

**IMPLEMENTASI ARSITEKTUR GOOGLNET DAN
XCEPTION UNTUK IDENTIFIKASI PENYAKIT
PADA DAUN TANAMAN KELAPA SAWIT**

SKRIPSI

OLEH :

DIAN IFANTISKA

17.816.0083



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2022

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 16/11/22

Access From (repository.uma.ac.id)16/11/22

**IMPLEMENTASI ARSITEKTUR GOOGLNET DAN
XCEPTION UNTUK IDENTIFIKASI PENYAKIT
PADA DAUN TANAMAN KELAPA SAWIT**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh

Gelar Sarjana (S1) di Fakultas Teknik

Universitas Medan Area

OLEH :

DIAN IFANTISKA

17.816.0083

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2022

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

ii

Document Accepted 16/11/22


Access From (repository.uma.ac.id)16/11/22

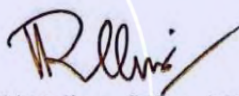
LEMBAR PENGESAHAN

Judul Skripsi : Implementasi Arsitektur Googlenet dan Xception Untuk Identifikasi Penyakit Pada Daun Tanaman Kelapa Sawit

Nama : Dian Ifantiska
NPM : 178160083
Fakultas : Teknik
Prodi : Informatika

Disetujui Oleh
Komisi Pembimbing


Muhathir, S.T, M.Kom
Pembimbing I


Rizki Muliono, S.Kom, M.Kom
Pembimbing II

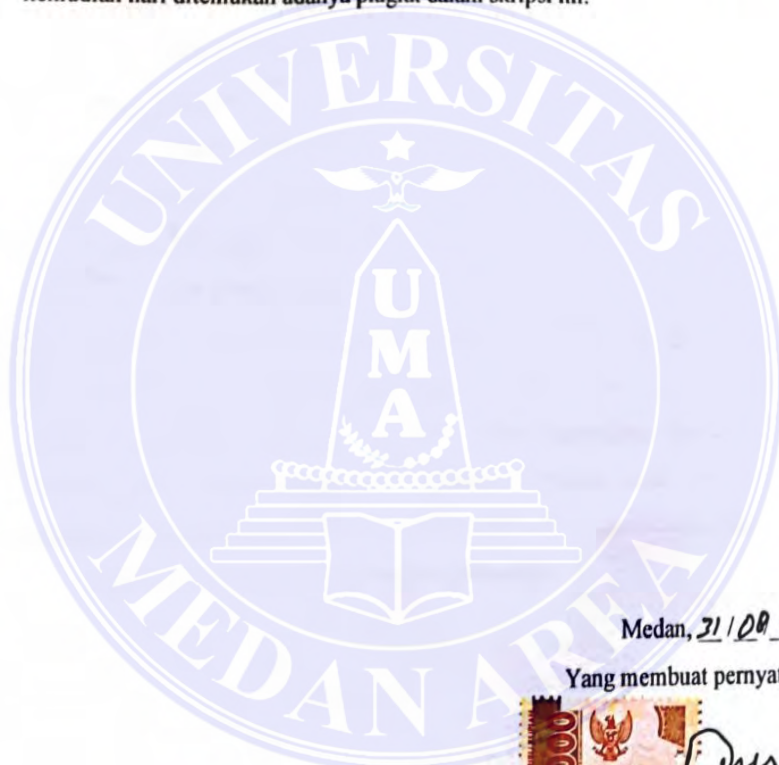
Mengetahui


Rahmatul Syah, S.Kom, M.Kom
MEDAN : 0105058804


Rizki Muliono, S.Kom, M.Kom
MEDAN AREA : 0109038902

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar serjana 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 sanksisanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.



Medan, 31/08/2022

Yang membuat pernyataan



Dian Ifantiska

178160083

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademika Universitas Medan Area, Saya yang bertanda tangan dibawah ini.

Nama : Dian Ifantiska
NPM : 178160083
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis Karya : 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 :

Implementasi Arsitektur Googlenet dan Xception Untuk Identifikasi Penyakit Pada Daun Tanaman Kelapa Sawit

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan 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.

Dibuat di : *Medan*

Pada tanggal : *21 Agustus 2022*

Yang menyatakan



(Dian Ifantiska)

ABSTRAK

Dalam aktivitas budidaya kelapa sawit para petani kerap kali mengalami bermacam-macam serbuan penyakit yang melanda Tanaman kelapa sawit. Minimnya wawasan petani sawit mengenai penyakit pada tanaman kelapa sawit jadi hambatan para petani akan menanggulangi sendiri penyakit yang melanda tanaman kelapa sawit yang dimiliki. Oleh karena itu dibuatlah penelitian untuk mengenali penyakit pada daun kelapa sawit khususnya pada daun yang terkena hama ulat api dan ulat kantung. Convolutional Neural Network (CNN) banyak digunakan pada penelitian terdahulu karena akurasi yang tinggi dan memiliki beberapa pengembangan arsitektur, diantaranya terdapat arsitektur Googlenet dan Xception. Arsitektur ini dipilih karena memiliki tingkat akurasi yang tinggi pada ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) dan menggunakan pendekatan transfer learning yang populer. Pada penelitian ini, Mengimplementasikan dan membandingkan Googlenet dan Xception untuk identifikasi penyakit pada daun tanaman kelapa sawit, dengan jumlah dataset sebanyak 1230 gambar , yang terdiri dari gambar daun sehat, daun yang terkena ulat api, dan daun yang terkena ulat kantung. Setelah data gambar penyakit daun dilatih, model data pelatihan akan disimpan untuk proses pengujian. Evaluasi pengujian disimpan sebagai confusion matriks. Hasilnya, Xception memiliki model akurasi sebesar 99,72 % sedangkan Googlenet memiliki model akurasi sebesar 93,22 %.

Kata Kunci : Kelapa Sawit; Googlenet; Xception.

ABSTRACT

In oil palm cultivation activities, farmers often experience various kinds of disease attacks that hit oil palm plants. The lack of knowledge of oil palm farmers about diseases in oil palm plants is an obstacle for farmers to overcome the diseases that affect their oil palm plants. Therefore, a study was made to identify diseases on oil palm leaves, especially on leaves affected by caterpillars and bagworms. Convolutional Neural Network (CNN) is widely used in previous studies because of its high accuracy and has several architectural developments, including the Googlenet and Xception architectures. This architecture was chosen because it has a high level of accuracy in the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) and uses the popular transfer learning approach. In this study, implementing and comparing Googlenet and Xception to identify diseases on oil palm leaves, with a total dataset of 1230 images, consisting of images of healthy leaves, leaves affected by caterpillars, and leaves affected by bagworms. After the leaf disease image data is trained, the training data model will be saved for the testing process. The test evaluation is stored as a confusion matrix. As a result, Xception has an accuracy model of 99.72% while Googlenet has an accuracy model of 93.22%.

Keywords : *Palm oil; Googlenet; Xception.*

RIWAYAT HIDUP

Penulis dilahirkan di Dumai Pada tanggal 08 Februari 1999 dari ayah Junaidi dan ibu Susri Wigati Penulis merupakan putra ke-dua (kedua) dari dua (2) bersaudara. Tahun 2017 Penulis lulus dari SMA Negeri 3 Pematangsiantar dan pada tahun 2017 terdaftar sebagai mahasiswa Fakultas Teknik Universitas Medan Area. Selama mengikuti perkuliahan, penulis aktif mengikuti organisasi seperti HMIF (Himpunan Mahasiswa Informatika) pada tahun ajaran 2019/2020. Dan pada tahun 2020 Penulis melaksanakan praktek kerja lapangan (PKL) di SMK Negeri 1 Siantar



KATA PENGANTAR

Puji dan Syukur penulis ucapkan kepada Allah SWT atas segala karunia dan Hidayah-Nya, sehingga Tugas Akhir yang berjudul :“ Implementasi Arsitektur Googlenet dan Xception Untuk Identifikasi Penyakit Pada Daun Tanaman Kelapa Sawit ” dapat diselesaikan dengan baik dan tepat waktu. Adapun tujuan penulisan laporan ini adalah untuk memenuhi persyaratan dalam menyelesaikan Tugas Akhir Program Studi Teknik Informatika Universitas Medan Area.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. **Tuhan Yang Maha Esa**, yang telah memberikan kesehatan dan keselamatan selama melakukan studi.
2. Orang tua, Ayah dan Bunda, Abang yang telah memberikan doa, semangat, dukungan, dan motivasi selama melakukan studi.
3. Bapak **Prof. Dr. Dadan Ramdan, M.Eng, M.Sc.**, selaku Rektor Universitas Medan Area.
4. Bapak **Dr. Rahmad Syah, S.Kom, M. Kom** selaku Dekan Fakultas Teknik Universitas Medan Area.
5. Ibu **Susilawati S.Kom, M.Kom** Selaku Wakil Dekan.
6. Bapak **Rizki Muliono, S.Kom, M.Kom** selaku Kepala Program Studi Teknik Informatika dan Dosen Pembimbing II Tugas Akhir.
7. Bapak **Muhathir, S.T, M.Kom** selaku Dosen Pembimbing I.
8. Bapak **Mahardika Gama Pradana, M.Si** selaku Dosen Pembimbing Lapangan.
9. Pusat Penelitian Kelapa Sawit (PPKS) yang telah mengizinkan saya untuk menjalankan riset.
10. Seluruh dosen Teknik Informatika yang selama ini telah membekali penulis dengan ilmu ilmu yang semoga di suatu hari nanti dapat digunakan dengan baik.
11. Seluruh pegawai Universitas Medan Area yang telah membantu dalam proses administrasi.
12. Teman-teman seperjuangan yang telah berupaya untuk membantu dan memberikan dukungan.

13. Semua pihak yang turut membantu dan tidak dapat penulis sebutkan satu per satu yang memiliki andil dalam penyusunan Tugas Akhir ini.

Penulis Menyadari bahwa penyusunan Tugas Akhir ini masih jauh dari kesempurnaan dikarenakan pengetahuan dan pengalaman penulis, maka dari itu kritik dan saran dari pembaca sangat diharapkan agar kedepannya penulis bisa lebih baik lagi dalam penulisan ilmiah.

Medan, 31 Agustus 2022

Dian Ifantiska



DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
ABSTRAK	vi
ABTRACT	vii
RIWAYAT HIDUP	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	viii
DAFTAR TABEL	xv
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
BAB II.....	7
LANDASAN TEORI	7
2.1 Citra Digital	7
2.2 Convolutional Neural Network(CNN)	7
2.3 Arsitektur.....	10
2.3.1 Googlenet	10
2.3.2 Xception	11
2.4 Tanaman Kelapa Sawit.....	12
2.5 Penyakit Daun Pada Tanaman Kelapa Sawit	13
2.5.1 Ulat Kantung	13
2.5.2 Ulat Api	14
2.6 Penelitian Terdahulu.....	14
BAB III.....	17
METODOLOGI PENELITIAN	17
3.1 Alat dan Bahan Penelitian	18

3.2	Diagram Alur Penelitian	18
3.3	Metode Pengumpulan Data	19
3.4	Rancangan Arsitektur	19
3.4.1	Googlenet	20
3.4.2	Xception	22
3.5	Data Training	24
3.6	Data Testing	24
3.7	Hasil	24
BAB IV		26
HASIL DAN PEMBAHASAN		26
4.1	Hasil	26
4.1.1	Pengumpulan Data	26
4.1.2	Hyper Paramater Model	28
4.1.3	Pra Pemrosesan	28
4.1.4	Implementasi	29
4.1.4.1	Implementasi Googlenet	29
4.1.4.2	Implementasi Xception	33
4.1.5	Hasil Training dan Validation Arsitektur Googlenet	37
4.1.6	Hasil Training dan Validation Arsitektur Xception	38
4.1.7	Hasil Perbandingan Arsitektur Googlenet dan Xception	49
4.1.9	Pengujian Data	40
4.2	Pembahasan	41
BAB V		44
KESIMPULAN DAN SARAN		44
5.1	Kesimpulan	44
5.2	Saran	44
DAFTAR PUSTAKA		45
Lampiran		48

DAFTAR GAMBAR

Gambar 2.1	Konsep Citra Digital.	7
Gambar 2.2	Proses Konvolusi	7
Gambar 3.3	Proses Max Pooling.	8
Gambar 2.2	Arsitektur Googlenet.....	9
Gambar 2.3	Arsitektur Xception.....	10
Gambar 2.4	Ulat kantung dan dampaknya pada daun kelapa sawit	11
Gambar 2.5	Ulat Api dan dampaknya pada daun kelapa sawit	12
Gambar 3.1	Alur Penelitian Secara umum	14
Gambar 3.2	Citra Daun Kelapa Sawit	15
Gambar 3.3	Simulasi Arsitektur Googlenet.....	18
Gambar 3.4	Inception Modul pada Arsitektur googlenet	19
Gambar 3.5	Konvolusi 5 x 5.....	19
Gambar 3.6	Max Pooling.....	20
Gambar 3.7	Simulasi Arsitektur Xception.....	20
Gambar 3.8	Entry Flow Arsitektur Xception.....	21
Gambar 3.9	Xception dengan satu konvolusi spasial per output.....	21
Gambar 4.1	Dataset Dataset yang digunakan terdiri dari a)Gambar daun yang terkena Ulat Api, b) Gambar daun yang terkena Ulat kantung, c) Gambar daun yang sehat	27
Gambar 4.2	Ilustrasi gambar menjadi array.....	29
Gambar 4.3	Training Accuracy Googlenet Epoch 15.....	29
Gambar 4.4	Loss Accuracy Googlenet Epoch 15.....	29

Gambar 4.5	Accuracy Validation Googlenet Epoch 15	30
Gambar 4.6	Loss Validation Googlenet Epoch 15	30
Gambar 4.7	Accuracy Training Googlenet Epoch 25.....	31
Gambar 4.8	Loss Accuracy Googlenet Epoch 25.....	31
Gambar 4.9	Accuracy Validation Googlenet Epoch 25	32
Gambar 4.10	Loss Validation Googlenet Epoch 25	32
Gambar 4.11	Training Accuracy Exception Epoch 15	33
Gambar 4.12	Loss Accuracy Exception Epoch 15	33
Gambar 4.13	Accuracy Validation Exception Epoch 15.....	34
Gambar 4.14	Loss Validation Exception Epoch 15.....	34
Gambar 4.15	Training Accuracy Exception Epoch 25	35
Gambar 4.16	Loss Accuracy Exception Epoch 25	35
Gambar 4.17	Accuracy Validation Exception Epoch 25.....	36
Gambar 4.18	Loss Validation Exception Epoch 25.....	36

DAFTAR TABEL

Tabel 2.1	Penelitian Terdahulu	13
Tabel 3.1	Perangkat Keras.....	15
Tabel 3.2	Perangkat Lunak.....	15
Tabel 3.3	Confusion Matrikx	22
Tabel 4.1	Pembagian Data.....	26
Tabel 4.2	Pembagian Data.....	27
Tabel 4.3	Hyper Paramater Model	28
Tabel 4.4	Hasil Hyper Paramter Arsitektur Googlenet	37
Tabel 4.5	Hasil Hyper Paramter Arsitektur Xception	38
Tabel 4.6	Perbandingan Rata-rata klasifikasi model deep learning	39
Tabel 4.7	Perbandingan Hyper Parameter Optimal Training Model	39
Tabel 4.8	Persentase Tingkat Akurasi Hasil Identifikasi Jenis Penyakit	
Tabel 4.8	Daun Pada Tanaman Kelapa Sawit	40
Tabel 4.9	Perbandingan Kinerja Model dengan penelitian terdahulu	42

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tanaman kelapa sawit hidup dinegara tropis seperti Indonesia, Malaysia, Papua Nugini, dan Thailand. Di Indonesia, tanaman kelapa sawit mempunyai peran besar bagi pertumbuhan perkebunan. Perkebunan kelapa sawit di Indonesia kian mengalami perkembangan dan menyingsing Indonesia sebagai penghasil nomor satu minyak sawit yang menghasilkan lebih dari 44% minyak sawit dunia (Widians, Taruk, Fauziah, & Setyadi, 2019).

Tanaman kelapa sawit tumbuh di kawasan semak belukar, kelapa sawit dapat tumbuh dengan baik di kawasan tropis dan tumbuh pada iklim dengan intensitas hujan stabil. Pola intensitas hujan per tahunnya juga berdampak pada pertumbuhan dan hasil produksi buah sawit (Yuliani, Labellapansa, & Yulianti, 2019). Pada tahun 2018 luas sector perkebunan kelapa sawit di Indonesia mencapai 14.3 juta ha dan luas perkebunan kalapa sawit di Provinsi Riau sampai tahun 2018 adalah 2.7 juta hektar.

Produktifitas kelapa sawit sangat terikat pada daun sebagai tempat terjadinya proses fotosintesis. Akan tetapi, tanaman kelapa sawit juga mudah terinfeksi Organisme Pengganggu Tanaman (OPT) atau hama (Harahap, Fajri, Syahputra, Rahmat, & Nababan, 2013) salah satunya penyakit daun. Hampir seluruh bagian tanaman kelapa sawit menjadi incaran dari aneka penyakit tanaman. Tentu hal ini dapat membuat gagalnya panen pada tanaman kelapa sawit atau berkurangnya nilai mutu buah kelapa sawit itu sendiri. Untuk mempertahankan produktifitas tumbuhan senantiasa tinggi dibutuhkan pemeliharaan yang pas serta salah satu faktor pemeliharaan Tumbuhan Menciptakan (TM) merupakan pengendalian penyakit (Saragih, Jean Cross Sihombing, & Rahmi, 2018).

Dalam aktivitas budidaya kelapa sawit para petani kerap kali mengalami bermacam-macam serbuan penyakit yang melanda tumbuhan kelapa sawit. Minimnya wawasan petani sawit mengenai penyakit pada tanaman kelapa sawit jadi hambatan para petani akan menanggulangi sendiri penyakit yang melanda

tanaman kelapa sawit yang dimiliki (Supriyanto, Jusak, & Sudarmaningtyas, 2014). Cara yang saat ini dilakukan untuk mengidentifikasi berbagai macam penyakit daun tanaman kelapa sawit ialah dengan cara melihat langsung indikasi yang bermasalah pada daun (Uktoro, 2017), Cara tersebut dinilai kurang akurat karena tidak banyak petani mempunyai pengetahuan tentang penyakit pada tanaman kelapa sawit. Pakar pertanian dalam perihal ini memiliki keahlian untuk menganalisa ciri-ciri penyakit tanaman tersebut, Akan tetapi untuk menanggulangi seluruh permasalahan yang dialami oleh petani sawit akan terkendala oleh waktu serta banyaknya petani sawit yang memiliki permasalahan dengan kebunnya (Saragih, Jean Cross Sihombing, & Rahmi, 2018).

Sesungguhnya tiap penyakit tumbuhan tersebut saat sebelum mencapai sesi yang lebih parah serta meluas biasanya menampilkan tanda- tanda penyakit yang dialami namun masih dalam kondisi yang tidak parah serta masih sedikit. Tetapi kerap menghiraukan tentang ini, serta akan menyangka indikasi tersebut telah umum terjadi pada masa tanam sawit, sampai suatu saat muncul indikasi yang sangat parah serta menyebar luas (Satria, Poningsi, & Saputra, 2019).

Pada pnelitian ini mencoba mengurangi kesalahan dan membantu dalam menentukan penyakit atau hama pada daun tanaman kelapa sawit melalui citra. Dengan demikian, penyakit pada daun tanaman kelapa sawit dapat diidentifikasi menggunakan citra, yang selanjutnya akan dilakukan pengklasifikasian agar nantinya identifikasi yang dilakukan memberikan hasil yang optimum.

Pada Penelitian yang dilakukan oleh Marcelina, Yulianti, dan Mair (2022), Identifikasi Penyakit Tanaman Kelapa Sawit menggunakan Metode Forward Chaining dengan sistem pakar, dengan memberikan kuisisioner kepada 35 orang responden untuk mendapat penilaian terhadap sistem. Hasil kuisisioner (Pakar) diperoleh nilai kepuasan sebesar 89.2%, akan tetapi penelitian tersebut berpusat pada sistem pakar yang mana penyakit dapat diketahui melalui pengetahuan yang disimpan kedalam sistem dan sistem akan meminta masukan ciri-ciri agar dapat mendiagnosa, dan pada penelitian yang dilakukan oleh Harahap, Fajri, Syahputra, Rahmat, & Nababan, (2018) pendeteksian penyakit daun kelapa sawit dengan pengolahan citra menggunakan metode *Support Vector Machine* (SVM) dengan

judul “Identifikasi Penyakit Daun Tanaman Kelapa Sawit dengan Teknologi *Image Processing* Menggunakan Aplikasi *Support Vector Machine*” menghasilkan tingkat akurasi yang tinggi, yaitu mencapai 90%. Dengan pengolahan citra, identifikasi penyakit tumbuhan kelapa sawit hanya membutuhkan citra atau foto tanaman, kemudian sistem akan menganalisa pola berdasarkan proses *training* (Rasywir, Sinaga, & Pratama, 2020). Setelah data citra penyakit daun tanaman kelapa sawit di *training*, maka model data *training* akan disimpan untuk proses *testing* identifikasi penyakit daun tanaman kelapa sawit. Hasil *testing* akan disimpan menjadi *confussion matrix*. Sehingga nantinya diketahui seberapa akurat sistem mengidentifikasi penyakit pada daun tanaman kelapa sawit.

Metode yang cukup banyak dipakai untuk mengolah citra adalah metode *Convolutional Neural Network* (CNN), dari berbagai jenis metode yang terkait dalam pengolahan citra. *Convolutional Neural Network* (CNN) yang adalah perluasan dari *Multi Layer Perceptron* (MLP) dan adalah salah satu metode dari *Deep Learning*. Metode *Convolutional Neural Network* (CNN) memiliki hasil yang cukup berpengaruh dalam pengolahan citra. Selain itu penelitian tentang CNN dilakukan juga oleh Asrianda, Aidilof, & Pangestu, (2021) tentang deteksi jenis penyakit daun kelapa sawit secara *visual* menggunakan metode *Convolutional Neural Network* (CNN).

Metode *Convolutional Neural Network* (CNN) memiliki beberapa arsitektur, diantaranya terdapat arsitektur *Googlenet* (*Inception v1*) dan *Xception* (*Extream Inception*). Pada arsitektur *Googlenet* yang dikembangkan oleh *Google* yang telah melakukan *training* dengan jutaan citra, Arsitektur ini dipilih karena memiliki tingkat akurasi yang tinggi dan menjadi pemenang pada kompetisi ILSVRC pada tahun 2014 (Szegedy, et al., 2015) dan juga mendapat akurasi yang tinggi pada penelitian (Patani-, Pardikar, Doshi, Rodge, & Khachane, 2021) dengan judul *Plant Leaf Recognition and Disease Detection Using GoogLeNet* yang mendapat akurasi 99.35 %. ILSVRC adalah suatu kompetisi tahunan yang melombakan klasifikasi gambar dengan berbagai modifikasi CNN. Sedangkan arsitektur *Xception* pengembangan dari *Inception* terdahulu yang memiliki 36 *layer convolutional* yang membentuk basis jaringan ekstraksi fitur. Digunakan arsitektur *Xception* dikarenakan arsitektur ini menggunakan pendekatan *transfer learning*

yang populer (Darmatasia, 2020) dan pada penelitian yang dilakukan oleh (Tirtana, Febriani, Masrui, & Aisyah, 2021) juga mendapat akurasi yang mencapai 96,3 %.

Oleh karena itu, peneliti mencoba untuk mengidentifikasi jenis penyakit daun pada tanaman kelapa sawit melalui citra dengan mengimplementasi Arsitektur *Googlenet* dan *Xception*. Dengan demikian, penyakit pada daun tanaman kelapa sawit dapat dikenali berdasarkan teksturnya, Dan Penelitian ini diharapkan mampu menunjukkan arsitektur terbaik di antara keduanya dalam mengklasifikasikan citra tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan permasalahan sebagai berikut:

Bagaimana Implementasi dan Perbandingan arsitektur *Googlenet* dan *Xception* untuk identifikasi penyakit pada daun tanaman kelapa sawit.

1.3 Tujuan

Tujuan dari penelitian ini adalah implementasi arsitektur *Googlenet* dan *Xception* serta mampu menunjukan arsitektur terbaik antara *Googlenet* dan *Xception* untuk identifikasi penyakit pada daun tanaman kelapa sawit.

1.4 Batasan Masalah

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

1. Pada Penelitian ini, Citra daun yang digunakan adalah daun yang sehat dan daun yang terkena serangan hama ulat api dan ulat kantung.
2. Citra yang diinput adalah tekstur permukaan daun.
3. Format gambar yang digunakan adalah JPG.
4. Jarak pengambilan citra tidak lebih dari 30 cm.
5. Pengambilan citra harus dengan pencayahayaannya yang baik.

1.5 Manfaat Penelitian

Penelitian tentang Identifikasi penyakit pada daun tanaman kelapa sawit dengan arsitektur *googlenet* dan *xception* 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 pengimplementasian arsitektur *Googlnet* dan *Xception* untuk mengidentifikasi penyakit pada daun tanaman kelapa sawit.

2. Secara Praktis

a. Bagi Peneliti

- 1) Memahami cara kerja arsitektur *Googlnet* dan *Xception* dan wujud implementasinya.
- 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 univeristas dalam mengembangkan pengetahuan berkaitan pengolahan citra.

1.6 Metodologi Penelitian

1. Studi Literatur

Pada studi literatur ini dilakukan cara mencari dan mengumpulkan data dari berbagai referensi yang menunjang untuk mengidentifikasi penyakit pada daun tanaman kelapa sawit menggunakan arsitektur *Googlnet* dan *Xception*. Beberapa sumber digunakan, yaitu : Buku, Artikel, Jurnal. *Output* dari studi literatur yang didapat akan dibuat sebagai landasan dari dasar teori dalam penyusunan tugas akhir.

2. Pengumpulan Data

Pengumpulan Data yang dilakukan pada penelitian ini meliputi hal-hal yang berkaitan dengan penyakit daun tanaman kelapa sawit dan Berikut beberapa jenis pengumpulan data yang dilakukan.

a. Observasi

Observasi yang dilakukan dengan cara mengamati ciri atau gejala penyakit pada daun tanaman kelapa sawit.

b. Wawancara

Teknik pengumpulan data yang dilakukan dengan bertatap muka langsung kepada pakar dan melakukan wawancara mengenai data citra yang diambil. Pada penelitian kali ini wawancara dilakukan kepada pegawai yang menangani bidang Proteksi tanaman di Pusat Penelitian Kelapa Sawit Marihat, Siantar.

3. Perancangan

Setelah data-data yang dibutuhkan cukup, maka dilakukan perancangan sistem dan analisa batasan-batasan pada sistem Identifikasi penyakit pada daun tanaman kelapa sawit yang nantinya akan mendiolah dalam arsitektur *Googlenet* dan *Xception* yang di implementasikan pada penelitian.

4. Implementasi

Pada tahap implementasi, tugas akhir di uji cobakan dengan data sebenarnya yang digunakan pada sistem Identifikasi penyakit pada daun tanaman kelapa sawit.

5. Pengujian

Pada tahap pengujian kita dapat menganalisis apakah sistem dapat bekerja dengan lancar dan sesuai dengan hasil yang diinginkan pada penelitian.

6. Output

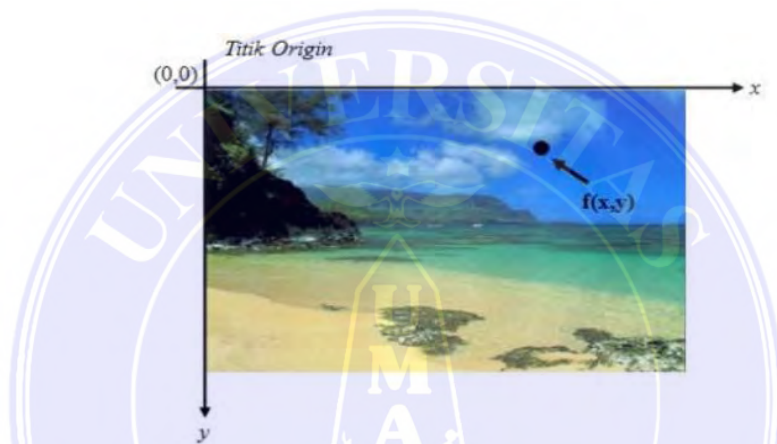
Output adalah sebuah program Identifikasi penyakit pada daun tanaman kelapa sawit yang dapat mengidentifikasi penyakit penyakit daun tersebut terhadap foto daun yang diproses. Dengan arsitektur *Googlenet* dan *Xception* diharapkan menghasilkan tingkat akurasi yang tepat pada citra daun yang dihasilkan.

BAB II

LANDASAN TEORI

2.1 Citra Digital

Citra adalah kumpulan piksel yang tersusun pada deret dua dimensi. Piksel memiliki arti yaitu representatif dari pemandangan yang memuat intensitas citra yang dinyatakan kedalam bilangan bulat. Piksel (0,0) ada pada sudut kiri atas citra, nilai x mengarah ke kanan dan nilai y mengarah ke bawah (Setiawan, Suryani, & Wiharto, 2014).



Gambar 2.1 Konsep Citra Digital (Zahrah, Saptono, & Suryani, 2016)

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan evolusi dari *Multilayer Perceptron* (MLP) yang ada pada *neural network* bertipe *feed forward* (tidak berulang). *Convolutional Neural Network* dirancang agar memproses informasi yang memiliki dua dimensi. CNN juga masuk kedalam golongan *Deep Neural Network* dikarenakan kedalaman *layer* yang besar juga sering diimplementasikan pada data citra (E. P, Wijaya, & Soelaiman, 2016). CNN sering dipakai untuk mengidentifikasi dan mengenali objek dalam gambar menganalisis gambar *visual*, yang memiliki vektor dengan ukuran tinggi nantinya menyangkutkan beberapa parameter untuk mendeteksi *layer*. Pada dasarnya, CNN tidak terlalu jauh berbeda dari *neural network* umumnya. CNN terdiri dari *neuron* yang mempunyai *weight*, *bias* dan *activation function*.

a. Convolutional Layer

Convolutional layer ialah 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 :

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

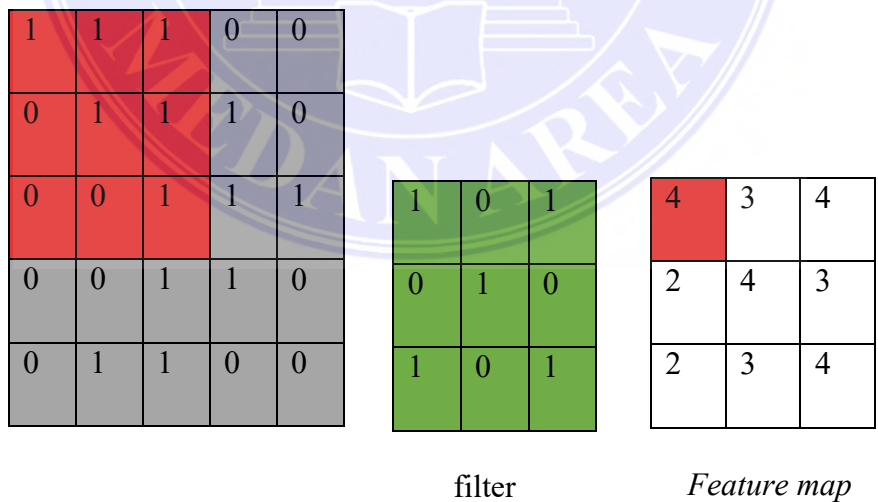
Keterangan :

$s(t)$ = 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

b. Rectification Linear Unit (ReLU)

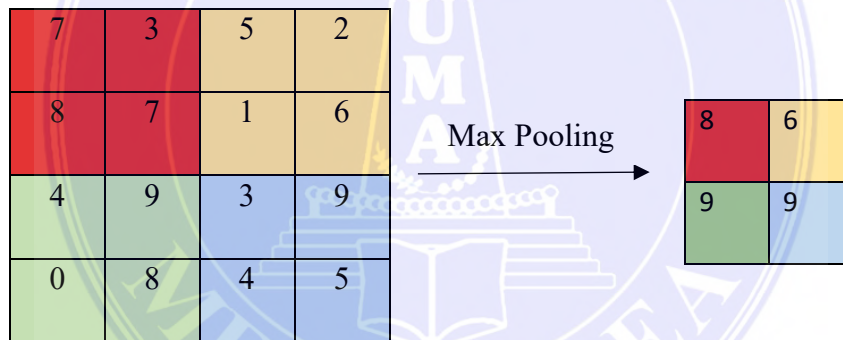
ReLU (Rectification Linear Unit) adalah cara untuk mengenali non-linearitas juga menambah dari kinerja dari suatu model. Rumus dan contoh operasi ReLU dapat dilihat pada persamaan 2.2 .

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

c. Pooling 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 2.3.

$$y = \max_{i,j}^{h,w} x_{i,j} \dots\dots\dots (2.3)$$



Gambar 2.3 Proses Max Pooling

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.

$$y = \sum x_i w_i + b \dots\dots\dots(2.4)$$

e. Softmax

Softmax adalah fungsi pengorganisasian yang bertujuan untuk dapat mempunyai hasil klasifikasi (Mahmud, Adiwijaya, & Faraby, 2019). *Point* kelas didapat dengan menggunakan aktivasi *softmax*. Fungsi aktivasi *softmax* dapat ditulis sebagai berikut:

$$S(y) = \frac{e^y}{\sum e^y} \dots\dots\dots(2.5)$$

2.3 Arsitektur

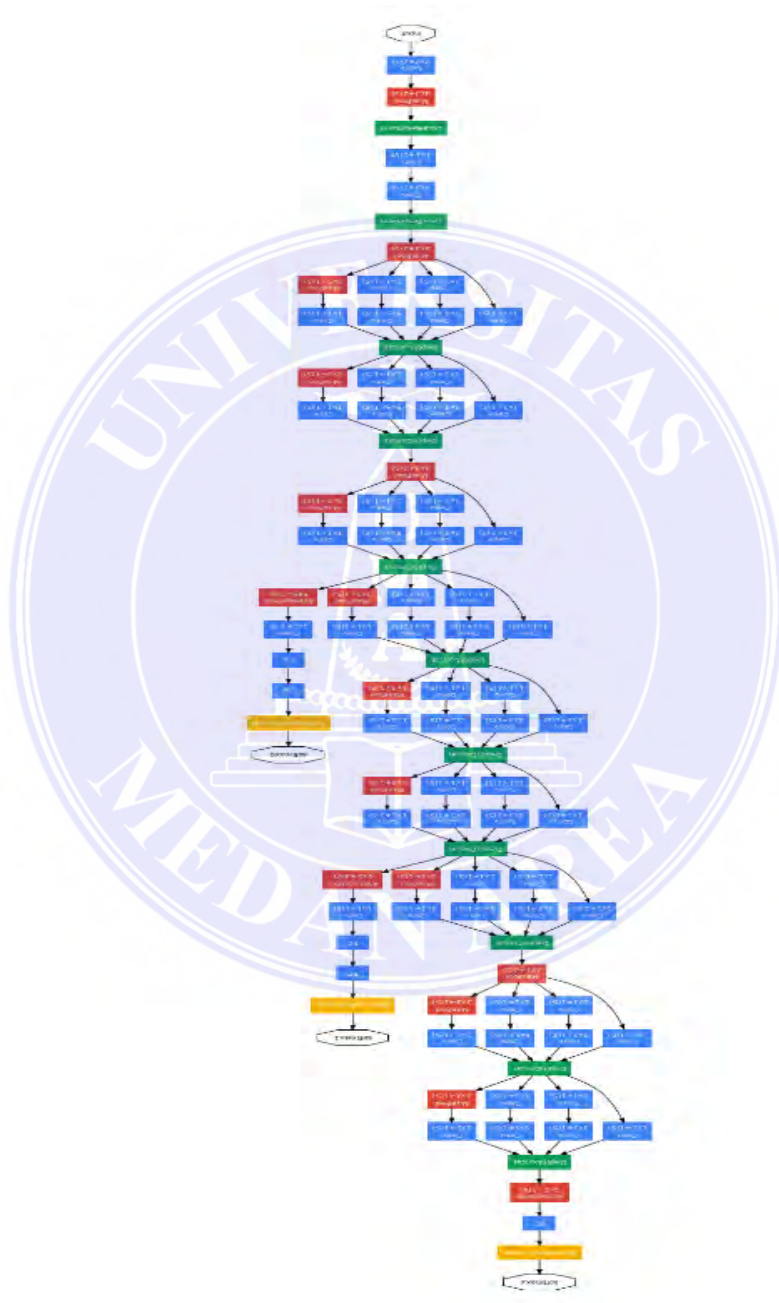
Imagenet Mengadakan pengembangan dari arsitektur *neural network* untuk klasifikasi objek pada kompetisi *ImageNet Large-Scale Visual Recognition Challenge*. Kompetisi pertama kali diadakan pada tahun 2012 dan dimenangkan oleh Alexnet yang telah diterapkan pada berbagai sistem *computer* yang lebih besar seperti *computer vision, object detection, video classification*, dan *object tracking*. Kesuksesan tersebut mendorong untuk dilakukan penelitian yang baru berfokus pada kinerja convolutional neural network. Pada tahun 2014, kualitas arsitektur meningkat secara drastic, VGG dan Googlenet menghasilkan kinerja yang lebih tinggi pada ILSVRC 2014 (Szegedy, et al., 2015).

2.3.1 Googlenet

Googlenet merupakan sebuah model dan arsitektur berdasarkan modifikasi arsitektur CNN. Arsitektur ini dikembangkan oleh *Google* dan telah mampu melakukan proses *training data* dengan jutaan gambar atau citra. Dikarenakan dasarnya adalah CNN, namun cara kerjanya tidak terlalu berbeda jauh dari modifikasi yang dikerjakan *Alexnet*.

Arsitektur *Googlenet* dapat dikenali melalui dengan memakai *layer inception* yang lebih dari satu *layer* dan biasanya lebih besar, yang akan berguna untuk menaikkan nilai hasil. *Googlenet* sanggup mengurangi tingkat kesalahan sampai 6,67% yang lebih unggul dari *alexnet* hanya 15,3%. *Googlenet* memakai

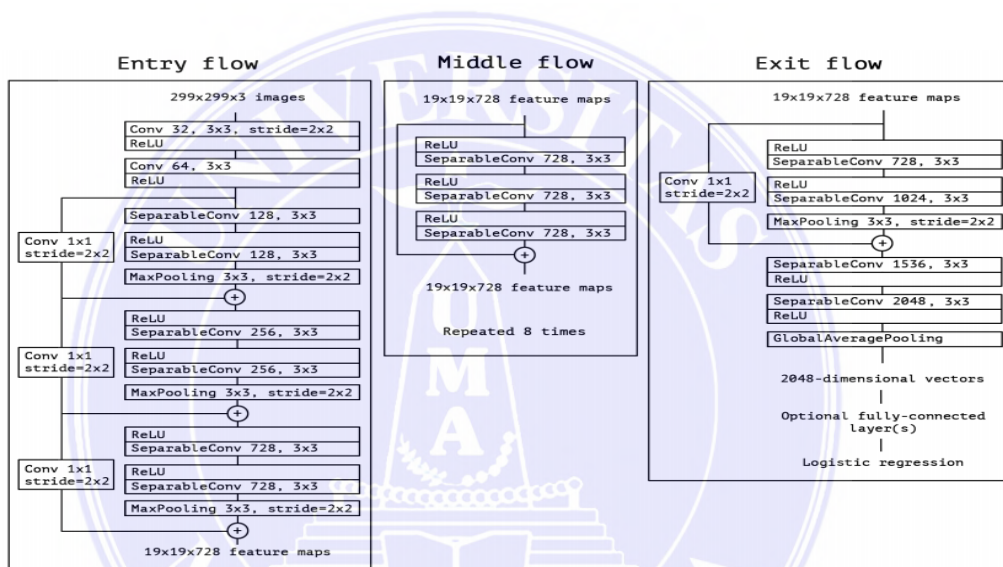
normalisasi *batch* yang digunakan agar mengatur *input* yang dilakukan beruntun, dan distorsi citra digunakan untuk beradaptasi ukuran sesuai keinginan. Kemudian, *Googlenet* melalui fase *inception* awalnya, nantinya ketika pada *layer* kelima, beberapa fitur akan menjalankan *fully connected neuron* (Hafifah, Rahman, & Asih, 2021).



Gambar 2.4 Arsitektur *Googlenet*
Sumber : (Glorot, Xavier, & Bengio, 2011)

2.3.2 Xception

Xception atau *Extreme Inception* adalah peningkatan dari *Inception* terdahulu yang mempunyai 36 *convolutional layer* yang menjadikan basis jaringan ekstraksi fitur. Lapisan konvolusional ini dipisahkan oleh jaringan residual yang menjalin satu sama lain (F. Chollet, 2017). Pada arsitektur ini, setelah blok konvolusional pertama dijalankan, pemisahan setiap saluran secara spasial dengan cara menggunakan konvolusi 1×1 , nantinya keluaran tersebut disimpan untuk mendapatkan korelasi *cross-channel* (Leonardo, Carvalho, Rezende, Zucchi, & Faria, 2018).



Gambar 2.5 Arsitektur *Xception*

Sumber : (Chollet, 2017)

2.4 Tanaman Kelapa Sawit

Kelapa sawit (*Elaeis guineensis Jacq.*) dari famili *Palmae* yang buahnya dapat diolah menjadi minyak. Perkebunan kelapa sawit menciptakan keuntungan besar berakibat banyaknya hutan dan perkebunan lama diubah menjadi perkebunan kelapa sawit. Perkebunan kelapa sawit di Indonesia tumbuh dengan sangat pesat, Kebun kelapa sawit dibudidayakan hampir di seluruh kepulauan nusantara, baik itu milik individu atau milik perusahaan negeri atau swasta (Satria, Poningsi, & Saputra, 2019). Hasil kelapa sawit ini dapat diolah pada industri tekstil, pangan, farmasi, kosmetik dan biodiesel. Karena itulah, Diperlukan adanya perawatan pada

tanaman sawit, dan penting untuk mengenali jenis penyakit apa saja yang dapat menyerang tanaman kelapa sawit, terutama pada daun.

2.5 Penyakit Pada Daun Kelapa Sawit

Kesehatan dan tumbuh kembang tanaman kelapa sawit ialah suatu hal yang harus dirawat dan diperhatikan secara teliti. Seluruh komponen dari tanaman kelapa sawit bisa saja menjadi sasaran dari penyakit tanaman dan hama, salah satunya pada segmen daunnya. (Defitri, 2015).

Kejadian tak jarang terjadi di perkebunan kelapa sawit ialah terjadinya perputaran ekosistem hama ulat api atau ulat kantung, hama-hama ini perlu dilakukan pengendalian secara teratur. Walau tidak mematikan tanaman, hama ini sangat merugikan secara ekonomi, pada daun yang telah dirusak akan sangat mengganggu proses fotosintesis tanaman kelapa sawit, yang pada nantinya akan mengurangi produktivitas kelapa sawit (Yuliani, Labellapansa, & Yulianti, 2019).

2.5.1 Ulat Kantung

Ulat kantung (*Cremastopsyche pendula*) adalah satu dari beberapa jenis ulat yang menginfeksi dan merugikan perkebunan kelapa sawit. *Metisa plana* mirip dengan ulat jenis ini, yang bersifat polifag. Tak hanya menyerang daun kelapa sawit kerap juga menyerang daun Kakao, Sagu, Acacia, Kopi, Teh dan Alzazia. sering juga menginfeksi secara bersamaan dengan *Metisa plana*. Kantung ulat ini langsung melekat di permukaan daun (Pangaribuan, Marheni, & Lubis, 2017).



Gambar 2.6 Ulat kantung dan dampaknya pada daun kelapa sawit

2.5.2 Ulat Api

Ulat api ialah salah satu jenis ulat pemakan daun kelapa sawit yang paling sering dijumpai dan menyebabkan kerusakan yang cukup besar di perkebunan kelapa sawit. Adapun jenis ulat api yang menyerang tanaman kelapa sawit yang belum menghasilkan, yaitu *Setothosea asigna*, *Setora nitens*, *Darna trima*, *Darna diducta*, *Darna brodley*, *Susi malayana*, *Birthose bisura*, *Thosea vetusta* dan *Olona gater* (Lukmana & Elafia, 2017).



Gambar 2.7 Ulat Api dan dampaknya pada daun kelapa sawit

2.6 Penelitian Terdahulu

Penelitian terdahulu dapat dilihat pada tabel 2.1.

No	Judul	Penulis & Tahun	Metode	Keterangan
1	Penerapan Metode Forward Chaining Pada Sistem Pakar Identifikasi Penyakit Tanaman Kelapa Sawit	(Marcelina, Yulianti, & Mair, 2022)	Forward Chaining	Identifikasi Penyakit Tanaman Kelapa Sawit menggunakan Metode Forward Chaining dengan sistem pakar, dengan memberikan kuisisioner kepada 35 orang responden untuk mendapat penilaian terhadap sistem. Hasil

				kuisisioner (Pakar) diperoleh nilai kepuasan sebesar 89.2%
2	Identifikasi Penyakit Daun Tanaman Kelapa Sawit dengan Teknologi Image Processing Menggunakan Aplikasi Support Vector Machine	(Harahap, Fajri, Syahputra, Rahmat, & Nababan, 2013)	Support Vector Machine (SVM)	Menggunakan SVM sebagai pengklasifikasian data penyakit pada daun kelapa sawit. Mendeteksi dengan pengenalan citra pada daun kelapa sawit. Dan memperoleh akurasi pengujian sebanyak 90%.
3	Machine Learning for Detection of Palm Oil Leaf Disease Visually using Convolutional Neural Network Algorithm	(Asrianda, Aidilof, & Pangestu, 2021)	Convolutional Neural Network (CNN)	Pendeteksian penyakit daun tanaman kelapa sawit menggunakan algoritma CNN. <i>Accuracy</i> yang dihasilkan menggunakan <i>confusion matrix</i> adalah 69%.
4	Plant Leaf Recogniton And Disease Detection	(Patani-, Pardikar, Doshi, Rodge, &	Convolutional Neural Network (CNN),	Mengklasifikasi daun tanaman menggunakan Googlenet. Mengenali jenis penyakit

	Using GoogLeNet	Khachane, 2021)	GoogLeNet	berdasarkan gambar.Dan mendapat nilai akurasi sebesar 99.35%.
5	Herbify : Aplikasi Perangkat Bergerak Berbasis Komputasi Awan Untuk Mengidentifikasi Tanaman Herbal Indonesua Menggunakan CNN Model Xception	(Tirtana, Febriani, Masrui, & Aisyah, 2021)	Convolutional Neural Network, Xception	Aplikasi perangkat mobile berbasis komputasi <i>cloud</i> dengan menggunakan algoritma CNN model Xception. Dan mendapat nilai akurasi yang sangat baik yaitu sebesar 96,3%.

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Dalam penelitian ini, menggunakan peralatan dan bahan pendukung agar penelitian ini berjalan lancar. Adapaun perangkat keras dan perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut:

3.1.1 Perangkat Keras

Tabel 3.1 Perangkat keras

No	Perangkat Keras	Deskripsi
1	Device	Laptop Asus Vivobook X442UX
2	Processor	Intel Core i5-8250U @1.60 GHz
3	SSD	240 GB
4	Ram	8.00 GB

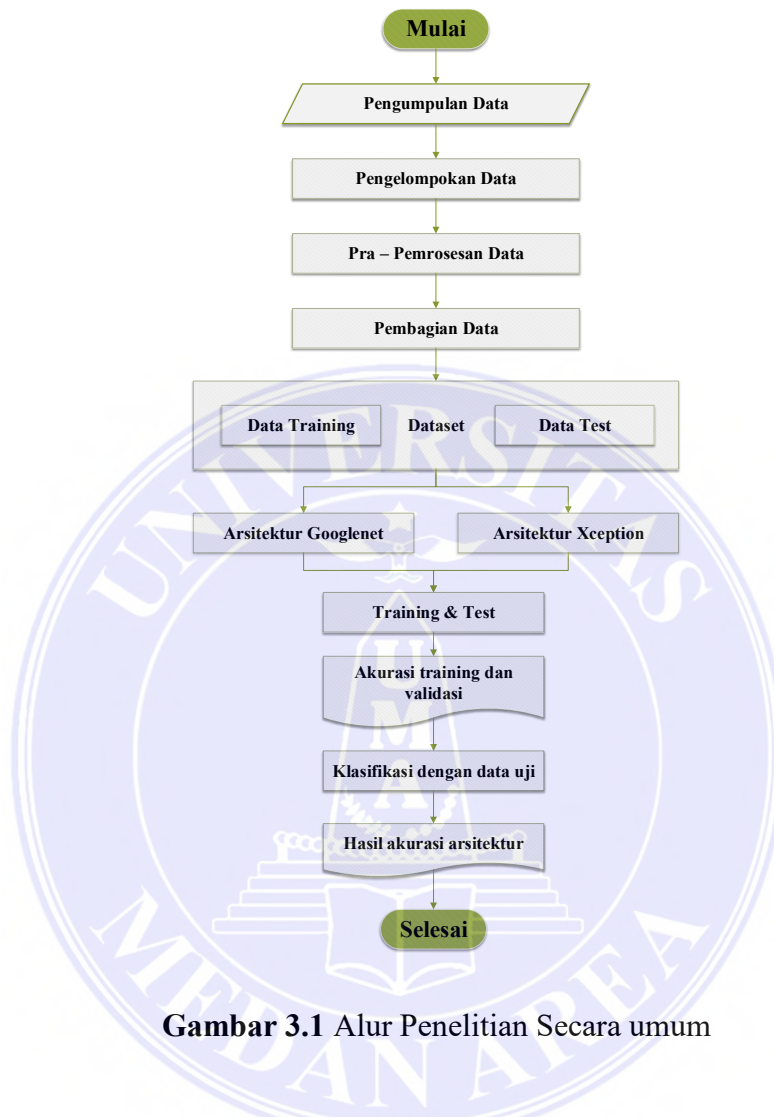
3.1.2 Perangkat Lunak

Tabel 3.2 Perangkat Lunak

Perangkat Lunak	Deskripsi
Sistem Operasi	Windows 10 Home Single Language 64-bit
Google Colab	<ul style="list-style-type: none"> - Runtime : GPU - Ram :12 GB - Disk : 78 GB
Phyton	Bahasa Pemrograman
Tensor Flow	Library

3.2 Diagram Alur Penelitian

Diagram Alur Penelitian ini diilustrasikan pada gambar dibawah ini.

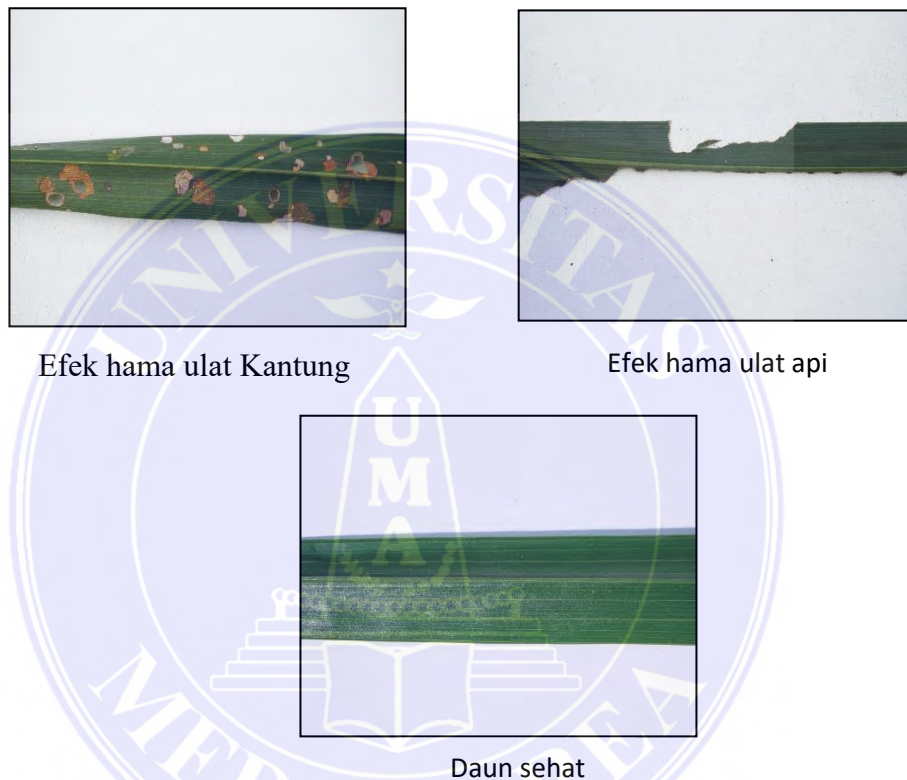


Gambar 3.1 Alur Penelitian Secara umum

Penelitian ini dimulai dengan melakukan mengumpulkan data gambar untuk dikumpulkan menjadi *dataset*. Setelah semua sudah terkumpul, data dikelompokkan sesuai *class* nya masing-masing. Dilanjutkan dengan pra-pemrosesan data gambar dengan meng*compress* ukuran gambar agar tidak terlalu besar, kemudian membagi *dataset* menjadi data *training* dan data *test*. Setelah itu melakukan perancangan arsitektur yang akan digunakan yaitu *googlenet* dan *xception*. Tahap selanjutnya adalah tahapan *training* dan *testing* data hingga menghasilkan akurasi *training* dan validasi kemudian diklasifikasikan dengan data uji agar diketahui hasil akurasi arsitektur.

3.3 Metode Pengumpulan Data

Dataset diambil dengan melakukan pengamatan secara langsung yaitu mengumpulkan citra daun kelapa sawit di Kebun Pohon Induk Pusat Penelitian Kelapa Sawit, Dolok Baja, Kecamatan Tanah Jawa, Kabupaten Simalungun, Provinsi Sumatera Utara. Serta berdiskusi dengan pegawai Pusat Penelitian Kelapa Sawit (PPKS) yang menangani dibidang Proteksi Tanaman.



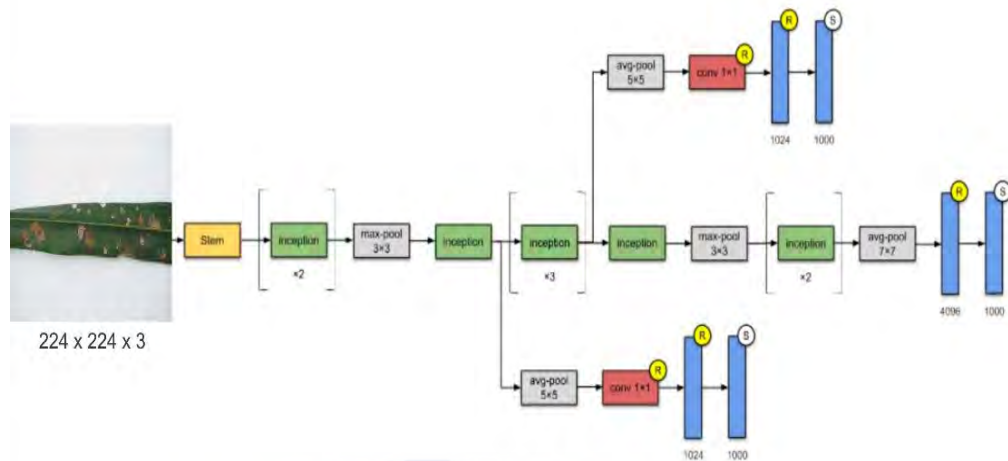
Gambar 3.2 Citra Daun Kelapa Sawit

3.4 Rancangan Arsitektur

Selanjutnya akan dilakukan perancangan model untuk pelatihan yang dibuat. Pada penelitian ini, Peneliti memakai metode *Transfer Learning* yang melakukan cara *training* dengan memakai model yang sudah dilatih terlebih dahulu.

3.4.1 Googlenet

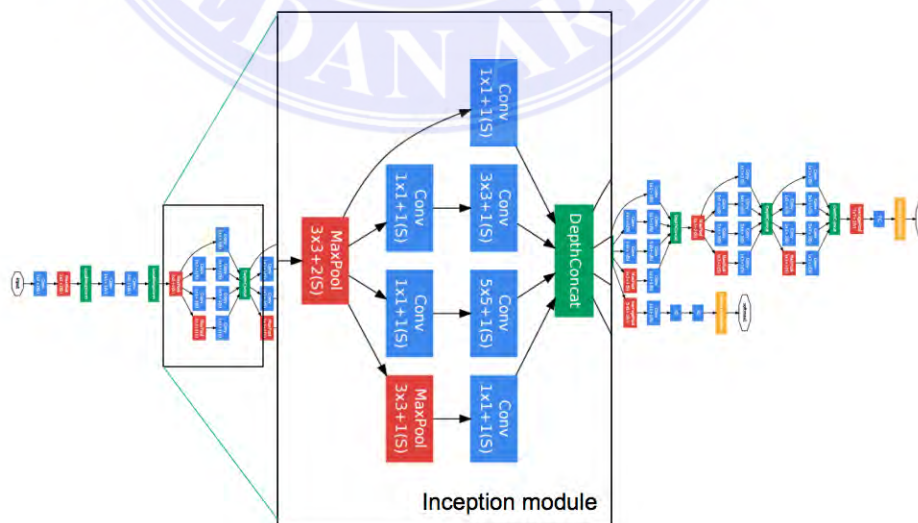
Berikut ini adalah gambaran arsitektur dari *googlenet* dengan input citra 224 x 224.



Gambar 3.3 Simulasi arsitektur *Googlenet*

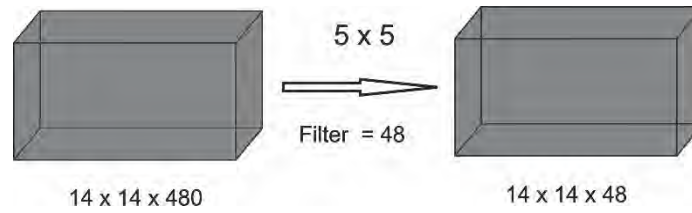
Googlenet memiliki *layer* dan *convolution* dengan total yang banyak. Secara singkat, *Googlenet* memiliki 5 *layer* sampai 13 *layer*. Jika *Alexnet* hanya memiliki 5 *layer* dan dibuat satu dengan *fully connected*, maka *googlenet* melintasi 5 fase *inception* terlebih dahulu. Pada saat *layer* kelima, sejumlah fitur melakukan *fully Conected Neuron*.

Googlenet memiliki 9 *Inception* modul yang ditumpuk secara linear. Arsitektur ini memiliki 22 *layer*, *googlenet* menggunakan *Average pooling global* pada akhir *inception* modul.



Gambar 3.4 *Inception* Modul pada Arsitektur *Googlenet*

Sumber : (Szegedy, et al., 2015)



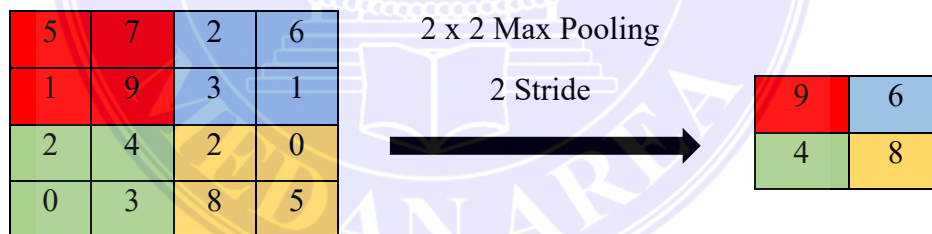
Gambar 3.5 Konvolusi 5 x 5

Misalnya, Jika kita ingin melakukan konvolusi 5x5 dengan 48 filter tanpa menggunakan konvolusi 1x1.

Total Jumlah Operasi : $(14 \times 14 \times 48) \times (5 \times 5 \times 480) = 112,9 \text{ M}$

Pada modul *Inception* 1×1 , 3×3 , 5×5 Konvolusi dan 3×3 *max pooling* dilakukan secara paralel pada *input* dan *outputnya* ditumpuk bersama untuk menghasilkan *output* akhir.

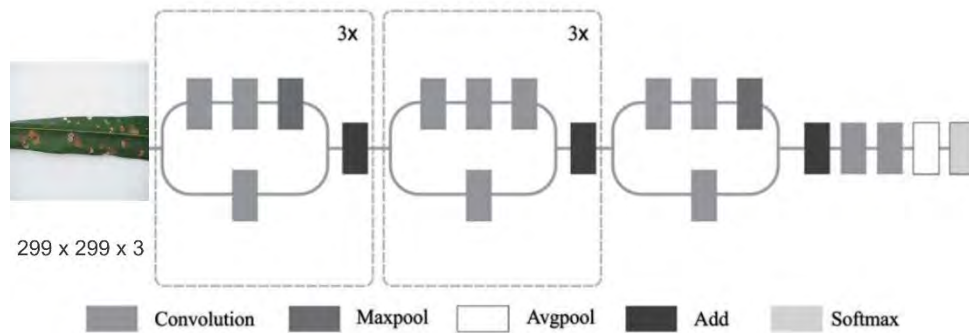
Pada bagian *max pooling* , menghitung nilai terbesar pada setiap blok, *max pooling* berfungsi mengurangi jumlah parameter ke depan karena beban komputasi dan Piksel bernilai lebih tinggi adalah yang paling aktif dan karenanya digunakan dalam operasi ini.



Gambar 3.6 Max Pooling

3.4.2 Xception

Berikut ini adalah gambaran arsitektur dari xception, Dengan input citra 299×299 .

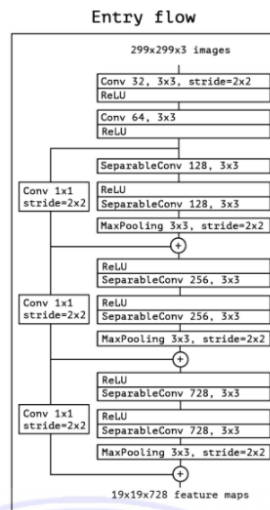


Gambar 3.7 Simulasi arsitektur *Xception*

Sumber : (Banumathi, et al., 2021)

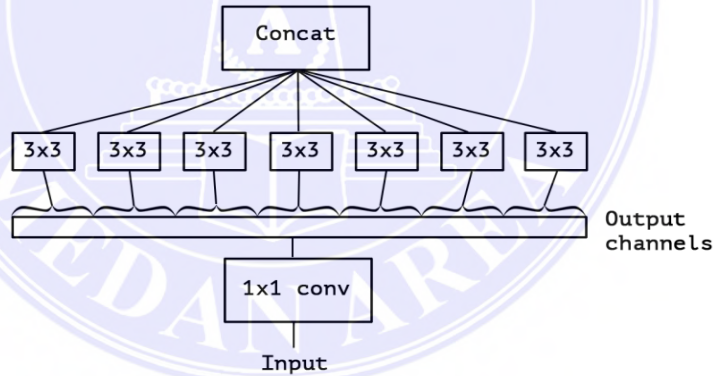
Arsitektur *Xception* memiliki 36 tahap konvolusi, dengan input citra berukuran 299 x 229 yang kemudian di konvolusi dengan filter 3 x 3 stide 2. dimana *inception* modul diganti dengan konvolusi yang dapat dipisah secara mendalam. *Xception* menggiring hipotesis secara *Extreme*, Seperti korelasi lintas channel (*Cross feature map*) dialihkan ke konvolusi 1 x 1 yang mengakibatkan korelasi special dalam setiap channel dialihkan ke konvolusi 3 x 3 biasa atau konvolusi 5 x 5. Maksud dari *Extreme* adalah melakukan konvolusi 1 x 1 setiap channel lalu melakukan konvolusi 3 x 3 untuk setiap output yang *identic*.

Pada gambar 2.3 Arsitektur *xception* , modul *xception* memiliki 3 bagian utama yaitu, aliran masuk, aliran tengah (looping 8 kali) , dan aliran keluar.



Gambar 3.8 Entry Flow Arsitektur *Xception*

Aliran masuk memiliki dua blok lapisan konvolusional yang diikuti dengan aktivasi ULT. Ada juga berbagai lapisan konvolusional yang dapat dipisahkan. Ada juga lapisan Max Pooling. Disini dimulai dengan ukuran gambar 299 x 299 x 3, dan setelah aliran entri, maka didapatkan ukuran gambar 19 x 19 x 728.



Gambar 3.9 *Xception* dengan satu konvolusi spasial per output

Sumber: (Chollet, 2017)

Pada umumnya ada dua perbedaan antara *Inception* dan *Xception* dalam *deftwise separable convolution*. Pertama dari segi urutan operasi, pada *deftwise separable convolution* digunakan *channel-wise* konvolusi spasial terlebih dahulu nantinya diteruskan dengan konvolusi 1x1, namun pada *Inception* dilakukan

konvolusi 1x1 terlebih dahulu sebelum mengimplementasi *depthwise separable convolution*.

3.5 Data Training

Pada Tahap selanjutnya adalah melakukan training dataset dengan menggunakan arsitektur *Googlenet* dan *Xception*. 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. Parameter yang digunakan di inisialisasi untuk proses training ini adalah *Learning rate*, *Batch size*, *Epoch*, dan *Optimizer*.

3.6 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* dan *Xception*. Setelah itu, sistem akan menampilkan akurasi berupa prediksi jenis penyakit pada daun yang *diinput*.

3.7 Hasil

Pada tahap hasil akan dilakukan perhitungan untuk mencari nilai precision, recall, dan F1-score sebagai berikut :

a) Accuracy

Merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data.

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

b) Precision

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

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

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) F1 Score

F1 Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan.

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

Keterangan :

TP = true positive

FP = false positive

FN = false negative

TN = true negative

TP arti sebagai data positif yang diduga sebagai positif dan TN memiliki arti sebagai data negatif yang di diduga sebagai negatif. Sedangkan FN ialah lawan dari TP yaitu data positif yang diduga sebagai negatif dan FP, lawan dari TN, yaitu data negatif yang diduga positif. Untuk *confusion matrix* bisa kita lihat pada table 3.3

Tabel 3.3 Confusion Matrix

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

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Implementasi arsitektur *Googlenet* dan *Xception* untuk mengidentifikasi penyakit pada daun tanaman kelapa sawit bekerja dengan baik.
2. Arsitektur *Googlenet* memiliki akurasi sebesar 93,22% dan *Xception* memiliki akurasi sebesar 99,72%. Keduanya memiliki epoch terbaik dengan epoch 15, *Googlenet* dengan batch size 32 dan *Xception* dengan batch size 64. Optimizer yang paling baik untuk *Googlenet* dan *Xception* pada penelitian ini adalah RMSprop, dan learning rate yang baik untuk *Googlenet* adalah 0,009 dan *Xception* 0,001.
3. Berdasarkan tingkat akurasi pada penelitian ini didapat bahwa arsitektur *Xception* adalah arsitektur terbaik dibandingkan arsitektur *Googlenet*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, peneliti memiliki beberapa saran sebagai berikut:

1. Sebaiknya gunakan lebih banyak dataset dan perbanyak steps training agar hasil identifikasi yang diperoleh lebih baik
2. Identifikasi jenis penyakit daun tanaman kelapa sawit ini dapat dikembangkan kedepannya untuk mendeteksi jenis penyakit lainnya pada tanaman lain.

DAFTAR PUSTAKA

- Asrianda, Aidilof, H. A., & Pangestu, Y. (2021). Machine Learning for Detection of Palm Oil Leaf Disease Visually using Machine Learning for Detection of Palm Oil Leaf Disease Visually using. *JITE (Journal of Informatics and Telecommunication Engineering)*.
- Banumathi, J., Muthumari, A., Dhanasekaran, S., Rajasekaran, S., Pustokhina, I. V., Pustokhin, D. A., & Shankar, K. (2021). An Intelligent Deep Learning Based Xception Model for Hyperspectral Image Analysis and Classification. *Computers, Materials & Continua*.
- Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *Google, Inc.*
- Darmatasia. (2020). DETEKSI PENGGUNAAN MASKER MENGGUNAKAN XCEPTION TRANSFER LEARNING. *JURNAL INSTEK*.
- Defitri, Y. (2015). Identifikasi Patogen Penyebab Penyakit Tanaman Sawit (*Elaeis guineensis* Jacq.) Di Desa Bertam Kecamatan Jambi Luar Kota. *Jurnal Ilmiah Universitas Batanghari Jambi*.
- Glorot, X., Xavier, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. *In Proceedings of the 14th International In Proceedings of the 14th International*.
- Hafifah, F., Rahman, S., & Asih, M. S. (2021). Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN). *TIN: Terapan Informatika Nusantara*.
- Harahap, L. A., Fajri, R. I., Syahputra, M. F., Rahmat, R. F., & Nababan, E. B. (2013). Identifikasi Penyakit Daun Tanaman Kelapa Sawit dengan Teknologi Image Processing Menggunakan Aplikasi Support Vector Machine. *Talenta Conference Series: Agricultural and Natural Resources (ANR)*.
- Jais, I. K., Ismail, A. R., & Nisa, S. Q. (2019). Adam Optimization Algorithm for Wide and Deep Neural Network. *Knowledge Engineering and Data Science (KEDS)*, Vol 2, No 1.
- Leonardo, M. M., Carvalho, T. J., Rezende, E., Zucchi, R., & Faria, F. A. (2018). Deep Feature-based Classifiers for Fruit Fly Identification (Diptera: Tephritidae). *SIBGRAPI Conference on Graphics, Patterns and Images*.
- Lukmana, M., & Elafia, N. (2017). Tingkat Serangan Hama Ulat Api Pada Tanaman Kelapa Sawit (*Elaeis guineensis* Jacq.) Fase Belum Menghasilkan Di PT.Barito Putera Plantation . *Jurnal Budidaya Tanaman Perkebunan Politeknik Hasnur*.

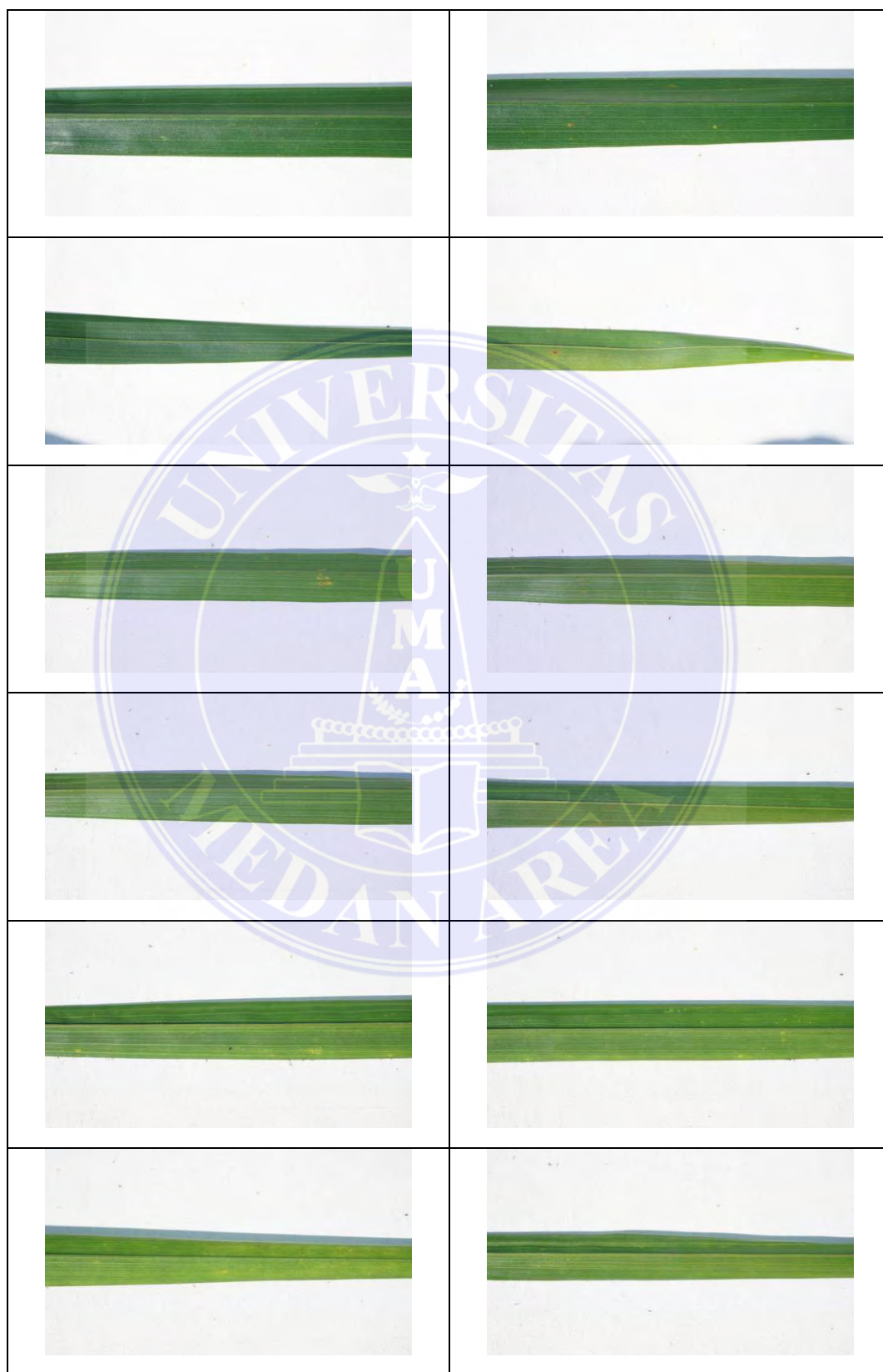
- Mahmud, K. H., Adiwijaya, & Faraby, S. A. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network. *e-Proceeding of Engineering*.
- Mahua, M. S. (2018). Sistem Pakar Untuk Mendiagnosis Penyakit Tanaman Jeruk (Limau) Menggunakan Metode Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*.
- Pangaribuan, R., Marheni, & Lubis, L. (2017). Tingkat Serangan Ulat Kantung Cremastopsyche pendula Joannis pada Tanaman Kelapa Sawit Menghasilkan dan Belum Menghasilkan di Rambong Sialang Estate PTPP. London Sumatera Indonesia. *Jurnal Agroekoteknologi FP USU*.
- Patani-, A., Pardikar, I., Doshi, P., Rodge, S., & Khachane, S. (2021). Plant Leaf Recognition and Disease Detection Using GoogLeNet. *JETIR*.
- Pratiwi, N. K., Ibrahim, N., Fu'adah, Y. N., & Rizal, S. (2021). Deteksi Parasit Plasmodium pada Citra Mikroskopis Hapusan Darah dengan Metode Deep Learning. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, Vol. 9, No. 2, Halaman 306 - 317.
- Rasywir, E., Sinaga, R., & Pratama, Y. (2020). Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN). *Paradigma – Jurnal Informatika dan Komputer*.
- Saragih, R., Jean Cross Sihombing, D., & Rahmi, E. (2018). Sistem Pakar Diagnosa Penyakit Kelapa Sawit Menggunakan Metode Dempster Shafer Berbasis Web. *Journal of Information Technology and Accounting*.
- Satria, D., Poningsi, & Saputra, W. (2019). Sistem Pakar Mendeteksi Penyakit Tanaman Kelapa Sawit Menggunakan Metode Forward Chaining. *Prosiding Seminar Nasional Riset Information Science (SENARIS)*.
- Setiawan, A., Suryani, E., & Wiharto. (2014). Segmentasi Citra Sel Darah Merah Berdasarkan Morfologi Sel Untuk Mendeteksi Anemia Defisiensi Besi. *JURNAL ITSMART*.
- Supriyanto, G., Jusak, & Sudarmaningtyas, P. (2014). Sistem Pakar Diagnosis Penyakit Pada Tanaman Kelapa Sawit Menggunakan Metode Certainly Factor. *Jurnal Sistem Informasi*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going Deeper with Convolutions. *Computer Vision Foundation*.
- Tirtana, A., Febriani, M. G., Masrui, D. I., & Aisyah, A. A. (2021). Herbify: Aplikasi Perangkat Bergerak Berbasis Komputasi Awan Untuk Mengidentifikasi Tanaman Herbal Indonesia Menggunakan Cnn Model Xception. *Jurnal Ilmiah Edutic*.

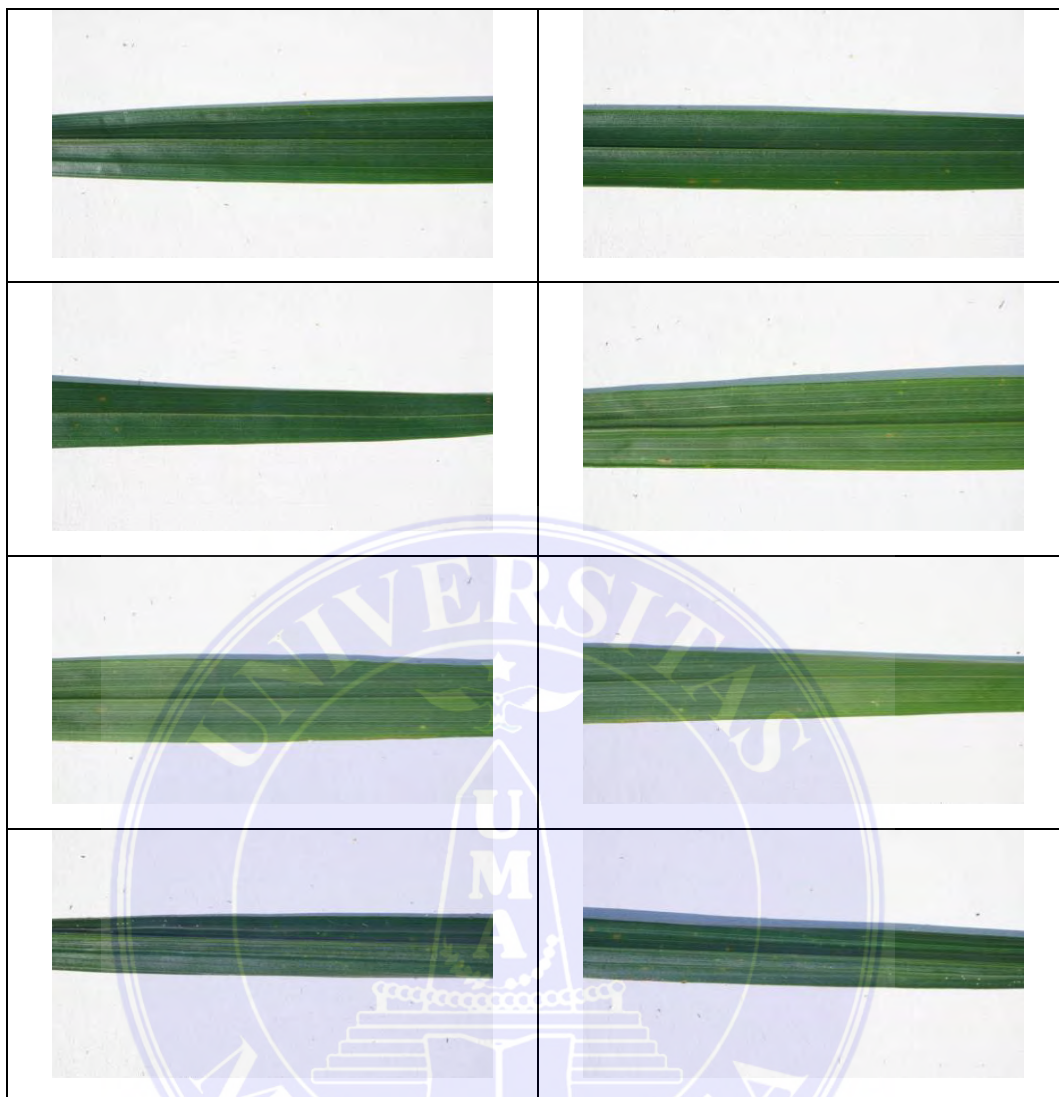
- Ukto, A. I. (2017). Analisis Citra Drone Untuk Monitoring Kesehatan Tanaman Kelapa Sawit. *Jurnal Agroteknose*.
- Widians, J., Taruk, M., Fauziah, Y., & Setyadi, H. (2019). Decision Support System on Potential Land Palm Oil Cultivation using Promethee with Geographical Visualization. *Cultivation using Promethee with Geographical Visualization*.
- 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*.
- Zahrah, S., Saptono, R., & Suryani, E. (2016). Identifikasi Gejala Penyakit Padi Menggunakan Operasi Morfologi Citra. *Seminar Nasional Ilmu Komputer*.



LAMPIRAN

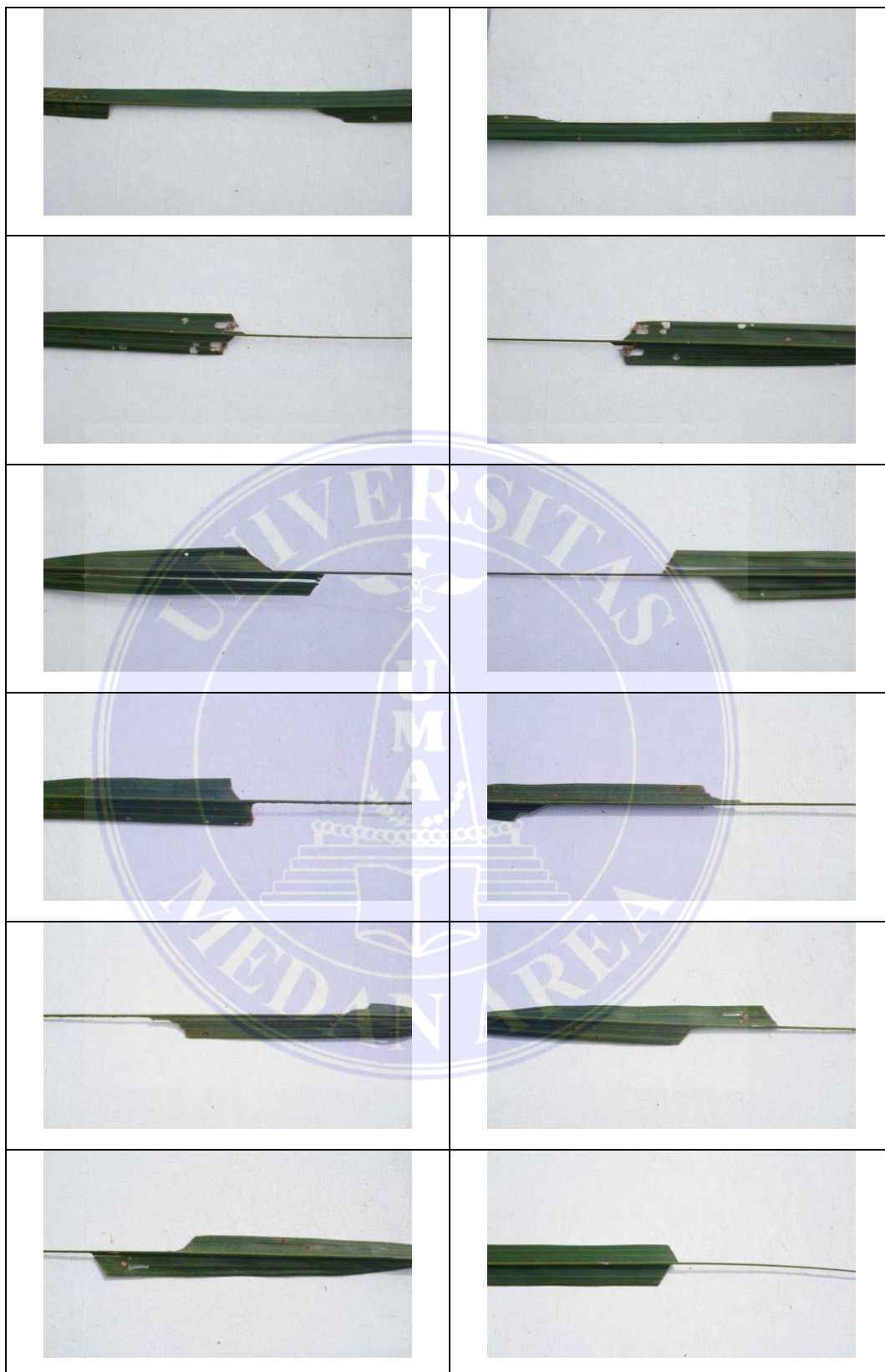
1. Dataset Daun Sehat

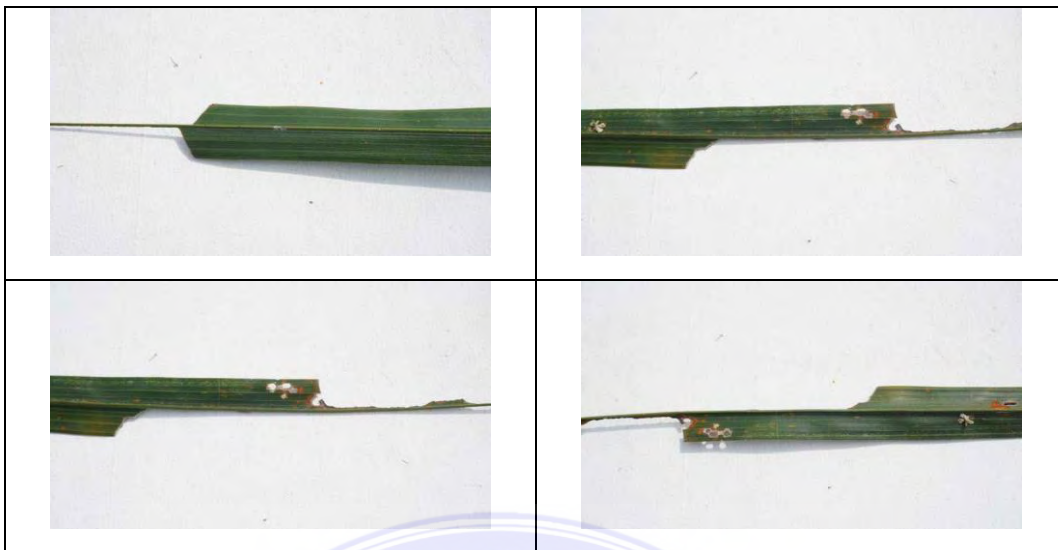




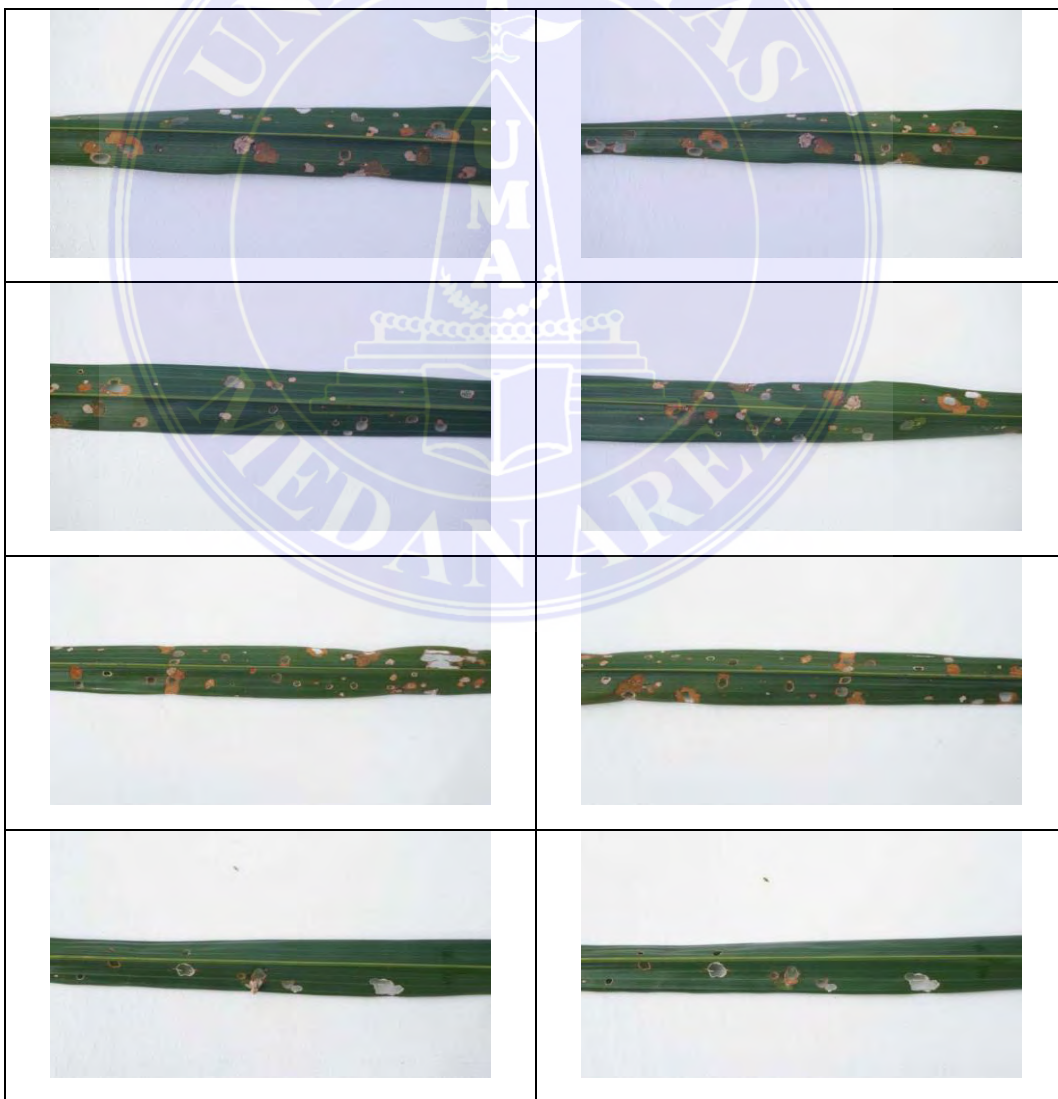
2. Dataset Daun Terkena Ulat Api

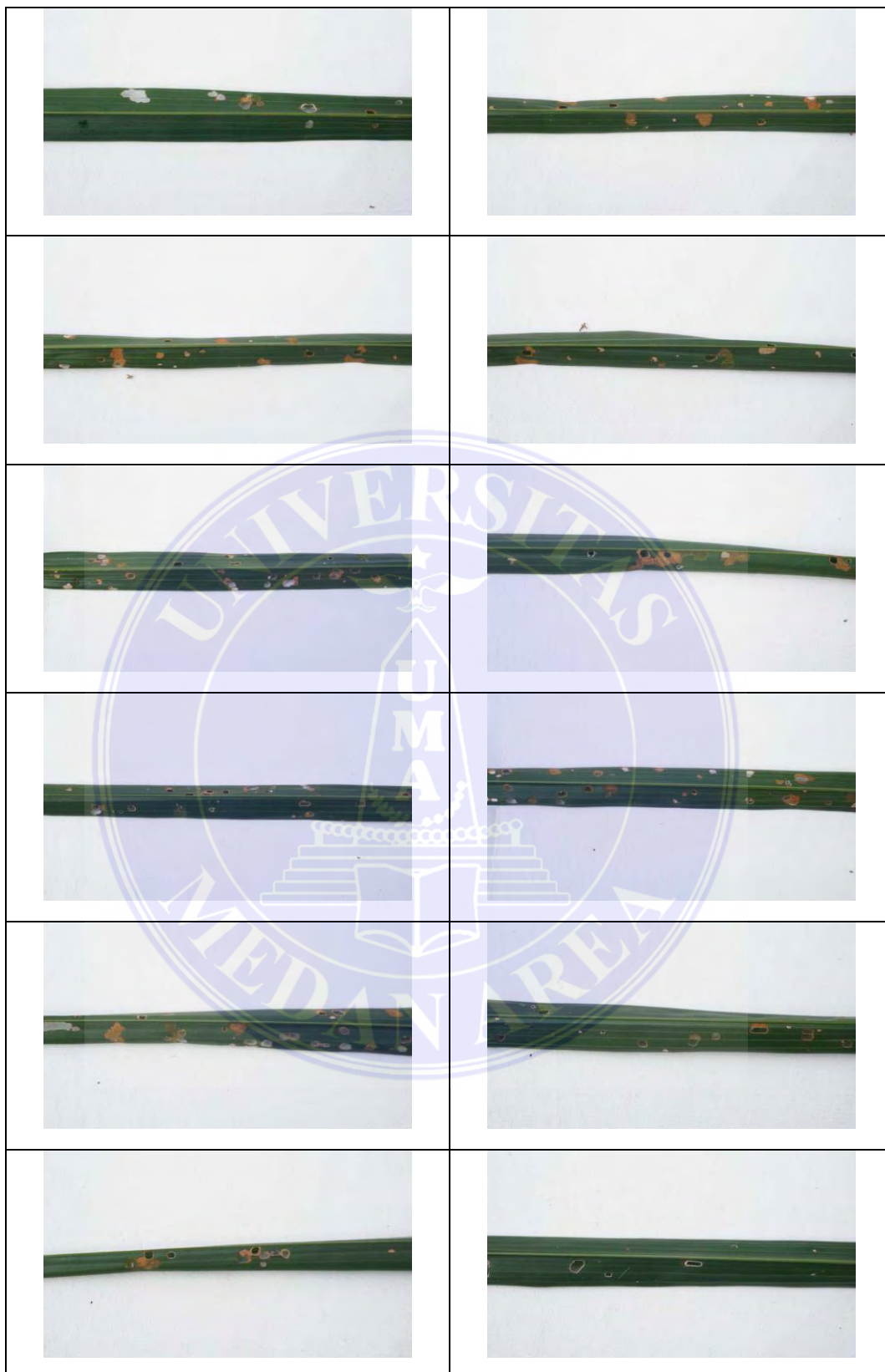






3. Dataset Daun Terkena Ulat Kantung





4. Sourcode

a) Import

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

from tensorflow.keras.utils import plot_model
from keras.applications.xception import Xception
from keras.preprocessing import image
from keras.layers import Conv2D, MaxPooling2D,
GlobalAveragePooling2D
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 keras.preprocessing.image 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

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

b) Hyper Paramater

```
EPOCHS = 15
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((224, 224))
```

```

image_size = 0
directory_root = '/content/drive/My Drive/Penyakit Daun'
width=224
height=224
depth=3
    
```

c) 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
    
```

d) Pemanggilan 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 Penyakit_Daun_folder in root_dir :
        Penyakit_Daun_Kelapa_Sawit_folder_list = listdir(f"{direct
ory_root}/{Penyakit_Daun_folder}")

        for Penyakit_folder in Penyakit_Daun_Kelapa_Sawit_folde
r_list :
            # remove .DS_Store from list
            if Penyakit_folder == ".DS_Store" :
                Penyakit_Daun_Kelapa_Sawit_folder_list.remove(Pen
yakit_folder)

        for Penyakit_Daun_Kelapa_Sawit_folder in Penyakit_Daun
_Kelapa_Sawit_folder_list:
    
```

```

        print(f"[INFO] Processing {Penyakit_Daun_Kelapa_Sawit_folder} ...")
        Penyakit_Daun_Kelapa_Sawit_image_list = listdir(f"{directory_root}/{Penyakit_Daun_folder}/{Penyakit_Daun_Kelapa_Sawit_folder}")

        for single_Penyakit_Daun_Kelapa_Sawit_image in Penyakit_Daun_Kelapa_Sawit_image_list :
            if single_Penyakit_Daun_Kelapa_Sawit_image == ".DS_Store" :
                Penyakit_Daun_Kelapa_Sawit_image_list.remove(single_Penyakit_Daun_Kelapa_Sawit_image)

            for image in Penyakit_Daun_Kelapa_Sawit_image_list[:410]:
                image_directory = f"{directory_root}/{Penyakit_Daun_folder}/{Penyakit_Daun_Kelapa_Sawit_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(Penyakit_Daun_Kelapa_Sawit_folder)
            print("[INFO] Image loading completed")
    except Exception as e:
        print(f"Error : {e}")

```

e) Train & Test

```

x_train, x_test, y_train, y_test = train_test_split(np_image_list,
image_labels, test_size=0.3, random_state = 42)

```

f) Data Generator

```

aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=a.2, horizontal_flip=True,
    fill_mode="nearest")

```

g) 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

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

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

    # Konvolusi 1 × 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 × 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 d
ikurangi
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_p
roj], axis=3, name=name)

return output

```

h) 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)
    # Konvolusi
    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 Average 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_v
1')
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())
```

i) Modul Xception

```
#### MODEL - BOTTLENECK FEATURES - OPTIMIZER
from keras.applications import xception
from keras.layers import GlobalAveragePooling2D, Dense, BatchNormalization, Dropout
from tensorflow.keras.optimizers import RMSprop, Adam
from keras.models import Model, Input

# Download and create the pre-trained Xception model for transfer learning
base_model = xception.Xception(weights='imagenet', include_top=False, input_tensor=Input(shape = (width, height, depth)))

# add a global spatial average pooling layer
x = base_model.output
x = BatchNormalization()(x)
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dropout(0.5)(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
# and a logistic layer -
- let's say we have NUM_CLASSES classes
predictions = Dense(n_classes, activation='softmax')(x)
# this is the model we will train
model = Model(inputs=base_model.input, outputs=predictions)
# first: train only the top layers (which were randomly initialized)
)
# i.e. freeze all convolutional Xception layers
for layer in base_model.layers:
    layer.trainable = False
model.summary()
```

j) Model Fit Generator

```

opt = 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)
    
```

l) Model Evaluasi

```

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

m) Classification 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):

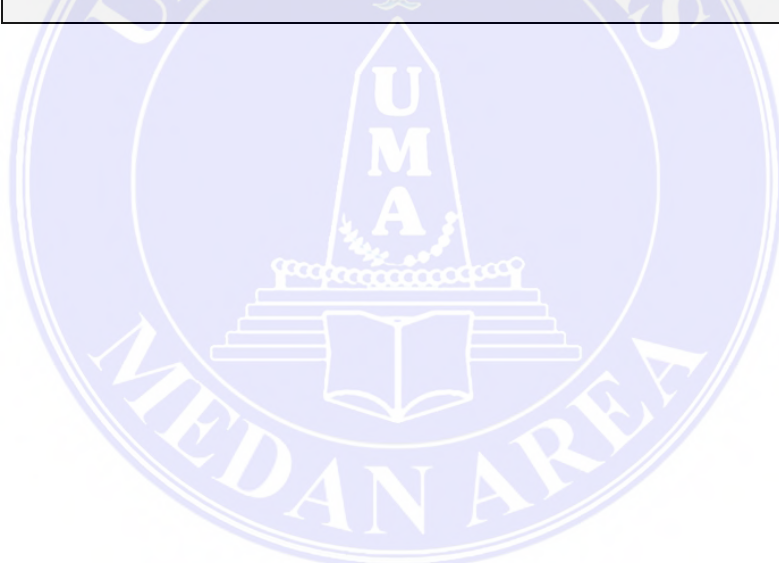
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    
```

```
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')
```





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 : 56/FT.6/01.10/II/2022
Lamp : -
Hal : **Perubahan Judul Tugas Akhir & Perpanjangan SK Pembimbing Tugas Akhir**

7 Februari 2022

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

Dengan hormat,
Sehubungan dengan adanya perubahan judul tugas akhir dan telah berakhirnya waktu masa berlaku SK pembimbing nomor 67/FT.6/01.10/VII/2021 pada tanggal 01 Juli 2021 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa tersebut :

N a m a : Dian Ifantiska
N P M : 178160083
Jurusan : Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

1. **Muhathir, ST, M.Kom** (Sebagai Pembimbing I)
2. **Rizki Muliono, S.Kom, M.Kom** (Sebagai Pembimbing II)

Adapun Tugas Akhir Skripsi berjudul :

“Implementasi Arsitektur *GoogleNet* dan *Xception* untuk Identifikasi Penyakit pada Daun Tanaman 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. Rahmad Syah, S. Kom, M. Kom



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 : 230/FT.6/01.10/XII/2021

6 Desember 2021

Lamp : -

Hal : **Penelitian Dan Pengambilan Data Tugas Akhir**

Yth. Pimpinan Pusat Penelitian Kelapa Sawit
Jl.Brigen Katamso No.51 Kampung Baru
Di
Medan

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	Dian Ifantiska	178160083	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 lengkap pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :

Implementasi Arsitektur *GoogleNet* dan *Xception* untuk Identifikasi Penyakit pada Daun Tanaman Kelapa Sawit

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Dekan,

Dr. Rahmad Syah, S. Kom, M. Kom

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



PT Riset Perkebunan Nusantara
PUSAT PENELITIAN KELAPA SAWIT
Implementasi Cui Ralahi Bussawardi Inceptable



SURAT KETERANGAN
Nomor : Int-59/PPKS/1.0/III/2022

Yang bertanda tangan dibawah ini Ka. Bagian Penelitian, Pusat Penelitian Kelapa Sawit (PPKS), dengan ini menyatakan yang namanya tersebut di bawah ini :


Nama : DIAN IFANTISKA
NPM : 17.816.0083
Fakultas : TEKNIK
Prodi : TEKNIK INFORMATIKA
Perguruan Tinggi : UNIVERSITAS MEDAN AREA

Dinyatakan benar telah selesai melakukan penelitian di Pusat Penelitian Kelapa Sawit (PPKS), sehubungan dengan penyusunan skripsi "**IMPLEMENTASI ARSITEKTUR GOOGLNET DAN XCEPTION UNTUK IDENTIFIKASI PENYAKIT PADA DAUN TANAMAN KELAPA SAWIT**".

Demikian surat keterangan ini diperbuat dengan sebenarnya untuk dapat dipergunakan seperlunya.

Medan, 16 Maret 2022

Hormat kami,


Dr. Sri Wening
Plt. Ka. Bag. Penelitian

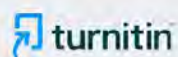
cc. Arsip

NS/RAA/map

Semua korespondensi mohon di hujukan kepada Kepala Pusat Penelitian melalui alamat atau email dibawah.
Please address all correspondence to the Head of Research Institute at the address or the e-mail below.



Medan
Jln. Brigjen Katamso No. 51, Kp. Baru, Medan 20158 - Sumatera Utara
+62 61 7862477 admin@iopri.org
+62 61 7862488 www.iopri.org



Similarity Report ID: oid:29477:21532237

PAPER NAME
SIDANG_DIAN IFANTISKA_FIX.pdf

AUTHOR
Dian Ifantiska

WORD COUNT
9075 Words

CHARACTER COUNT
59001 Characters

PAGE COUNT
60 Pages

FILE SIZE
3.0MB

SUBMISSION DATE
Aug 31, 2022 3:26 AM GMT+7

REPORT DATE
Aug 31, 2022 3:27 AM GMT+7

● **25% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 23% Internet database
- 8% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

● **Excluded from Similarity Report**

- Small Matches (Less than 10 words)

Summary