa gambar, dan tabel 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 berisikan saran untuk penelitian selanjutnya yang ingin mengangkat tema yang sama dengan penelitian ini.



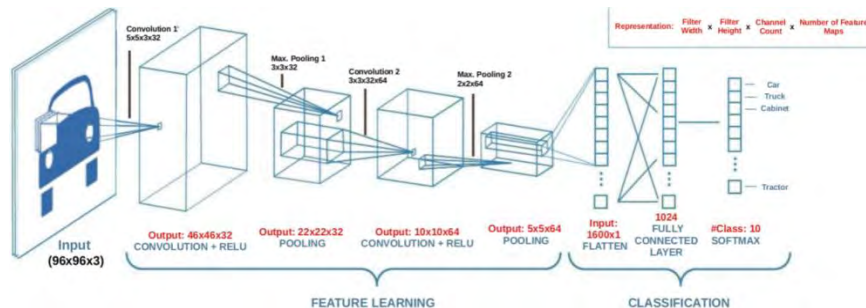
BAB II

TINJAUAN PUSTAKA

1.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan evolusi dari Multi Layer Perceptron (MLP) dan juga salah satu jenis neural network yang biasa digunakan pada data gambar (Peryanto dkk., 2020), (Arsitektur dkk., 2022). CNN juga merupakan adalah algoritma Deep Learning yang dirancang khusus untuk memproses data dalam bentuk dua dimensi, seperti gambar atau audio (Marpaung dkk., 2024)

Pada tahun 1970-an, arsitektur jaringan ini dirancang untuk bekerja dengan gambar dan menjadi aplikasi yang berguna. Pada tahun 2014, pemenang ILSVRC danmendapatkansatu tempat di kedua gambar deteksi dan klasifikasi adalah GoogleNet (Salma dkk., 2021). Arsitektur CNN terdiri darisatu lapis masukan (*input layer*), suatu lapis keluaran (*output layer*), dan sejumlah lapis tersembunyi (*hidden layers*), (Aryanto & Augusman, 2021). Secara umum CNN terdiri dari *convolution*, *pooling*, *Rectification Linear Unit (ReLU)*, dan *fully connected layer* (Suta dkk., 2019).



Gambar 2.1 Convolutinal Neural Network (Pulung, Indah & Rudy, 2020)

a. Convolutional Layer

Convolutional layer adalah lapisan pertama yang adalah dasar arsitektur sebuah CNN. Pada Convolutional layer terjadi proses operasi konvolusi dengan cara operasi kombinasi linear filter kepada wilayah lokal. Pada pemrosesan citra, konvolusi memiliki arti menerapkan satu kernel pada citra di semua offset yang membolehkan. Biasanya library yang memakai CNN, korelasi silang disebut juga dengan konvolusi yang juga adalah sebuah konvolusi, tetapi tidak membalik kernel. Persamaan dan contoh operasi konvolusi :

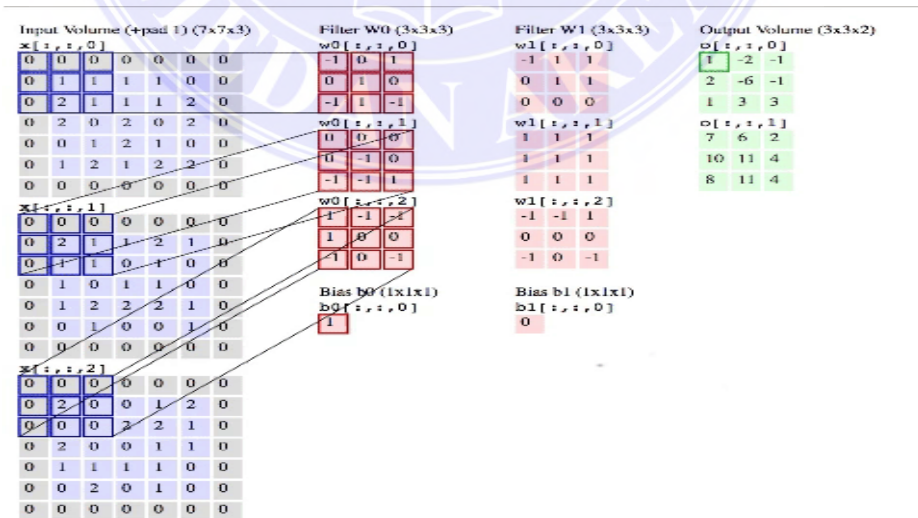
$$s(t) = (x * w)(t) \dots \dots \dots (2.1)$$

Keterangan :

s(t) = Fungsi hasil operasi konvolusi

x = Input w = bobot (kernel)

Fungsi s(t) mengeluarkan hasil tunggal yaitu feature map. Argumen pertama ialah masukan yang adalah x kemudian argumen kedua w sebagai filter atau kernel.

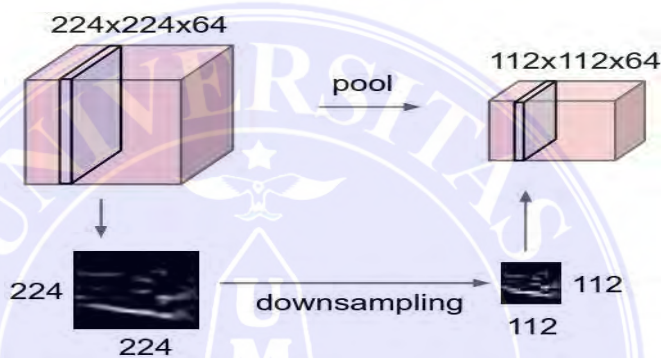


Gambar 2.2 Proses Konvolusi dengan 2 filter untuk menghasilkan Feature Map/Activation map (Nugroho, dkk., 2020)

b. *Poling Layer*

Pooling atau subsampling adalah lapisan yang berfungsi akan mempekecil nilai suatu matriks dan mengurangi informasi yang ada, Pada *max pooling* prosesnya yaitu dengan cara membagi matriks menjadi sejumlah grid yang lebih kecil, kemudian memasukan *value* tertinggi melalui masing-masing *grid*. Persaman dan contoh operasi *max pooling* dapat dilihat pada persamaan 2.2.

$$y = \max_i h, x, \dots \dots \dots (2.2)$$

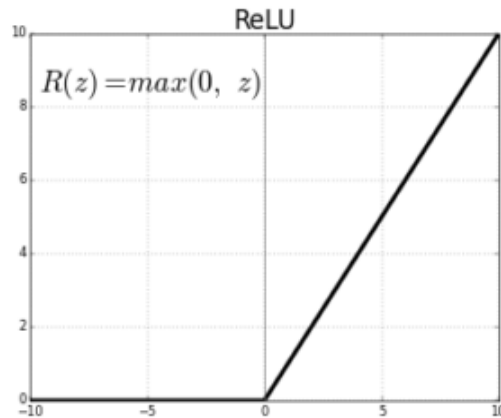


Gambar 2.3 Contoh Diagram *Max Pooling Layer* (Santoso & Ariyanto, 2018)

c. *Rectification Linear Unit (ReLU) layer*

ReLU (*Rectification Linear Unit*) adalah cara untuk mengenali non-linearitas juga menambah dari kinerja dari suatu model. Rumus dan contoh operasi ReLU dapat dilihat pada persamaan 2.2 .

$$f(x) = \max(0, x) \dots \dots \dots (2.3)$$

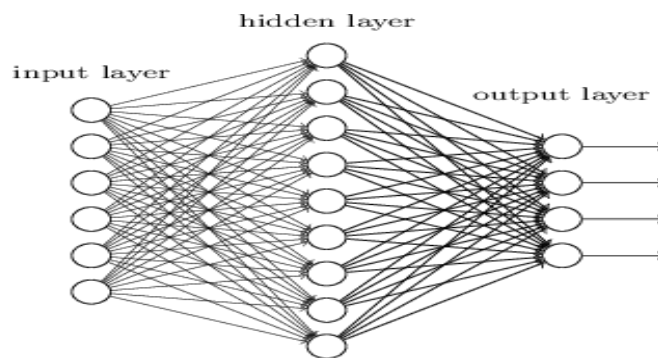


Gambar 2.4 Fungsi Aktivitas ReLU (Maysanjaya, 2020)

d. *Fully Connected Layer*

Pada layer ini ialah kesatuan dari output dari proses konvolusi. Pada tahap ini sejumlah neuron dari layer sebelumnya terhubung dengan neuron lainnya sehingga menjadi seperti jaringan syaraf tiruan. *Layer* ini menerima masukan dari keluaran pada proses sebelumnya untuk mempunyai fitur mana yang paling cocok hubungannya pada kelas tertentu. Guna layer ini ialah untuk mengumpulkan keseluruhan yang sebelumnya node dua dimensi dijadikan satu dimensi (*vector*). Proses penggabungan semua node itu dapat disebut juga dengan *flatten*. Rumus persamaan dan ilustrasi dari *fully connected layer* dapat dilihat di bawah ini.

$$y = \sum x_i w_{ib} \dots \dots \dots (2.4)$$

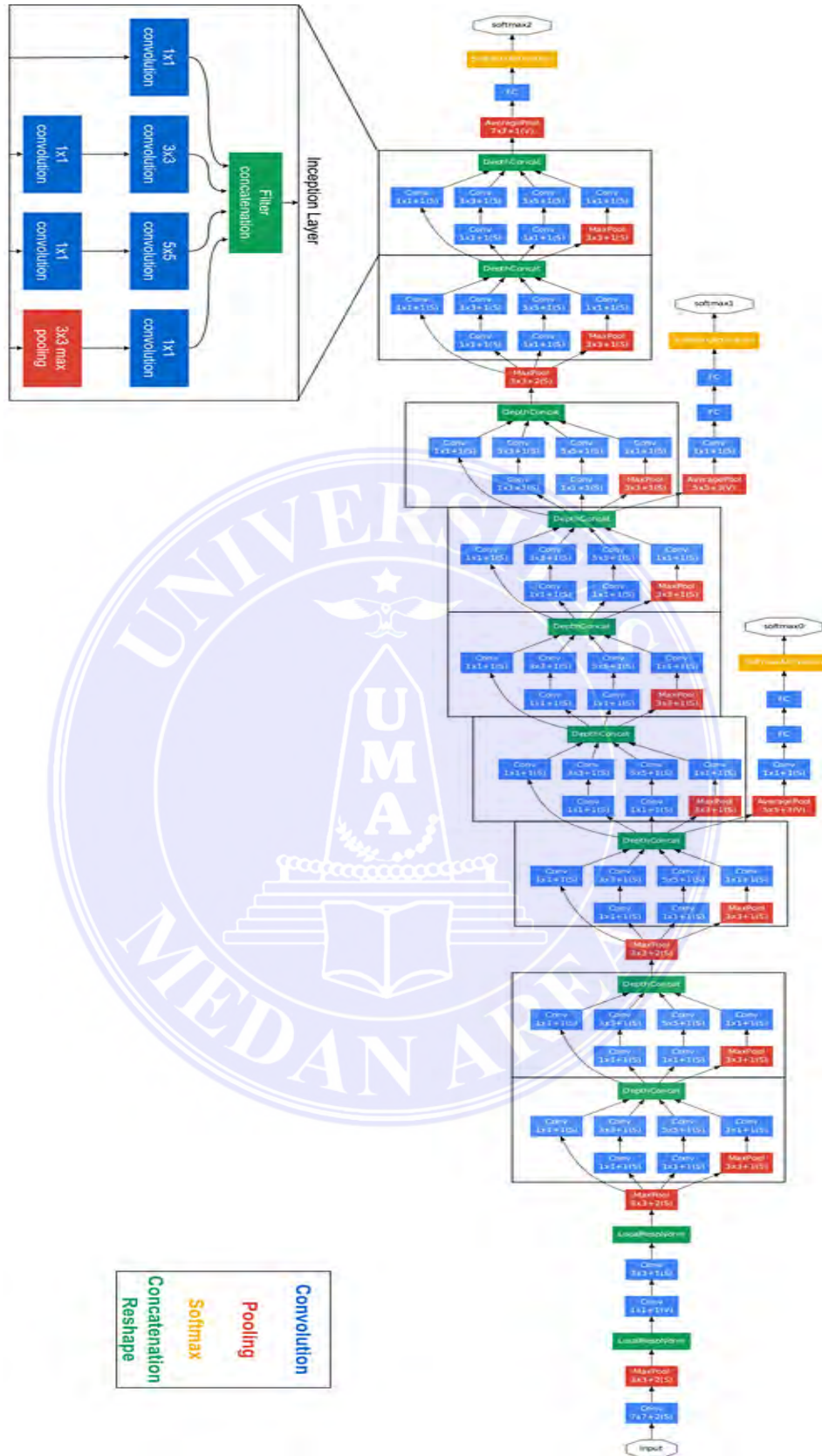


Gambar 2.1 Contoh *Full Connected Layer* (Nugroho, dkk., 2020)

1.2. *GoogleNet*

GoogLeNet atau disebut *InceptionVI* adalah model CNN yang dibuat oleh Google dan merupakan jaringan *Deep* pertama yang tersedia pada tahun 2014 yang berfokus pada efisiensi dan pemrosesan yang lebih cepat atau meminimalkan ukuran jaringan yang meningkat secara seragam dan meningkatkan sumber daya komputasi secara proporsional juga merupakan maksud dari *GoogLeNet* (Haripriya, 2019).

Arsitektur *GoogLeNet* yang diusulkan oleh Szegedy dkk., dibangun dari 22 lapisan termasuk 9 modul awal. Arsitektur ini juga menjadi pemenang di ajang kompetisi ILSVRC 2014 (Zeng dkk., 2016), (AL-Huseiny & Sajit, 2021). Salah satu kelebihan dari arsitektur *GoogLeNet* sanggup mengurangi tingkat kesalahan sampai 6,67% yang lebih unggul dari *alexnet* hanya 15,3%. Arsitektur *googlenet* memakai normalisasi batch yang digunakan agar mengatur input yang dilakukan beruntun, dan distorsi citra digunakan untuk beradaptasi ukuran sesuai keinginan. Kemudian, *Googlenet* melalui fase inception awalnya, nantinya ketika pada layer kelima, beberapa fitur akan menjalankan *fully connected neuron* (Hafifah dkk., 2021). Berikut merupakan gambar arsitektur *googlenet* dapat dilihat pada gambar 2.5.



Gambar 2.5 Arsitektur *GoogleNet*

1.3. *Tools* Pendukung

Dalam implementasi proyek menggunakan metode CNN, diperlukan bahasa pemrograman yang *stabil*, *fleksibel*, dan dilengkapi dengan sekumpulan *tools* seperti pustaka (*library*). Hal ini akan memudahkan dan mempercepat proses implementasi. Saat ini, *Python* adalah bahasa pemrograman yang memenuhi kriteria tersebut. Untuk mengimplementasikan metode CNN, terdapat beberapa pustaka *Python* yang sering digunakan, seperti *Tensorflow*, *Keras*, *Sklearn*, *Numpy*, *Pandas*, dan *Matplotlib*.

1.3.1. *Python*

Python adalah bahasa pemrograman yang populer dan sering digunakan oleh berbagai kalangan, seperti akademisi dan masyarakat umum, untuk berbagai tujuan, seperti membuat aplikasi *desktop*, *website*, *game*, melakukan analisis, serta membangun proyek kecerdasan buatan, termasuk metode CNN. Ada banyak *library* dan *library-library* ini memiliki fitur yang memadai untuk mempermudah proses implementasi dan pengembangan proyek CNN (Allaam & Wibowo, 2021), yaitu sebagai berikut:

1. *Tensorflow*

Tensorflow adalah *platform open source end-to-end* yang dibuat dan dikembangkan oleh *Google Brain* untuk mendukung *Machine Learning*. *Tensorflow* memiliki alat, pustaka, dan sumber daya yang komprehensif dan *fleksibel* yang memungkinkan penelitian menggunakan *Machine Learning* yang mutakhir dan *developer* dengan mudah membangun dan menerapkan aplikasi yang didukung

Machine Learning. *Tensorflow* dapat digunakan dengan bahasa pemrograman *Python* yang stabil. *Tensorflow* menawarkan pemrograman yang melakukan berbagai macam tugas ML/DL (regresi dan klasifikasi). *Tensorflow* yang digunakan pada penelitian ini adalah *Tensorflow* versi 2.2.0 dan dijalankan pada *Google Colaboratory*.

2. *Keras*

Keras adalah sebuah API (*Application Programming Interface*) tingkat tinggi dari *Tensorflow* yang dibuat oleh François Chollet, seorang engineer di *Google*. *Keras* dirancang agar mudah dipelajari dan digunakan, terutama bagi para peneliti, sehingga mereka dapat fokus pada eksperimen dan bukan pada penggunaan *library*. *Keras* dapat digunakan untuk membangun CNN, seperti melakukan *one-hot-encoding* pada label citra, membuat arsitektur CNN, menggunakan arsitektur CNN populer yang tersedia, melakukan plotting model CNN, memuat dataset citra ke dalam arsitektur CNN, serta melakukan training dan testing model CNN untuk menghasilkan evaluasi seperti akurasi dan loss. Selain itu, *Keras* juga dapat digunakan untuk menyimpan dan mengakses model CNN dalam format h5.

3. *Sklearn*

Sebuah tools untuk *Machine Learning* yang digunakan untuk berbagai model statistik seperti *classification*, *regression*, *clustering*, dan *dimensionality reduction*. Dalam implementasi CNN, *Sklearn* dapat digunakan sebagai alat untuk menerapkan teknik *K-Fold Cross*

Validation pada dataset, membuat *Confusion Matrix*, dan *Classification Report*.

4. *Numpy*

Sebuah *library* yang berfungsi untuk memanipulasi dan mengolah array serta matriks. Salah satu contoh penggunaannya dalam CNN adalah dengan melakukan reshape pada dimensi citra, menghitung rata-rata dan standar deviasi akurasi model CNN, dan melakukan pengolahan data berbasis s dan matriks lainnya.

5. *Pandas*

Berguna untuk melakukan manipulasi dan analisis data. Dalam konteks penggunaan pada CNN, contohnya adalah menyimpan riwayat pelatihan model (dalam bentuk akurasi dan loss) ke dalam sebuah tabel CSV.

6. *Matplotlib*

Digunakan untuk membuat grafik dan visualisasi data. Contoh penggunaannya pada CNN adalah memplot dataset beserta labelnya dan memplot hasil pelatihan dan pengujian model CNN ke dalam grafik untuk memudahkan pemahaman dan analisis.

2.3.2. Google Colaboratory

GoogleColaboratory atau *Google Colab* adalah sebuah platform gratis berbasis cloud yang menyediakan lingkungan pengembangan *Jupyter Notebook* (*Python*). *Colab* memungkinkan pengguna untuk melatih dan menguji model *Machine Learning* dan *Deep Learning* di CPU, GPU, dan TPU tanpa memerlukan konfigurasi apapun. Ada 3 keunggulan utama *Colab* yaitu tidak memerlukan konfigurasi, akses gratis ke GPU, dan mudah dibagikan dengan rekan penelitian. *Colab* hanya memerlukan akun *Google* dan *internet browser*, serta menyediakan waktu penggunaan *notebook* maksimal 12 jam/hari dengan RAM 12GB dan disk 100GB untuk opsi gratis/tanpa berbayar. (Allaam & Wibowo, 2021). Adapun tampilan dari *Google Colab* dapat dilihat pada Gambar 2.3.2.1.



Gambar 2.6 Tampilan *Google Colab*

Gambar 2.3.2.1 menunjukkan bahwa tampilan *Google Colab* mirip dengan *Jupyter Notebook*. Salah satu keuntungan penggunaan *Google Colab* adalah tidak perlu melakukan konfigurasi karena menggunakan teknologi *cloud computing*, memberikan akses gratis untuk menggunakan mesin dengan kecepatan tinggi

seperti GPU, dan mudah terhubung dengan *Google Drive*. Dalam penelitian ini, *Google Colab* digunakan untuk melatih dan menguji model *Convolutional Neural Network* dengan arsitektur *GoogLeNet* dalam melakukan klasifikasi hama serangga pada perkebunan kelapa sawit.

1.4. Tanaman Kelapa Sawit

Kelapa sawit (*Elaeis guineensis Jacq*) merupakan tumbuhan tropis golongan palma yang termasuk tanaman tahunan dan habitat aslinya adalah daerah semak belukar. Pada tanaman kelapa sawit terdapat buah pada kelapa sawit, ada tiga jenis pada ketebalan cangkang buah kelapa sawit, yakni *Dura*, *Pisifera*, dan *Tenera*. (Nanda dkk., 2022). Kelapa sawit juga merupakan sumber utama minyak nabati yang memberikan kontribusi besar minyak nabati dalam perdagangan dunia. Sejarah asal usul dan evolusinya menarik interpretasi yang membuat bingung berdasarkan literatur yang berakhir dengan pandangan dan konsep yang berlawanan. Tanaman kelapa sawit sendiri berasal dari afrika dan amerika (Babu dkk., 2021). Tanaman kelapa sawit memiliki produktivitas tertinggi di antara biji minyak utama tanaman, mencapai 3-8 kali lebih banyak minyak per area daripada tanaman lainnya (Qaim dkk., 2020). Pada 2021/2022, kira-kira 82 juta metrik ton minyak sawit dan minyak inti sawit dikonsumsi diseluruh dunia, menjadikan kelapa sawit sebagai sumber minyak nabati konsumsi nomor satu di dunia (Bittencourt dkk., 2022).

1.5. Hama Serangga Pada Kelapa Sawit

Hama merupakan organisme pengganggu tanaman (OPT) yang memiliki dampak *negative* pada produktifitas tanaman yang menjadi masalah bagi para petani karena dapat menurunkan hasil panen (Zarliani dkk., 2020),(District dkk., 2022). organisme pengganggu tanaman secara garis besar terbagi 3 yaitu Hama, Penyakit, dan Gulma (Setyaningrum & Prasetyo, 2018).Terdapat beberapa penyebab kegagalan dalam budidaya tanaman kelapa sawit yaitu, pemeliharaan tanaman, pemupukan tanaman, dan pengendalian serangan hama tanaman (Nurhasnita dkk., 2020).

Pada tanaman kelapa sawit sangat penting bagi petani untuk produksi panen. Banyak para petani kelapa sawit mengalami penurunan hasil panen yang disebabkan oleh OPT (Organisasi Pengganggu Tanaman). Salah satu hama serangga yang sering menyerang tanaman kelapa sawit adalah Hama Ulat Pemakan Daun Kelapa Sawit (UPDKS).

2.4.1. Ulat Api

Ulat api merupakan hamautama serangga pada tanaman kelapa sawit. Hama ulat api juga merupakan musuh bagi para petani. Dampak dari serangan hama ulat api pada tanamana kelapa sawit membuat para petani tepuk dahi dan menghelakan nafas, artinya serangan dari hama ini sangat berbahaya. Apa dampak dari serangan hama ini diantaranya, menurunkan hasil panen dan bahkan dalam pengendalian hama ini juga dapat menguras dompet para petani. Karena serangan hama ini dapat kehilangan daun pada kelapa sawit mencapai 70-90 % (Pratama, 2021).

2.4.2. Ulat Kantong

Ulat kantong merupakan hama penting yang paling sering muncul pada perkebunan kelapa sawit dengan kerusakan dari hama ini dapat menimbulkan penyusutan produksi sampai 40% (Saragih & Afrianti, 2021)(Yuliani dkk., 2019). Minimnya pengetahuan para petani tentang pengendalian hama pada tanaman tersebut membuat para petani mengalami kerugian. Hal ini juga perlu di ketahui bagi para petani hama apa saja yang sering menyerang tanaman para petani.

1.6. Penelitian Terkait

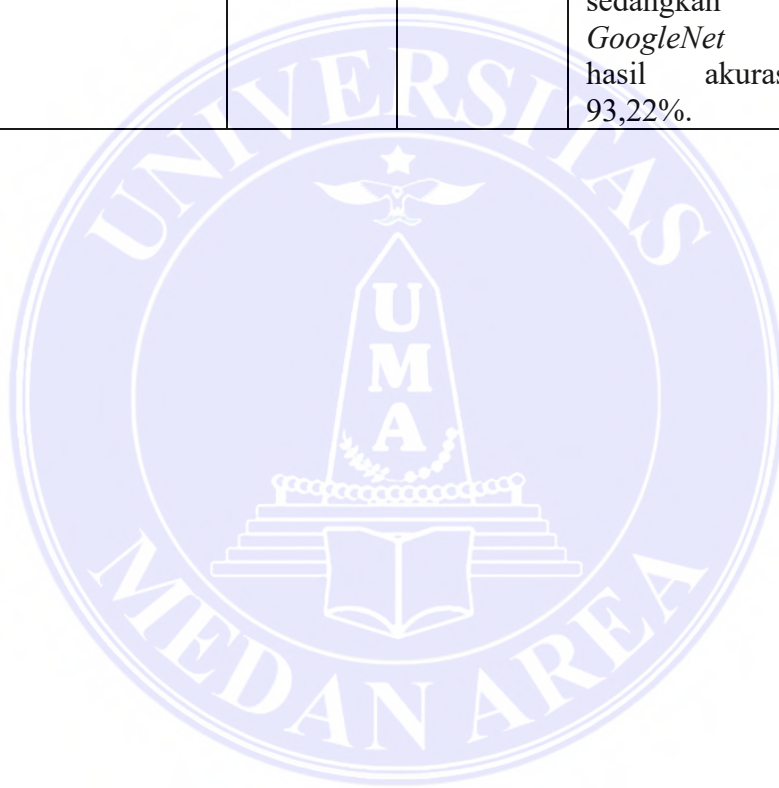
Dalam melakukan penelitian ini tidak terluput dari yang namanya acuan penelitian terdahulu sehingga dapat diketahui kontribusi penelitian yang dilakukan terhadap ilmu pengetahuan. Berikut merupakan tabel yang menjelaskan penelitian terdahulu yang berkaitan dengan penelitian ini.

Tabel 2.1 Penelitian Terkait

No	Judul	Penulis & Tahun	Metode	Keterangan
1.	Identifikasi Hama Kelapa Sawit menggunakan Metode <i>CertaintyFactor</i>	(J.Widians, &F.Riskyani, 2020)	<i>CertaintyFactor</i>	Identifikasi Hama Kelapa Sawit menggunakan Metode <i>CertaintyFactor</i> dengan perhitungannya yaitu berupa nilai persentase keyakinan dan pengguna dapat langsung mengetahui jenis hama yang teridentifikasi dan cara pengendaliannya. Serangan hama terbesar adalah hama rayap <i>Coptotermescurvignathus</i> sebesar 88,8% pada tanaman kelapa sawit
2.	Klasifikasi Citra Daun	(A.Yuliani, A.Labellapa	<i>K-NearestNeig</i>	Mengklasifikasi dengan algoritmaK-

	Kelapa Sawit Yang Terkena Dampak Hama Menggunakan Metode <i>K-Nearest Neighbor</i>	nsa, A.Yulianti, 2019)	<i>hbor</i>	<i>NearestNeighbor</i> dengan mengembangkan suatu aplikasi citra daun kelapa sawit yang terkena dampak hama ulat kantong, ulat api, dan daun normal mendapatkan hasil dengan nilai akurasi 83,3 %
3.	<i>Transfer learning with GoogLeNet for detection of lung cancer</i>	(M.AI-Huseiny, A.Sajit, 2021)	Arsitektur <i>GoogLeNet</i>	penggunaan jaringan saraf dalam (DNN) untuk mendeteksi gambar dengan nodul ganas paru-paru <i>computedtomography</i> (CT) dengan menggunakan dataset kasar yang belum diproses, dataset kanker paru-paru IQ-OTH/NCCD digunakan untuk melatih/memvalidasi algoritma <i>GoogLeNet</i> mendapatkan hasil akurasi sebesar 94,38%. yang mengungguli metode benchmark yang sebelumnya digunakan dengan dataset ini.
4.	Perbandingan Arsitektur Lenet Dan <i>GoogLeNet</i> Dalam Klasifikasi <i>Diabetic Retinopathy</i> Pada Citra Retina Fundus	(arif faizin, moh.lutfi, achmyatari, 2022)	Arsitektur <i>Lenet</i> Dan <i>GoogLeNet</i>	membandingkan arsitektur <i>Lenet</i> dan <i>GoogLeNet</i> dalam mengklasifikasi <i>DiabeticRetinopathy</i> Pada Citra <i>Retina</i> Fundus dengan menggunakan dataset dari <i>kaggle</i> sebanyak 2.500 data citra dengan hasil lebih baik arsitektur <i>Lenet</i> yang mendapatkan akurasi sebesar 95,40% dengan jumlah <i>batchsize</i> 16 dan <i>epoch</i> sebanyak 100 sedangkan dengan menggunakan arsitektur <i>GoogLeNet</i> hanya mendapatkan akurasi sebesar 84,16 % dengan jumlah <i>batchsize</i> 4 dan <i>epoch</i> sebanyak 300
5.	Implementasi Arsitektur	(Ifantiska.D, 2022)	Arsitektur <i>GoogLeNet</i>	Mengimplemenasikan dan membandingkan mana yang

	<p><i>GoogleNet</i> dan <i>Xception</i> Untuk Identifikasi Penyakit Pada Daun Tanaman Kelapa Sawit</p>		<p>dan <i>Xception</i></p>	<p>lebih baik antara arsitektur <i>GoogleNet</i> dengan <i>Xception</i> untuk idrntifikasi penyakit pada daun tanaman kelapa sawit, dengan jumlah dataset sebanyak 1230 gambar, yang terdiri dari gambar daun segar, daun yang terkena ulat api, dan yang terkena ulat kantong. Hasil dari arsitektur <i>Xception</i> mendapatkan akurasi sebesar 99,72 % sedangkan arsitektur <i>GoogleNet</i> mendapatkan hasil akurasi sebesar 93,22%.</p>
--	--	--	----------------------------	---

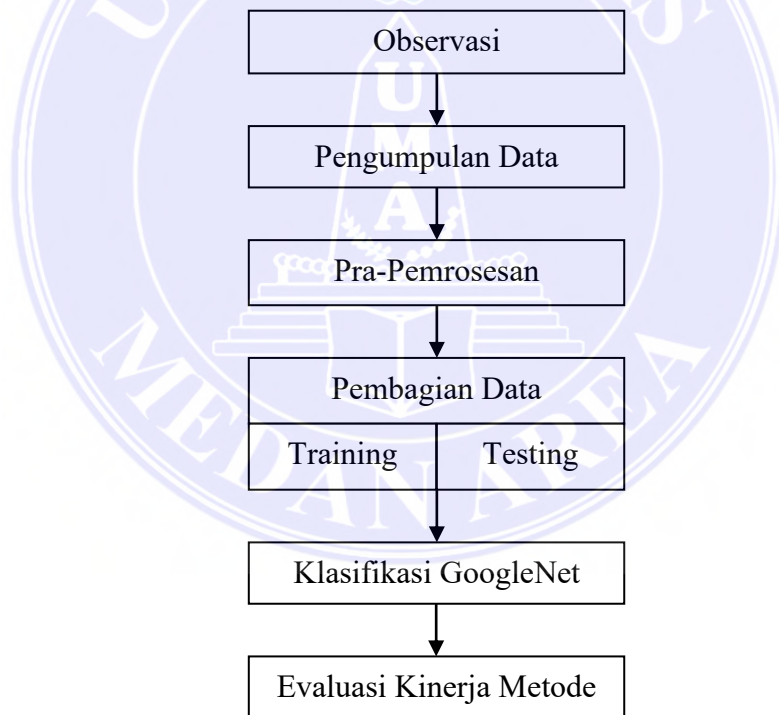


BAB III

METODE PENELITIAN

3.1. Langkah Penelitian

Dalam penelitian yang dilakukan membutuhkan alur ataupun tahapan yang jelas dalam setiap prosesnya. Langkah penelitian sendiri berfungsi sebagai panduan untuk melakukan penelitian. Alur dari setiap prosesnya juga direncanakan dan digunakan sebagai pedoman untuk melakukan penelitian dari awal hingga akhir. Langkah penelitian ini sendiri bertujuan agar penelitian berjalan sesuai rencana. Berikut adalah langka-langkah investigasinya.



Gambar 3.1 Diagram Langkah Penelitian


Pada gambar 3.1 pertama observasi persiapan data yang dikumpulkan menjadi *dataset*. setelah semua terkumpul, data dikelompokkan sesuai *class* nya masing-masing. Dilanjutkan dengan pra-pemrosesan data gambar dengan


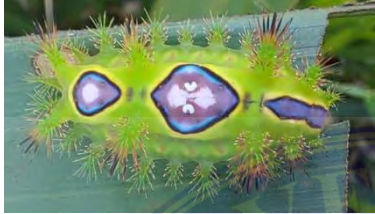
mencropping data kemudian mengcompresskan ukuran gambar agak tidak terlalu besar dan menjadi sama rata, kemudian meresize data dan mengaugmentasi data gambar. Selanjutnya pada pembagian data di bagimenjadi 2 yaitu data training dan data testing. Pada data training dilakukandengantujuan agar computer dapat mengenali objek.Selanjutnya menggunakan hyper parameter dengan hasil model yang terbaik sehingga mendapatkan klasifikasi pada data training, selanjutnya pada tahap evaluasi dilakukan dengan mencoba menghitung atau mengukur tingkat keberhasilan jika pola mirip atau mendekati pola training maka *output* dari klasifikasi yaitu hasil akurasi asrsitektur. Sedangkan data *testing* langsung pada tahap klasifikasi.

3.2. Teknik Pengumpulan Data

Dataset diambil dengan melakukan pengamatan secara langsung dengan mengumpulkan citra serangga ulat api dan ulat kantong pada Perkebunan Kelapa Sawit Afd IV Kebun Ajamu I PTPN IV , Kecamatan Panai hulu, Kabupaten Labuhanbatu, Provinsi Sumatera Utara. Serta berdiskusi dengan selaku Asisten / Mandor bahkan karyawan pada perkebunan ajamu tersebut.

Tabel 3.1 Deskripsi Citra dan Jumlah Data Citra

CITRA HAMA UPDKS	JUMLAH DATA CITRA
 <p data-bbox="395 1877 777 1917">Ulat Kantong (<i>Metisa Plana</i>)</p>	<p data-bbox="1043 1742 1106 1783">1002</p>

 <p>Ulat Api (<i>Setora Nitens</i>)</p>	<p>1024</p>
 <p>Ulat Api (<i>Setothsea Asigna</i>)</p>	<p>1071</p>

3.3. Pra-Pemrosesan

Pada tahap ini dilakukan *cropping* data gambar yang terlalu jauh atau objek terlalu kecil secara otomatis dengan menggunakan *microsoft office picture manager 2010*. Setelah selesai *cropping* semua data, proses selanjutnya yaitu *resize* citra untuk mengecilkan ukuran *pixel* secara merata menggunakan bantuan dari aplikasi *Caesium Image Compressor*. Kemudian setelah selesai *resize* yaitu pemberian nama data (*Labeling*) data pada setiap *class*-nya. Kemudian pembagian data menjadi dua, yaitu data *training* (latih) dan data *testing* (uji), setelah diproses masuk pada tahap perancangan arsitektur *CNN* dimana dalam tahapan ini digunakan untuk melatih dalam mengenali objek. Tahap selanjutnya melakukan pengenalan model dengan menggunakan data latih sebelumnya. Kemudian dilakukan untuk menguji tingkat akurasi dari arsitektur *GooleNet*.

3.3.1 Cropping data gambar

Pada tahap cropping data gambar ini berfungsi untuk memotong gambar yang di ambil terlalu jauh ataupun gambar yang terlalu ke bawah atau ke atas dan ke samping, maka dengan mengcroping data tersebut mendapatkan hasil tata letak yang bagus dan tidak kepotong objek tersebut. Crop data gambar tersebut menggunakan aplikasi bantuan yaitu *microsoft office picture manager* 2010. Berikut adalah cara cropping data dengan masing masing classnya dengan menggunakan *aplikasi microsoft office picture manager* 2010 dapat dilihat pada gambar 3.2.



Gambar 3.2 Proses *Cropping* setiap class dengan *Microsoft Office Picture Manager*

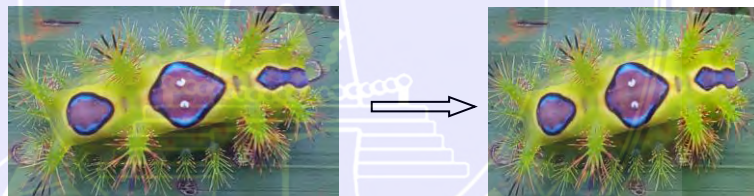
Data yang sudah ter-*crop* ternyata juga ter-*resize* namun peneliti me-*resize* data lagi agar data *pixel*-nya sama rata. Berikut adalah salah satu gambar yang belum di *crop* dan sudah di *crop* dengan menggunakan *MS.Office picture manager* dapat dilihat pada gambar 3.3.



Gambar 3.3 Proses Cropping Sesudah Dan Sebelum

3.3.2 Resizing Data

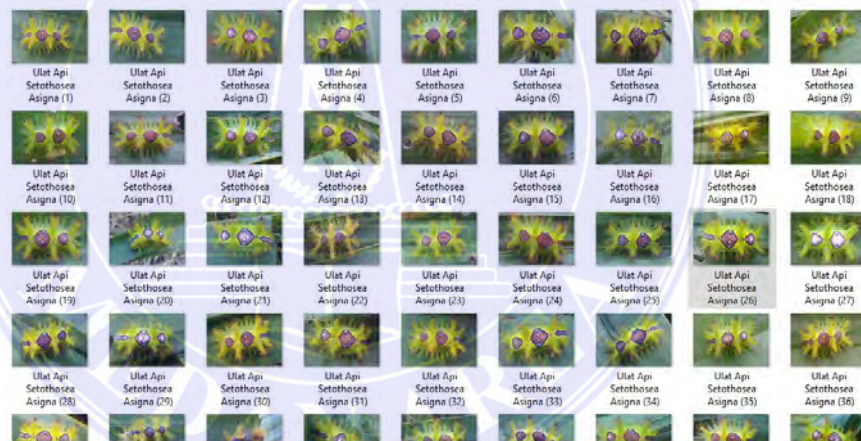
Pada tahapan ini tiap data sampel citra hama tanaman pada tanaman kelapa sawit akan diubah ukuran *pixel*-nya dengan panjang dan lebar yang sama ukurannya yaitu 300 x 300 *pixel*. Sehingga, setiap data memiliki ukuran yang sama secara merata. Berikut adalah simulsi *reaize image* sesudah dan sebelum dapat dilihat pada gambar 3.4.



Gambar 3.4 Simulasi *Resize Image* Sesudah Dan Sebelum

Setelah melakukan tahapan *resize* pada dataset, selanjutnya melakukan pemberian nama (*Labeling*) pada *File Explorer* di Laptop *Windows* dengan memilih semua *file* yang ingin diubah namanya dengan menekan *Ctrl + A* di *Keyboard*, lalu klik kanan pada daftar *file* tersebut kemudian klik *Rename*. jika sudah selesai mengetik nama *file* yang baru, tekan *Enter* di *Keyboard* maka semua nama *file* yang tadi ditandai akan otomatis berubah namanya pada setiap data citra sesuai dengan ketentuan sebagai berikut:

1. Untuk kelas “Ulat Api *Setothosea Asigna*” akan diberi label sesuai dengan jenis hama UPDKS, yaitu Ulat Api *Setothosea Asigna* (1), Ulat Api *Setothosea Asigna* (2), dan seterusnya hingga Ulat Api *Setothosea Asigna* (1071).
2. Untuk kelas “Ulat Api *Setora Nitens*” akan diberi label sesuai dengan jenis jenis hama UPDKS, Ulat Api *Setora Nitens* (1), Ulat Api *Setora Nitens* (2), Ulat Api *Setora Nitens*(1024).
3. Untuk kelas “Ulat Kantong *Metisa Plana*” akan diberi label sesuai dengan jenis hama UPDKS, yaitu “Ulat Kantong *Metisa Plana* (1), Ulat Kantong *Metisa Plana* (2), Ulat Kantong *Metisa Plana*(1002).



Gambar 3.5 Proses Pelabelan Data

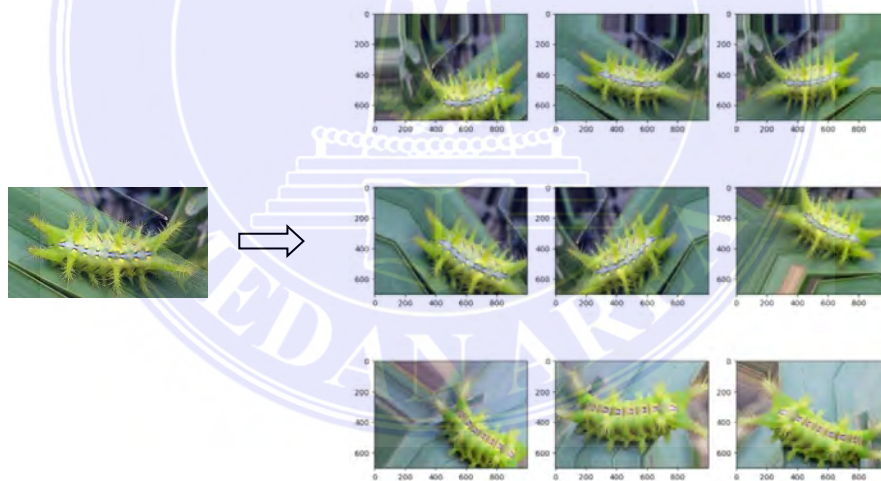
3.3.3 *Augmentasi* Dataset

Pada tahapan ini melakukan modifikasi atau manipulasi suatu citra, sehingga citra asli dalam bentuk standar akan diubah bentuk dan posisinya guna membantu untuk memperbanyak sumber data atau dataset serta mengurangi kemungkinan terjadinya overfitting. Fungsi dari *augmentasi* data akan memutar secara acak, dengan mengubah tinggi dan lebar, serta membalik gambar secara

horizontal.

Berikut hasil *Augmentasi* dataset pada ketiga kelas yaitu Ulat Api *Setora Nitens*, Ulat Kantong *Metisa Plana*, dan Ulat Api *SetothoseaAsigna* sebagai berikut :

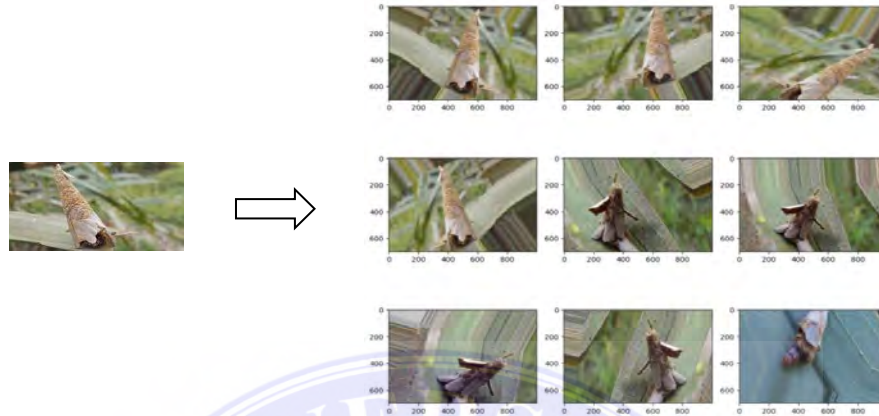
1. Hasil *Augmentasi* Dataset pada kelas Ulat Api *SetoraNitens* dengan *rotation* 20 derajat menggunakan fungsi *rotation_range=20*, pergeseran lebar dengan fungsi *width_shif_range=0.2*, pergeseran tinggi dengan fungsi *height_shif_range=0.2*, hasil *augmentasi* dengan fungsi *shear_range=0.2*, kemudian pembesaran (*zoom*) dengan fungsi *zoom_range=0.2*, dan pencerminan (*flipping*) secara *horizontal* sebesar 180 derajat dengan fungsi *horizontal_flip=True*.



Gambar 3.6 Hasil *Augmentasi* Dataset Pada Ulat Api *SetoraNitens*

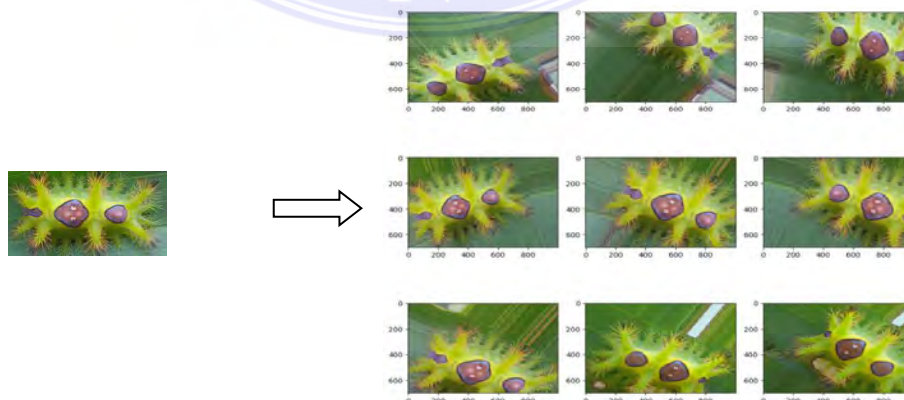
2. Hasil *Augmentasi* Dataset pada kelas Ulat Kantong *MetisaPlana* dengan *rotation* 20 derajat menggunakan fungsi *rotation_range=20*, pergeseran lebar dengan fungsi *width_shif_range=0.2*, pergeseran tinggi dengan fungsi *height_shif_range=0.2*, hasil *augmentasi* dengan fungsi *shear_range=0.2*, kemudian pembesaran (*zoom*) dengan fungsi

$zoom_range=0.2$, dan pencerminan (*flipping*) secara *horizontal* sebesar 180 derajat dengan fungsi $horizontal_flip=True$.



Gambar 3.7 *Augmentasi* Dataset Pada Ulat Kantong *Metisaplana*

3. Hasil *Augmentasi* Dataset pada kelas Ulat Api *Setothosea Asigna* dengan *rotation* 20 derajat menggunakan fungsi $rotation_range=20$, pergeseran lebar dengan fungsi $width_shif_range=0.2$, pergeseran tinggi dengan fungsi $height_shif_range=0.2$, hasil *augmentasi* dengan fungsi $shear_range=0.2$, kemudian pembesaran (*zoom*) dengan fungsi $zoom_range=0.2$, dan pencerminan (*flipping*) secara *horizontal* sebesar 180 derajat dengan fungsi $horizontal_flip=True$.



Gambar 3.8 Hasil *Augmentasi* Dataset Pada Ulat Api *Setothosea Asigna*

3.3.4 Data Training

Pada Tahap selanjutnya adalah melakukan training dataset dengan menggunakan arsitektur *Googlenet*. Metode CNN termasuk dalam metode supervised learning yaitu identifikasi suatu citra dengan training data citra yang telah ada. Pada proses konvolusi ini akan digunakan perkalian matriks terhadap filter dan area pada citra daun. Sebelum itu, Perlu dibuat inisialisasi hyper parameter pelatihan. Paramater yang digunakan di inisialisasi untuk proses training ini adalah *Learning rate*, *Batch size*, *Epoch*, dan *Optimizer*.

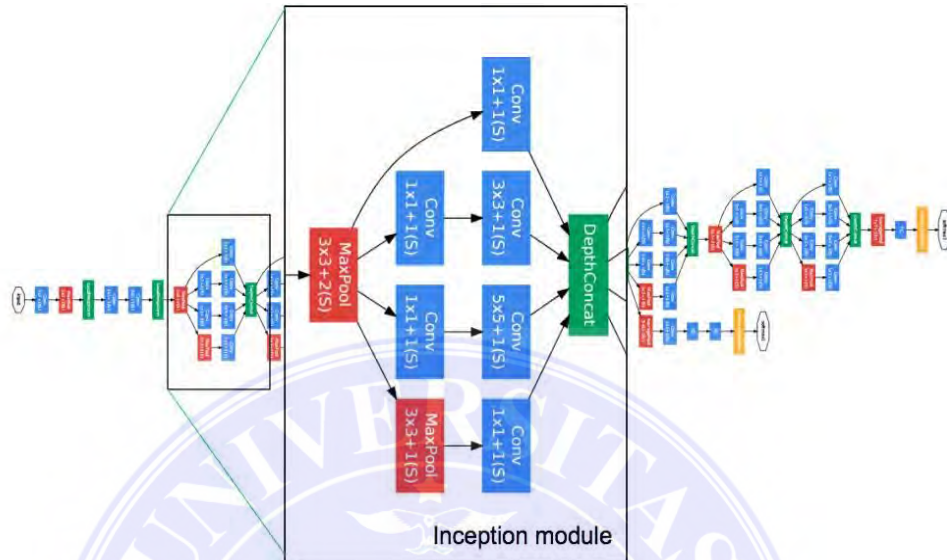
3.3.5 Data Testing

Tahap data testing dimulai dengan memproses input citra yang akan diuji, Kemudian sistem akan mengidentifikasi model yang sudah dilatih sebelumnya. nantinya citra yang diuji akan diproses menggunakan arsitektur *GoogleNet*. Setelah itu, sistem akan menampilkan akurasi berupa prediksi jenis penyakit pada daun yang diinput.

3.4. Pemodelan *GoogleNet*

GoogleNet mempunyai karakteristik yang menggunakan jaringan *inception* yang lebih luas, yaitu berfungsi untuk meningkatkan akurasi. *GoogleNet* (*Inception V1*) yang merupakan sebuah arsitektur dan model dari arsitektur *CNN*. *GoogleNet* menggunakan normalisasi *batch* untuk mengelola masukan/*input* secara bertahap, dan distorsi gambar untuk penyesuaian ukuran sesuai kebutuhan. Selanjutnya *GoogleNet* melewati tahap *Inception* dahulu, lalu kemudian pada layer ke lima, beberapa fitur akan melakukan *fully connected neuron*, kemudian

fitur yang lain akan dilanjutkan pada proses ekstraksi dengan *convolution*.
(Sabilla, 2020)



Gambar 3.9 Modul *Inception* pada *GoogleNet*

Berdasarkan gambar modul diatas terlihat ada banyak *layer* dan *convolution* yang dimiliki oleh *GoogleNet*, yaitu dengan memiliki 5 *layer* sampai 13 *layer*. *GoogleNet* disusun secara linier yang memiliki 9 *Inception* modul. Selain itu *GoogleNet* juga memiliki 22 *layer*, dimana menggunakan *Average pooling global* di akhir *Inception*.

3.4.1. Perhitungan Manual CNN dalam Mengolah Gambar



Gambar 3.10 Ulat api *Setora Nitens*

Gambar diatas merupakan data sampel dalam mengelola gambar menggunakan algoritma CNN. Berikut adalah proses konvolusi dengan

memberikan nilai filter pada matrix. Pixel citra dengan 3 channel Red, Green, dan Blue, kemudian diambil pixelnya dan ditambahkan padding same (0) disetiap pixel. Dapat dilihat pada gambar dibawah ini.

Red					
0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

Green					
0	0	0	0	0	0
0	114	112	117	127	0
0	113	112	114	121	0
0	111	113	112	113	0
0	112	110	110	107	0
0	0	0	0	0	0

Blue					
0	0	0	0	0	0
0	106	105	111	121	0
0	105	104	106	114	0
0	103	104	104	106	0
0	103	101	102	99	0
0	0	0	0	0	0

Gambar 3.11 Pixel citra dengan 3 channel

Selanjutnya tahapan convolutional layer menggunakan kernel 3x3 dengan nilai random dapat dilihat pada gambar berikut :

1	0	-1
1	0	-1
1	0	-1

Langkah selanjutnya melakukan proses perhitungan di setiap channel terhadap kernel 3x3. Tahapan ini dilakukan secara berulang dengan pergeseran kernel sebanyak 1 stride disetiap channel. Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai stride adalah 1, maka conv. filter akan bergeser sebanyak 1 pixel secara horizontal lalu vertikal.

Posisi 1 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*0) + ((-1)*0) +$$

$$(1*0) + (0*127) + ((-1)*124) +$$

$$(1*0) + (1*126) + ((-1)*124) = - 122$$

Posisi 2 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*0) + ((-1)*0) +$$

$$(1*127) + (0*124) + ((-1)*126) +$$

$$(1*126) + (1*124) + ((-1)*124) = 127$$

Posisi 3 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*0) + ((-1)*0) +$$

$$(1*124) + (0*126) + ((-1)*132) +$$

$$(1*124) + (1*124) + ((-1)*127) = 113$$

Posisi 4 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*0) + ((-1)*0) +$$

$$(1*126) + (0*132) + ((-1)*0) +$$

$$(1*124) + (1*127) + ((-1)*0) = 377$$

Posisi 5 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*127) + ((-1)*124) +$$

$$(1*0) + (0*126) + ((-1)*124) +$$

$$(1*0) + (1*124) + ((-1)*126) = - 250$$

Posisi 6 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*127) + (0*124) + ((-1)*126) +$$

$$(1*126) + (0*124) + ((-1)*124) +$$

$$(1*124) + (1*126) + ((-1)*125) = 128$$

Posisi 7 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*124) - (0*126) + ((-1)*132) -$$

$$(1*124) + (0*124) + ((-1)*127) -$$

$$(1*126) + (1*125) + ((-1)*122) = 118$$

Posisi 8 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*126) + (0*132) + ((-1)*0) +$$

$$(1*124) + (0*127) + ((-1)*0) +$$

$$(1*125) + (1*122) + ((-1)*0) = 497$$

Posisi 9 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*126) + ((-1)*124) +$$

$$(1*0) + (0*124) + ((-1)*126) +$$

$$(1*0) + (1*126) + ((-1)*124) = -248$$

Posisi 10 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*126) + (0*124) + ((-1)*124) -$$

$$(1*124) + (0*126) + ((-1)*125) -$$

$$(1*126) - (1*124) + ((-1)*124) = 127$$

Posisi 11 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*124) + (0*124) + ((-1)*127) -$$

$$(1*126) + (0*125) + ((-1)*122) -$$

$$(1*124) + (1*124) + ((-1)*120) = 129$$

Posisi 12 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*124) + (0*127) + ((-1)*0) +$$

$$(1*125) + (0*122) + ((-1)*0) +$$

$$(1*124) + (1*120) + ((-1)*0) = 493$$

Posisi 13 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*0) + (0*124) + ((-1)*126) +$$

$$(1*0) + (0*126) + ((-1)*124) +$$

$$(1*0) + (1*0) + ((-1)*0) = -250$$

Posisi 14 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*124) + (0*126) + ((-1)*125) +$$

$$(1*126) + (0*124) + ((-1)*124) +$$

$$(1*0) - (1*0) + ((-1)*0) = 1$$

Posisi 15 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*126) - (0*125) + ((-1)*122) -$$

$$(1*124) - (0*124) + ((-1)*120) +$$

$$(1*0) - (1*0) + ((-1)*0) = 8$$

Posisi 16 di channel Red :

0	0	0	0	0	0
0	127	124	126	132	0
0	126	124	124	127	0
0	124	126	125	122	0
0	126	124	124	120	0
0	0	0	0	0	0

$$(1*125) + (0*122) + ((-1)*0) +$$

$$(1*124) + (0*120) - ((-1)*0) +$$

$$(1*0) + (1*0) - ((-1)*0) = 249$$

Setelah di perhitungan channel Red maka akan membentuk matrix 4x4 dapat dilihat hasilnya seperti berikut :

-122	127	113	377
-250	128	118	497
-248	127	129	493
-250	1	8	249

Kemudian dilakukan cara yang sama terhadap channel Green dan Blue, maka akan menghasilkan matrix 4x4 sebagai berikut:

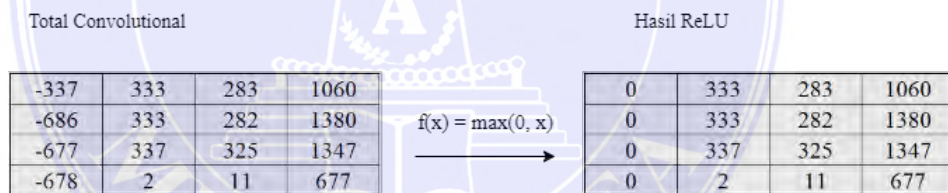
-111	108	90	352
-226	108	88	456
-223	110	104	443
-223	1	3	222

-104	98	80	331
-210	97	76	427
-206	100	92	411
-205	0	0	206

Setiap nilai dari masing – masing channel yang sudah melalui tahap perhitungan, selanjutnya akan dijumlahkan dan akan menghasilkan output seperti berikut :

-337	333	283	1060
-686	333	282	1380
-677	337	325	1347
-678	2	11	677

Tahapan selanjutnya adalah menggunakan fungsi aktivasi ReLU yang berfungsi untuk mengubah semua nilai negatif menjadi nol (0), sehingga hasilnya sebagai berikut :



Gambar 3.12 fungsi aktivasi ReLU

Gambar diatas merupakan hasil tahapan dengan menggunakan fungsi aktivasi ReLU, sehingga tahapan convolutional ini menghasilkan banyak feature map.

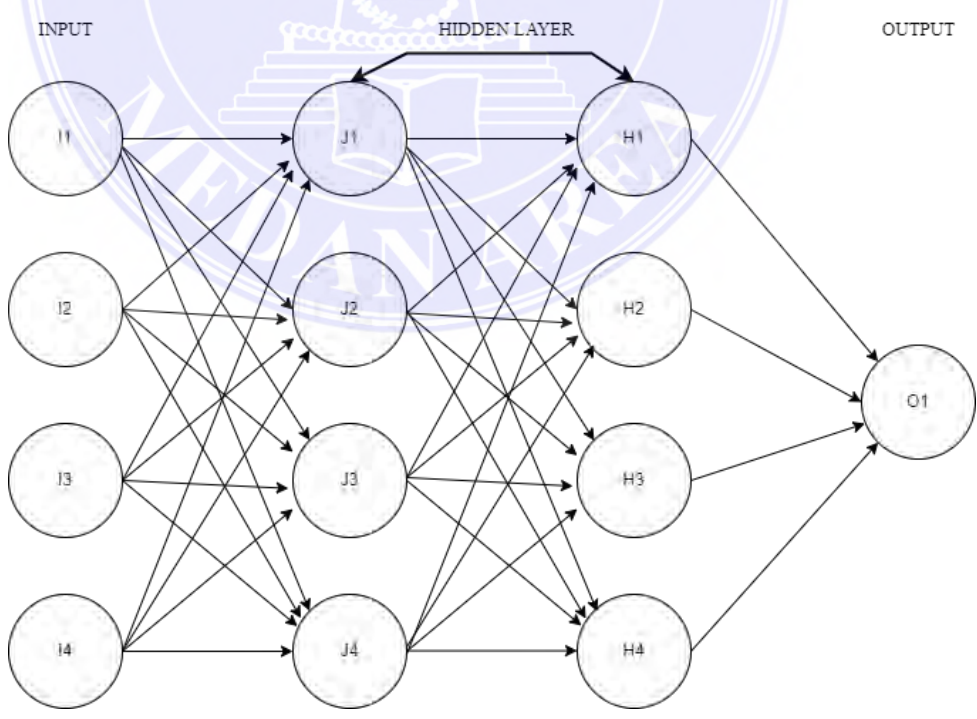
Hasil pada gambar akan dijadikan inputan pada pooling layer dengan menggunakan kernel size 2x2 denga 2 stride artinya mengambil nilai terbesar dari matrix 2x2. Dapat dilihat pada gambar berikut :

0	333	283	1060			
0	333	282	1380	→	333	
0	337	325	1347			
0	2	11	677			

0	333	283	1060			
0	333	282	1380	→	333	1380
0	337	325	1347		337	
0	2	11	677			

0	333	283	1060			
0	333	282	1380	→	333	1380
0	337	325	1347		337	1347
0	2	11	677			

Maka hasilnya akan mendapatkan nilai matrix 2x2. Tahapan flatten layer berfungsi untuk mengubah nilai matrix menjadi nilai vector untuk dijadikan nilai inputan terhadap proses selanjutnya yaitu fully connected layer. Adapun ilustrasinya dapat dilihat pada gambar berikut :



Gambar 3. 13 Fully Connected Layer.

Proses akhir dari algoritma CNN yang dibangun ini merupakan dense dengan fungsi aktivasi softmax. Untuk melakukan perhitungan di setiap hidden layer adalah sebagai berikut:

$$\sum_{i=1}^N I_i * v_{ij} = J_i$$

$$\sum_{i=1}^N J_i * w_{ij} = H_i$$

$$\sum_{i=1}^N H_i * x_{ij} = O_i$$

$$J_1 = (333 * 0,4) + (1380 * 0,4) + (337 * 0,4) + (1347 * 0,4) = 1358,8$$

$$J_2 = (333 * 0,2) + (1380 * 0,2) + (337 * 0,2) + (1347 * 0,2) = 679,4$$

$$J_3 = (333 * 0,1) + (1380 * 0,1) + (337 * 0,1) + (1347 * 0,1) = 339,7$$

$$J_4 = (333 * 0,3) + (1380 * 0,3) + (337 * 0,3) + (1347 * 0,3) = 1019,1$$

Untuk melakukan perhitungan hidden layer tersebut menggunakan inputan yang sudah melewati tahapan flatten layer. Adapun inputan dikalikan menggunakan nilai acak yang berbeda terhadap masing masing layer. Maka hasil untuk $J_1 = 1358,8$, $J_2 = 679,4$, $J_3 = 339,7$, $J_4 = 1019,1$.

Selanjutnya menghitung hidden layer kedua menggunakan bobot yang baru saja dihitung, dapat dilihat sebagai berikut :

$$H_1 = (1358,8 * 0,1) + (679,4 * 0,1) + (339,7 * 0,1) + (1019,1 * 0,1) = 339,7$$

$$H_2 = (1358,8 * 0,3) + (679,4 * 0,3) + (339,7 * 0,3) + (1019,1 * 0,3) = 1019,1$$

$$H3 = (1358,8 * 0,4) + (679,4 * 0,4) + (339,7 * 0,4) + (1019,1 * 0,4) = 1358,8$$

$$H4 = (1358,8 * 0,2) + (679,4 * 0,2) + (339,7 * 0,2) + (1019,1 * 0,2) = 679,4$$

Setiap layer dikalikan dengan nilai acak yang berbeda untuk menghasilkan H1 = 339.7, H2 = 1019.1, H3 = 1358.8, H4 = 679.4. Maka selanjutnya menghitung output menggunakan bobot baru dari hidden layer kedua dengan melakukan perkalian terhadap nilai acak.

$$O1 = (339,7 * 0,2) + (1019,1 * 0,2) + (1358,8 * 0,2) + (679,4 * 0,2) = 679,4$$

Langkah selanjutnya adalah perhitungan softmax dengan rumus sebagai berikut:

$$s(O_i) = \frac{e^{O_i}}{\sum_{j=1}^n e^{O_j}} = \frac{e^{679,4}}{e^{679,6}} = \frac{7,6}{7,6} = 1$$

Sehingga didapatkan nilai probabilitas 1 yang berarti gambar tersebut terprediksi hama.

Confusion Matrix :

tabel 3. 2 confusion matrix

PREDICTED VALUE	ACTUAL VALUE	
	POSITIVE	NEGATIF
POSITIVE	1	0
NEGATIVE	0	0

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% = \frac{1 + 0}{1 + 0 + 0 + 0} = 100\%$$

3.5. Evaluasi Model

3.5.1. Confusion Matrix

Confusion matrix adalah alat evaluasi visual yang digunakan dalam sistem klasifikasi (Xu dkk., 2020). *Confusion matrix* berguna untuk mengukur seberapa baik model klasifikasi yang dibuat (Purwono dkk., 2021). *Confusion matrix* berukuran $n \times n$, dimana n merupakan jumlah kelas yang berbeda (Markoulidakis dkk., 2021). *Confusion matrix* menentukan akurasi yang didapat dari nilai beberapa parameter, seperti *TruePositif* (TP), *FalsePositif* (FP), *TrueNegatif* (TN), dan *False Negatif*(FN) (Foeady, 2019). Tabel *confusion matrix* ditunjukkan pada tabel *Confusion matrix*.

Tabel 3.3 *ConfusionMatrix*

		Kelas Sebenarnya	
		Positif	False Positive
Prediksi	Positif	True Positive	False Positive
	Negatif	False Negative	True Negative

Pada *confusion matrix* memiliki beberapa model perhitungan untuk mencari nilai *accuracy*, *precision*, *recall*, dan *F1-score* ditunjukkan pada persamaan dibawah ini :

a) *Accuracy*

Merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(3.1)$$

b) *Precision*

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. TP

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3.2)$$

urukan data.

c) *Recall*

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(3.3)$$

d) *F1Score*

F1Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \dots\dots\dots(3.4)$$

Keterangan :

TP = *true positive*

FN = *false negative*

TN = *true negative*

FP = *false positive*

TP arti sebagai data positif yang diduga sebagai positif ,

TN memiliki arti sebagai data negatif yang di diduga sebagai negatif.,

FN ialah lawan dari TP yaitu data positif yang diduga sebagai negatif

FP, lawan dari TN, yaitu data negatif yang diduga positif.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian sistem dalam melakukan klasifikasi hama tanaman pada kelapa sawit dengan menggunakan arsitektur *GoogleNet* yang telah melakukan pengujian sistem dengan menggunakan 16 skenario adalah sebagai berikut:

1. Penerapan metode *Convulation Neural Network* dengan arsitektur *GoogleNet* menghasilkan peforma yang baik terhadap semua dataset sehingga dapat menghasilkan klasifikasi jenis hama pada tanaman kelapa sawit secara akurat
2. Berdasarkan hasil *training* pada 16 skenario model yang diuji, diperoleh model terbaik pada skenario 10 dengan tingkat akurasi 100% yang menggunakan *hyperparameter* dengan jumlah *epoch* 25, *batchsize* 32, *optimizer RMSprop*, *learning rate* 0.001,
3. Berdasarkan tingkat akurasi dari hasil *training* pada skenario 1 sampai skenario 16 mendapatkan hasil *good model* dimana *Good Model* adalah model yang dapat menjelaskan data tanpa dipengaruhi oleh data noise. Sehingga model tidak akan *fit* terhadap tiap data poin, namun mampu menjelaskan atau mengklasifikasikan kelompok data. Namun ada satu yang terjadi proses *Underfitting* pada Skenario 8 dimana *Underfitting* merupakan situasi ketika model tidak dapat melihat logika dibelakang data, yang menyebabkan model tidak dapat memprediksi data dengan tepat, baik untuk dataset training, testing, maupun validasi.

4. Hasil akurasi yang diperoleh dari model setelah diuji dengan menggunakan *data testing* (3097 citra) dengan dibagi menjadi 3 *class* yaitu didapatkan hasil akurasi sebesar 100% dengan nilai persisi (*Precision*) sebesar 100%, *recall* 100% dan *f1.score* 100%.

5.2. Saran

Saran yang dapat penulis berikan untuk penelitian lebih lanjut adalah sebagai berikut:

1. Menggunakan jumlah dataset yang beragam dan dengan jumlah yang banyak serta kualitas data citra ditingkatkan sehingga nantinya sistem dapat bekerja dengan optimal agar tidak kedapatan yang namanya *underfitting* pada hasil *training*, *testing*, maupun validasi.
2. Menggunakan spesifikasi perangkat yang lebih tinggi, khususnya *Random Access Memory* (RAM) dan *Graphics Processing Unit* (GPU) untuk mempercepat proses *training*.
3. Klasifikasi hama tanaman kelapa sawit ini dapat dikembangkan kedepannya dengan menambahkan *class*nya dan memperbanyak jenis hama pada tanaman kelapa sawit dan dengan menggunakan *optimizer* lainnya

DAFTAR PUSTAKA

- Agustina, N. A. (2021). Tingkat serangan hama ulat api. *Jurnal Ilmiah Rhizobia*, 3(1), 50–57.
- AGUSTINA, R., MAGDALENA, R., & PRATIWI, N. K. C. (2022). Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), 446. <https://doi.org/10.26760/elkomika.v10i2.446>
- Ahmad, M. N., Mohamed Shariff, A. R., Aris, I., Abdul Halin, I., & Moslim, R. (2021). IDENTIFICATION and DETERMINATION of the SPECTRAL REFLECTANCE PROPERTIES of LIVE and DEAD BAGWORMS, *Metisa plana* WALKER (Lepidoptera: Psychidae) USING VIS/ NIR SPECTROSCOPY. *Journal of Oil Palm Research*, 33(3), 425–435. <https://doi.org/10.21894/jopr.2020.0099>
- Ahmad, S. N., Mazmira, M., Masri, M., & Kamarudin, N. (2020). *Assessment on Voracity and Predation Es Es*. 32(September), 480–487.
- AL-Huseiny, M. S., & Sajit, A. S. (2021). Transfer learning with GoogLeNet for detection of lung cancer. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 1078. <https://doi.org/10.11591/ijeecs.v22.i2.pp1078-1086>
- Angga Irawan, F., Sudarma, M., & Care Khrisne, D. (2021). Rancang Bangun Aplikasi Identifikasi Penyakit Tanaman Pepaya California Berbasis Android Menggunakan Metode Cnn Model Arsitektur Squeezenet. *Jurnal SPEKTRUM*, 8(2), 18. <https://doi.org/10.24843/spektrum.2021.v08.i02.p3>
- Arif Faizin, Moh. Lutfi, & Achmyatari. (2022). Perbandingan Arsitektur Lenet Dan Googlenet Dalam Klasifikasi Diabetic Retinopathy Pada Citra Retina Fundus. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(1), 342–347. <https://doi.org/10.36040/jati.v6i1.4581>
- Arumugam, G., Karuppiah, H., Sreeramulu, B., Paulchamy, R., & Sundaram, J. (2019). Occurrence of natural lectin with bacterial agglutination property in the serum of lepidopteran pest, *Parasa lepida*. *Entomological Science*, 22(3), 239–249. <https://doi.org/10.1111/ens.12360>
- Aryanto, D., & Augusman, V. (2021). Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga. *Jurnal TIMES*, 10(1), 39–43.
- Babu, B. K., Mathur, R. K., Anitha, P., Ravichandran, G., & Bhagya, H. P. (2021). Phenomics, genomics of oil palm (*Elaeis guineensis* Jacq.): way forward for making sustainable and high yielding quality oil palm. *Physiology and Molecular Biology of Plants*, 27(3), 587–604. <https://doi.org/10.1007/s12298-021-00964-w>

- Bittencourt, C. B., Carvalho da Silva, T. L., Rodrigues Neto, J. C., Vieira, L. R., Leão, A. P., de Aquino Ribeiro, J. A., Abdelnur, P. V., de Sousa, C. A. F., & Souza, M. T. (2022). Insights from a Multi-Omics Integration (MOI) Study in Oil Palm (*Elaeis guineensis* Jacq.) Response to Abiotic Stresses: Part One—Salinity. *Plants*, *11*(13). <https://doi.org/10.3390/plants11131755>
- Dawei, W., Limiao, D., Jiangong, N., Jiyue, G., Hongfei, Z., & Zhongzhi, H. (2019). Recognition pest by image-based transfer learning. *Journal of the Science of Food and Agriculture*, *99*(10), 4524–4531. <https://doi.org/10.1002/jsfa.9689>
- Dibisono, M. Y., Nadhira, A., Wijaya, H., & Sitorus, R. (2021). UJI POTENSI EKSTRAK DAUN GAMBIR (*Uncaria gambir* Roxb) UNTUK MENGENDALIKAN HAMA ULAT KANTONG (*Metisa plana*). *Jurnal Al Ulum LPPM Universitas Al Washliyah Medan*, *9*(1), 8–14. <https://doi.org/10.47662/alulum.v9i1.135>
- District, T. B., Regency, O. I., Arsi, A., Hamidson, H., Gunawan, B., Pujiastuti, Y., Pratama, R., Mauluddin, M., Studi, P., Tanaman, P., Pertanian, F., Sriwijaya, U., Ilir, O., Selatan, S., Agroekoteknologi, P. S., Pertanian, F., Sriwijaya, U., Ilir, O., & Selatan, S. (2022). *Teknik Budidaya Petani Tanaman Karet (Hevea brasiliensis) terhadap Hama dan Penyakit di Kecamatan Tanjung Batu , Kabupaten Ogan Ilir*. 6051, 898–909.
- Dominic, N., Daniel, Cenggoro, T. W., Budiarto, A., & Pardamean, B. (2021). Transfer learning using inception-resnet-v2 model to the augmented neuroimages data for autism spectrum disorder classification. *Communications in Mathematical Biology and Neuroscience*, *2021*, 1–21. <https://doi.org/10.28919/cmbn/5565>
- Farhan, M., Ryandra, D., Arsitektur, P., Inceptionv, R.-, Dalam, I., Covid, K., Farhan, M., Ryandra, D., Arsitektur, P., & Inceptionv, R.-. (2021). *PERBANDINGAN ARSITEKTUR RESNET-50 DAN BERDASARKAN CITRA X-RAY MUHAMMAD FARHAN DWI RYANDRA PERBANDINGAN ARSITEKTUR RESNET-50 DAN INCEPTIONV3 DALAM KLASIFIKASI COVID 19 BERDASARKAN CITRA X-RAY SKRIPSI* Diajukan sebagai salah satu syarat untuk memperoleh Gela.
- Foeady, A. Z. (2019). *Sistem klasifikasi kanker kulit berdasarkan data citra dermoscopic dengan menggunakan metode deep extreme learning machine*. 144.
- Ghosal, P., Nandanwar, L., Kanchan, S., Bhadra, A., Chakraborty, J., & Nandi, D. (2019). Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. *2019 2nd International Conference on Advanced Computational and Communication Paradigms, ICACCP 2019, February*. <https://doi.org/10.1109/ICACCP.2019.8882973>
- Hafifah, F., Rahman, S., & Asih, S. (2021). Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN).

TIN: Terapan Informatika Nusantara, 2(5), 292–301.

- Hari, P., Pradana, M. G., & Susanto, A. (2020). Kemunculan kembali ulat api *Narosa rosipuncta* Holloway (Lepidoptera: Limacodidae) dan pengendaliannya di perkebunan kelapa sawit Sumatera Utara. *Warta PPKS*, 2(2), 86–91.
- Haripriya, P. (2019). Deep learning pre-trained architecture of alex net and googlenet for DICOM image classification. *International Journal of Scientific and Technology Research*, 8(11), 3107–3113.
- Ifantiska, D. (2022). *SKRIPSI OLEH : FAKULTAS TEKNIK Diajukan sebagai salah satu syarat untuk memperoleh Gelar Sarjana (S1) di Fakultas Teknik Universitas Medan Area OLEH : DIAN IFANTISKA.*
- Liu, Z., Gao, J., Yang, G., Zhang, H., & He, Y. (2016). Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network. *Scientific Reports*, 6(January), 1–12. <https://doi.org/10.1038/srep20410>
- Lubis, A. M. (2020). Sifat Fisik Tanah Lahan Kelapa Sawit yang Dipengaruhi Air Pasang Surut di Desa Selat Besar Kecamatan Bilah Hilir Kabupaten Labuhanbatu. *Skripsi*, 34.
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, 9(4). <https://doi.org/10.3390/technologies9040081>
- Marpaung, F., Khairina, N., & Muliono, R. (2024). *Klasifikasi Daun Teh Siap Panen Menggunakan Convolutional Neural Network Arsitektur Mobilenetv2*. 18(1), 215–225. <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>
- Mustamin, F., Sari, Y., Khatimi, H., Informasi, S. T., Teknik, F., Mangkurat, U. L., & Banjarmasin, K. (2021). *MENGGUNAKAN ARSITEKTUR CNN*. 8(1), 49–59.
- Nanda, B. T., Lestari, W., & Sitanggang, K. D. (2022). Pollen Load and Flower Constan. *Jurnal pertanian Agros*, 24(2), 559–566.
- Nashrullah, F., Wibowo, S. A., & Budiman, D. G. (2020). COMPLETE Journal of Computer, Electronic, and Telecommunication Investigasi Parameter Epoch Pada Arsitektur ResNet-50 Untuk Klasifikasi Pornografi. *Journal of Computer, Electronic, and Telecommunication*, 1(1).
- Nisa, C., Puspaningrum, E. Y., & Maulana, H. (2020). *Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data*. 1, 169–175.
- Nurhasnita, F., Yaherwandi, F., & Efendi, S. (2020). Survei Hama Pada Perkebunan Kelapa Sawit Rakyat di Kecamatan Sembilan Koto Kabupaten

- Dharmasraya. *Agriprima : Journal of Applied Agricultural Sciences*, 4(1), 6–17. <https://doi.org/10.25047/agriprima.v4i1.347>
- Nurona Cahya, F., Hardi, N., Riana, D., Hadiani, S., Mandiri Jakarta Cipinang Melayu, N., Makasar, K., Jakarta Timur, K., & Khusus Ibukota Jakarta, D. (2021). *SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)*. 10, 618–626.
- Oktafanda, E. (2022). Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Informatika Ekonomi Bisnis*, 4(3), 72–77. <https://doi.org/10.37034/infeb.v4i3.143>
- Patani, A., Pardikar, I., Doshi, P., Rodge, S., & Khachane, S. (2021). *Plant Leaf Recognition and Disease Detection Using GoogLeNet*. 8(5), 595–602.
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format: Jurnal Ilmiah Teknik Informatika*, 8(2), 138. <https://doi.org/10.22441/format.2019.v8.i2.007>
- Pratama, Y. (2021). Penggunaan Predator (*Sycanus annulicornis*) Tanaman Kelapa Sawit (*Elaeis guineensis* Jacq.) Dalam Mengendalikan Hama Pemakan Daun (*Setothosea asigna*). *Jurnal Ilmiah Mahasiswa Pertanian ...*, 1(4), 1–10.
- Purwono, P., Wirasto, A., & Nisa, K. (2021). Komparasi Algoritma Machine Learning Untuk Klasifikasi Kelompok Obat. *Sisfotenika*, 11(2), 196.
- Qaim, M., Sibhatu, K. T., Siregar, H., & Grass, I. (2020). Environmental, economic, and social consequences of the oil palm boom. *Annual Review of Resource Economics*, 12, 321–344. <https://doi.org/10.1146/annurev-resource-110119-024922>
- Raffar, M. S., Sugiardi, S., & Youlla, D. (2022). *Faktor-Faktor yang Mempengaruhi Produksi Kelapa Sawit Petani Mandiri di Kecamatan Rasau Jaya Kubu Raya*. 2(1), 174–179.
- Rahman, C. R., Arko, P. S., Ali, M. E., Iqbal Khan, M. A., Apon, S. H., Nowrin, F., & Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*, 194, 112–120. <https://doi.org/10.1016/j.biosystemseng.2020.03.020>
- Ramaneswaran, S., Srinivasan, K., Vincent, P. M. D. R., & Chang, C. Y. (2021). Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification. *Computational and Mathematical Methods in Medicine*, 2021. <https://doi.org/10.1155/2021/2577375>
- Salma, A., Bustamam, A., & Sarwinda, D. (2021). *Diabetic Retinopathy Detection Using GoogleNet Architecture of Convolutional Neural Network Through Fundus Images*. 2021, 1–6.

- Saragih, H. J., & Afrianti, S. (2021). Tingkat serangan hama ulat kantung (mahasena corbetti) pada areal tanaman menghasilkan (tm) kelapa sawit Pt. Indo Sepadan Jaya. *Jurnal Pertanian Berkelanjutan*, 9(2), 88–93.
- Sartika, D. (2022). *Sejarah Perkebunan Kelapa Sawit Di Desa Bakti Mulya Kecamatan Sungai Bahar 1984-2020*. 1–89.
- Setyaningrum, C. A., & Prasetyo, S. Y. J. (2018). Sistem Peramalan Serangan Organisme Pengganggu Tanaman Menggunakan Metode Double Exponential Smoothing Berbasis Google Map. *Indonesian Journal of Computing and Modeling*, 1(1), 1–9. <https://doi.org/10.24246/j.icm.2018.v1.i1.p1-9>
- Setyawan, Y. P., Advento, A. D., & Naim, M. (2021). Konservasi Musuh Alami Ulat Pemakan Daun Kelapa Sawit: Populasi Serangga Parasitika Berdasarkan Jaraknya terhadap Tanaman Berbunga. *Prosiding Seminar Nasional Perhimpunan Entomologi Indonesia Bandung, August 2022*, 96–102.
- Susanto, L. A., Nilogiri, A., & Handayani, L. (2023). *Jurnal Sistem dan Teknologi Informasi Indonesia Klasifikasi Citra Lesi Kulit Serupa Virus Monkeypox Menggunakan VGG-19 Convolutional Neural Network Image Classification of Skin Lesion Similar to Monkeypox Virus Using VGG-19 Convolutional Neural Network*. 8(1), 1–9.
- Suta, I. B. L. M., Hartati, R. S., & Divayana, Y. (2019). Diagnosa Tumor Otak Berdasarkan Citra MRI (Magnetic Resonance Imaging). *Majalah Ilmiah Teknologi Elektro*, 18(2). <https://doi.org/10.24843/mite.2019.v18i02.p01>
- Turnip, K. N. T. T. (2021). artikel INVENTARISASI JENIS HAMA DAN CARA PENGENDALIANNYA DI PEMBIBITAN KELAPA SAWIT (Elaeis guineensis Jacq.) PT. PERKEBUNAN NUSANTARA IV DOLOK SINUMBAH. *Biologica Samudra*, 3(1), 87–93. <https://doi.org/10.33059/jbs.v3i1.2414>
- Widians, J. A., & Rizkyani, F. N. (2020). Identifikasi Hama Kelapa Sawit menggunakan Metode Certainty Factor. *ILKOM Jurnal Ilmiah*, 12(1), 58–63. <https://doi.org/10.33096/ilkom.v12i1.526.58-63>
- Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, 507, 772–794. <https://doi.org/10.1016/j.ins.2019.06.064>
- Yuesheng, F., Jian, S., Fuxiang, X., Yang, B., Xiang, Z., Peng, G., Zhengtao, W., & Shengqiao, X. (2021). Circular Fruit and Vegetable Classification Based on Optimized GoogLeNet. *IEEE Access*, 9, 113599–113611. <https://doi.org/10.1109/ACCESS.2021.3105112>
- Yuliani, A., Labellapansa, A., & Yulianti, A. (2019). Klasifikasi Citra Daun Kelapa Sawit Yang Terkena Dampak Hama Menggunakan Metode K-Nearest Neighbor. *Seminar Nasional Informatika Medis (SNIMed)*, 73–78.

Zarliani, W. O. Al, Purnamasari, W. Od. D., & Muzuna. (2020). Cara Pengendalian Organisme Pengganggu Tanaman (Opt) Tanaman Sayuran Di Kelurahan Ngkaring-Karing. *Jurnal Pengabdian Kepada Masyarakat Membangun Negeri*, 4(2), 188–195.

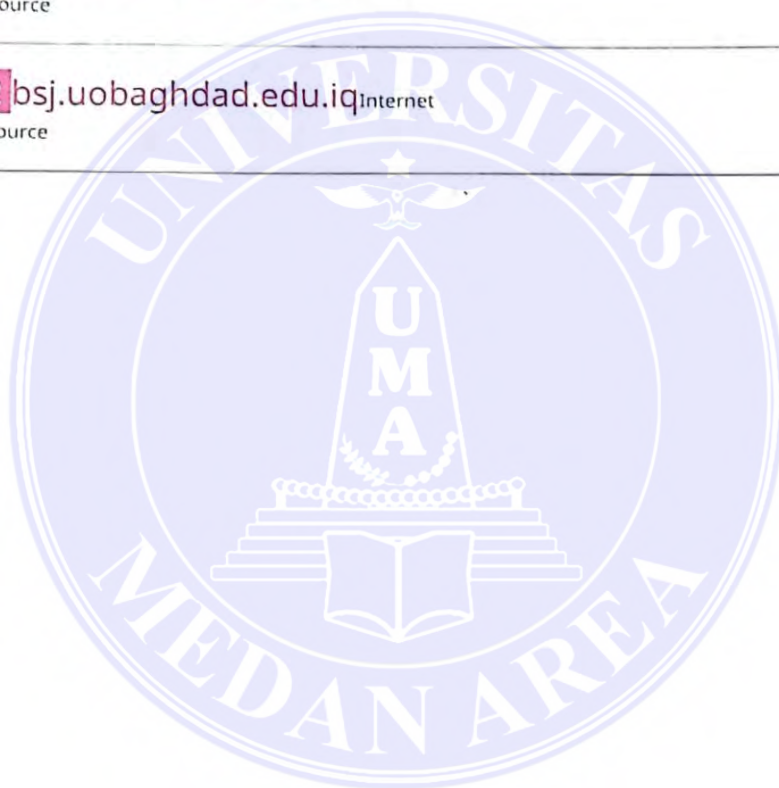
Zeng, G., He, Y., Yu, Z., Yang, X., Yang, R., & Zhang, L. (2016). Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube. *Journal of Chemical Technology and Biotechnology*, 91(8), 2322–2330. <https://doi.org/10.1002/jctb.4820>



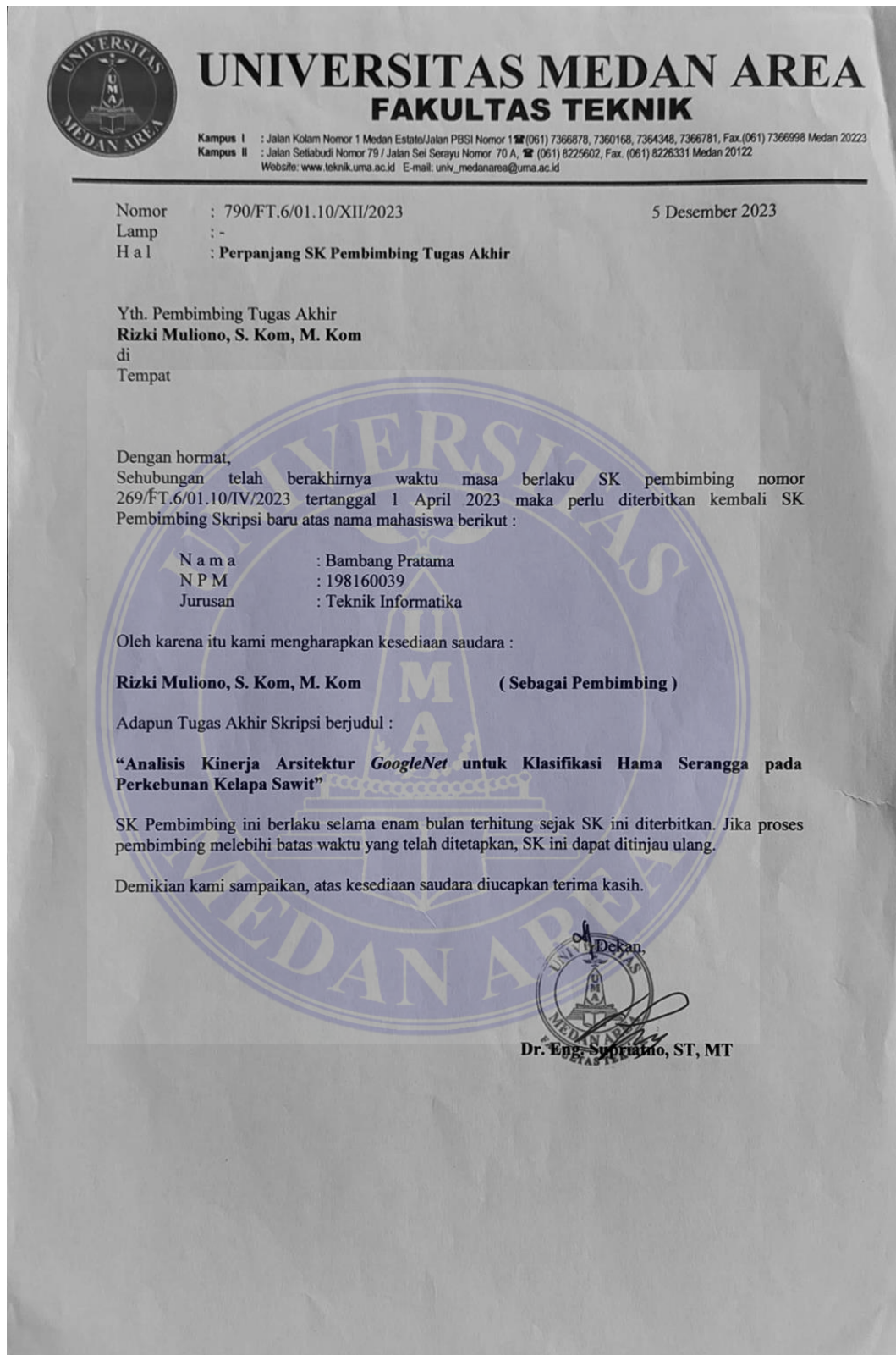
LAMPIRAN


1. Lampiran Hail Plagiasi

SIMILARITY INDEX	22 %	22	INTERNET	4%	0%
SOURCES%			PUBLICATIONS		STUDENT PAPERS
PRIMARY SOURCES					
1	repositori.uma.ac.id	Internet			20%
Source					
2	bsj.uobaghdad.edu.iq	Internet			2%
Source					



2. Lampiran SK Pembimbing Tugas Akhir



 **UNIVERSITAS MEDAN AREA**
FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 (061) 7366878, 7360168, 7364348, 7366781, Fax.(061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 790/FT.6/01.10/XII/2023 5 Desember 2023
Lamp : -
Hal : Perpanjang SK Pembimbing Tugas Akhir

Yth. Pembimbing Tugas Akhir
Rizki Muliono, S. Kom, M. Kom
di
Tempat

Dengan hormat,
Sehubungan telah berakhirnya waktu masa berlaku SK pembimbing nomor 269/FT.6/01.10/IV/2023 tertanggal 1 April 2023 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa berikut :

N a m a : Bambang Pratama
N P M : 198160039
Jurusan : Teknik Informatika

Oleh karena itu kami mengharapkan kesediaan saudara :


Rizki Muliono, S. Kom, M. Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

“Analisis Kinerja Arsitektur *GoogleNet* untuk Klasifikasi Hama Serangga pada Perkebunan Kelapa Sawit”

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,
Dr. Eng. Suprianto, ST, MT

3. Lampiran Surat Penelitian dan Pengambilan Data Tugas Akhir



UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 (061) 7366878, 7360168, 7364348, 7366781, Fax.(061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, (061) 8225602, Fax, (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 451 /FT.6/01.10/VI/2023 13 Juni 2023
Lamp : -
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Pimpinan PTPN IV Kebun Ajamu
Ajamu, Kec. Panai Hulu
Di
Labuhan Batu

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

NO	N A M A	N P M	PRODI
1	Bambang Pratama	198160039	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 Kinerja Arsitektur *GoogleNet* untuk Klasifikasi Hama Serangga pada Perkebunan Kelapa Sawit

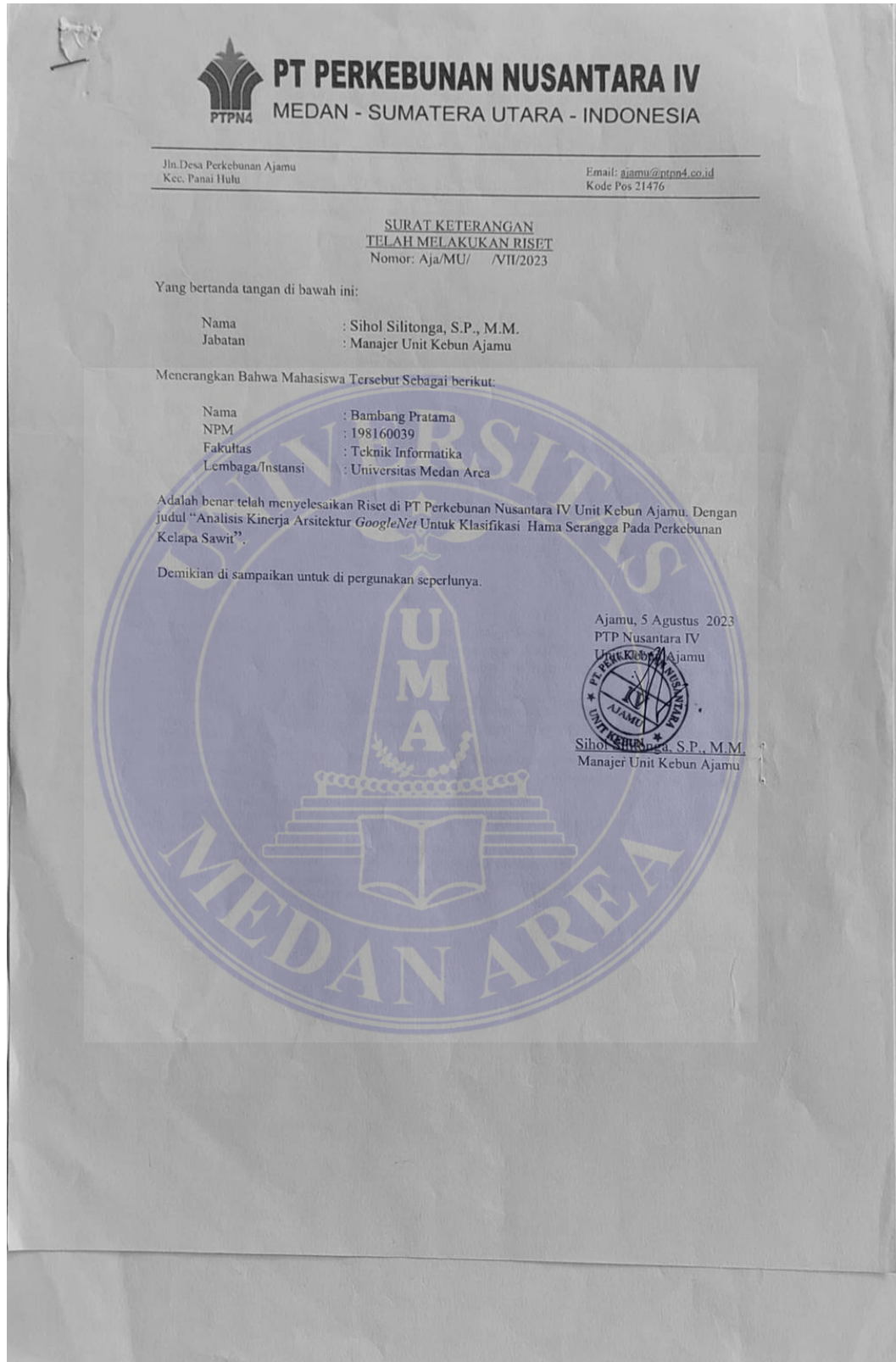
Atas perhatian dan kerja sama yang baik diucapkan terima kasih.



Dr. R. S. Kom, M. Kom

- Tembusan :
1. Ka. BAMAI
 2. Mahasiswa
 3. File

4. Lampiran Surat Keterangan Selesai Penelitian



5. Lampiran Source Code Google Colabatory

a) Import

```
import numpy as np
import pickle
import cv2
import keras
import seaborn as sns
import tensorflow as tf

from tensorflow.keras.utils import plot_model
from keras.applications.xception import Xception
from keras.preprocessing import image
from keras.layers import Conv2D, MaxPool2D, Dropout, Dense, Input,
concatenate, GlobalAveragePooling2D, AveragePooling2D, Flatten
from keras.layers import Flatten, Dense
from keras.models import Model
from keras.layers import BatchNormalization

from os import listdir
from keras import backend as K
from keras.layers import Input
from tensorflow.keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from tensorflow.keras.utils import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from IPython.display import display, clear_output
from skimage.io import imsave, imread
from warnings import filterwarnings
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import LearningRateScheduler

import matplotlib.pyplot as plt
import time
import datetime
import ipywidgets as widgets
import io
```

b) Pemanggilan Google Drive ke Google Colab

```
from google.colab import drive
drive.mount('/content/drive')
```

c) Hyper Parameter

```
EPOCHS = 15
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((224, 224))
image_size = 0
directory_root = '/content/drive/My Drive/Hama Tanaman Klapa Sawit'
width=224
height=224
depth=3
```

d) Image to Array

```
def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None :
            image = cv2.resize(image, default_image_size)
            return img_to_array(image)
        else :
            return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None
```

e) Pmanggilan Dataset dari direktori

```
image_list, label_list = [], []
try:
    print("[INFO] Loading images ...")
    root_dir = listdir(directory_root)
    for directory in root_dir :
        # remove .DS_Store from list
        if directory == ".DS_Store" :
            root_dir.remove(directory)

    for Hama_tanaman_kelapa_sawit_folder in root_dir :
        Hama_UPDKS_folder_list =
```

```

listdir(f"{directory_root}/{Hama_tanaman_kelapa_sawit_folder}")

for Hama_folder in Hama_UPDKS_folder_list :
    # remove .DS_Store from list
    if Hama_folder == ".DS_Store" :
        Hama_UPDKS_folder_list.remove(Hama_folder)

for Hama_UPDKS_folder in Hama_UPDKS_folder_list:
    print(f"[INFO] Processing {Hama_UPDKS_folder} ...")
    Hama_UPDKS_image_list =
listdir(f"{directory_root}/{Hama_tanaman_kelapa_sawit_folder}/{Hama_U
PDKS_folder}/")

    for single_Hama_UPDKS_image in Hama_UPDKS_image_list :
        if single_Hama_UPDKS_image == ".DS_Store" :
            Hama_UPDKS_image_list.remove(single_Hama_UPDKS_image)

    for image in Hama_UPDKS_image_list[:]:
        image_directory =
f"{directory_root}/{Hama_tanaman_kelapa_sawit_folder}/{Hama_UPDKS_
folder}/{image}"
        if image_directory.endswith(".jpg") == True or
image_directory.endswith(".JPG") == True:
            image_list.append(convert_image_to_array(image_directory))
            label_list.append(Hama_UPDKS_folder)
    print("[INFO] Image loading completed")
except Exception as e:
    print(f"Error : {e}")

```

f) Train & Test

```

x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels,
test_size=0.2, random_state = 42)

```

g) Data Generator

```

aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2, horizontal_flip=True, fill_mode="nearest")

```

h) Inception Module

```
# Defining Inception Model

# Impor paket yang diperlukan Model
import tensorflow as tf
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import concatenate

# Inialisasi Core
kernel_init = tf.keras.initializers.glorot_uniform()

# Inialisasi offset
bias_init = tf.keras.initializers.Constant(value=0.2)

# Fungsi yang menghasilkan Modul Inception
def inception_module(x,
                    filters_1x1,
                    filters_3x3_reduce,
                    filters_3x3,
                    filters_5x5_reduce,
                    filters_5x5,
                    filters_pool_proj,
                    name=None):

    # Konvolusi 1 x 1
    conv_1x1 = Conv2D(filters_1x1,
                     (1, 1),
                     padding='same',
                     activation='relu')(x)
    conv_1x1 = BatchNormalization()(conv_1x1)

    # Konvolusi 1 x 1 untuk reduksi dimensi Konvolusi 3x3
    conv_3x3 = Conv2D(filters_3x3_reduce,
                     (1, 1),
                     padding='same',
                     activation='relu')(x)
    conv_3x3 = BatchNormalization()(conv_3x3)
```



```
# Konvolusi 3x3
conv_3x3 = Conv2D(filters_3x3,
                 (3, 3),
                 padding='same',
                 activation='relu')(conv_3x3)
conv_3x3 = BatchNormalization()(conv_3x3)

# Konvolusi 1 x 1 untuk reduksi dimensi Konvolusi 5x5
conv_5x5 = Conv2D(filters_5x5_reduce,
                 (1, 1),
                 padding='same',
                 activation='relu')(x)
conv_5x5 = BatchNormalization()(conv_5x5)

# Konvolusi 5x5
conv_5x5 = Conv2D(filters_5x5, (5, 5),
                 padding='same',
                 activation='relu')(conv_5x5)
conv_5x5 = BatchNormalization()(conv_5x5)

# Max pooling
pool_proj = MaxPool2D((3, 3), strides=(1, 1), padding='same')(x)

# Konvolusi 1 x 1 untuk mencerna dimensi maksimum yang dikurangi
pool_proj = Conv2D(filters_pool_proj,
                 (1, 1),
                 padding='same',
                 activation='relu')(pool_proj)
pool_proj = BatchNormalization()(pool_proj)

# Stack merge
output = concatenate([conv_1x1, conv_3x3, conv_5x5, pool_proj], axis=3,
                    name=name)

return output
```

i) Modul GoogleNet

```

# Impor paket yang diperlukan
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense

# Mendefinisikan GoogleNet / Inception-V1
class GoogleNet:
    @staticmethod
    def build(width, height, channel, classes):

        input_layer = Input(shape=(width, height, channel))

        # Inisialisasi inti
        kernel_init = tf.keras.initializers.glorot_uniform()

        # Inisialisasi offset
        bias_init = tf.keras.initializers.Constant(value=0.2)
        # Konvolus
        x = Conv2D(64,
                  (7, 7),
                  padding='same',
                  strides=(2, 2),
                  activation='relu',
                  name='conv_1_7x7/2')(input_layer)
        x = BatchNormalization()(x)

        # Max pooling
        x = MaxPool2D((3, 3), padding='same', strides=(2, 2),
                     name='max_pool_1_3x3/2')(x)

        # Konvolusi
        x = Conv2D(64,
                  (1, 1)

```

```
padding='same',
strides=(1, 1),
activation='relu',
name='conv_2a_3x3/1')(x)
x = BatchNormalization()(x)

# Konvolusi
x = Conv2D(192,
          (3, 3),
          padding='same',
          strides=(1, 1),
          activation='relu',
          name='conv_2b_3x3/1')(x)
x = BatchNormalization()(x)

# Max pooling
x = MaxPool2D((3, 3), padding='same', strides=(2, 2),
              name='max_pool_2_3x3/2')(x)

# Inception Module
x = inception_module(x,
                    filters_1x1=64,
                    filters_3x3_reduce=96,
                    filters_3x3=128,
                    filters_5x5_reduce=16,
                    filters_5x5=32,
                    filters_pool_proj=32,
                    name='inception_3a')

# Inception Module
x = inception_module(x,
                    filters_1x1=128,
                    filters_3x3_reduce=128,
                    filters_3x3=192,
                    filters_5x5_reduce=32,
                    filters_5x5=96,
                    filters_pool_proj=64,
                    name='inception_3b')

# Max pooling
x = MaxPool2D((3, 3), padding='same', strides=(2, 2),
              name='max_pool_3_3x3/2')(x)
```

```
# Inception Module
x = inception_module(x,
    filters_1x1=192,
    filters_3x3_reduce=96,
    filters_3x3=208,
    filters_5x5_reduce=16,
    filters_5x5=48,
    filters_pool_proj=64,
    name='inception_4a')

# Inception Module
x = inception_module(x,
    filters_1x1=160,
    filters_3x3_reduce=112,
    filters_3x3=224,
    filters_5x5_reduce=24,
    filters_5x5=64,
    filters_pool_proj=64,
    name='inception_4b')

# Inception Module
x = inception_module(x,
    filters_1x1=128,
    filters_3x3_reduce=128,
    filters_3x3=256,
    filters_5x5_reduce=24,
    filters_5x5=64,
    filters_pool_proj=64,
    name='inception_4c')

# Inception Module
x = inception_module(x,
    filters_1x1=112,
    filters_3x3_reduce=144,
    filters_3x3=288,
    filters_5x5_reduce=32,
    filters_5x5=64,
    filters_pool_proj=64,
    name='inception_4d')

# Inception Module
x = inception_module(x,
```



```
filters_1x1=256,
filters_3x3_reduce=160,
filters_3x3=320,
filters_5x5_reduce=32,
filters_5x5=128,
filters_pool_proj=128,
name='inception_4e')

# Max pooling
x = MaxPool2D((3, 3), padding='same', strides=(2, 2),
name='max_pool_4_3x3/2')(x)

# Inception Module
x = inception_module(x,
filters_1x1=256,
filters_3x3_reduce=160,
filters_3x3=320,
filters_5x5_reduce=32,
filters_5x5=128,
filters_pool_proj=128,
name='inception_5a')

# Inception Module
x = inception_module(x,
filters_1x1=384,
filters_3x3_reduce=192,
filters_3x3=384,
filters_5x5_reduce=48,
filters_5x5=128,
filters_pool_proj=128,
name='inception_5b')

# Global Avarage Pooling
x = GlobalAveragePooling2D(name='avg_pool_5_3x3/1')(x)

# Random inactivation
x = Dropout(0.40)(x)

# Full connection/output
x = Dense(classes, activation='softmax', name='output')(x)

# Create GoogleNet model
```

```
# return Model(input_layer, [x, x1, x2], name='inception_v1')
return Model(input_layer, x, name='inception_v1')

# Test GoogleNet class instantiation and output summary information of
GoogleNet model
if __name__ == "__main__":
    model = GoogleNet.build( width=224, height=224, channel=3, classes=3)
    print(model.summary())
```

j) Model Fit Generator

```
opt = tf.keras.optimizers.legacy.Adam(learning_rate=INIT_LR, decay=INIT_LR /
    EPOCHS)
# distribution
model.compile(loss="categorical_crossentropy",
    optimizer=opt,metrics=["accuracy"])
# train the network
print("[INFO] training network...")

history = model.fit(
    aug.flow(x_train, y_train, batch_size=BS),
    steps_per_epoch=len(x_train) // BS,
    epochs=EPOCHS,
    validation_data=(x_test, y_test),
    validation_steps=10,
    verbose=1
)
print('Training time: %s' %(now()-t))
```

k) Grafik Training & Validation

```
def show_plots(history):
    loss_vals = history['loss']
    val_loss_vals = history['val_loss']
    epochs = range(1, len(history['accuracy'])+1)

    f, ax = plt.subplots(nrows=1,ncols=2,figsize=(15,6))

    # plot loss
    ax[0].plot(epochs, loss_vals, color='red',marker='o', label='Training Loss')
```

```
ax[0].plot(epochs, val_loss_vals, color='black', marker='*', label='Validation
Loss')
ax[0].set_title('Training & Validation Loss')
ax[0].set_xlabel('Epochs')
ax[0].set_ylabel('Loss')
ax[0].legend(loc='best')
ax[0].grid(True)

# plot accuracy
acc_vals = history['accuracy']
val_acc_vals = history['val_accuracy']

ax[1].plot(epochs, acc_vals, color='red', marker='o', label='Training
Accuracy')
ax[1].plot(epochs, val_acc_vals, color='black', marker='*', label='Validation
Accuracy')
ax[1].set_title('Training & Validation Accuracy')
ax[1].set_xlabel('Epochs')
ax[1].set_ylabel('Accuracy')
ax[1].legend(loc='best')
ax[1].grid(True)

plt.show()
plt.close()

# delete locals from heap before exiting
del loss_vals, val_loss_vals, epochs, acc_vals, val_acc_vals
show_plots(history.history)
```

1) Model Evaluasi

```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

m) Clasificatio Report

```
pred = model.predict(x_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(y_test,axis=1)

print(classification_report(y_test_new,pred))
```

n) Confusion Matrix

```
import matplotlib.pyplot as plt

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```