

**ANALISIS BOOSTING ALGORITHM DAN HOG DALAM
KLASIFIKASI PADA PENYAKIT DAUN MANGGA**

SKRIPSI

**OLEH :
RIZKI SUPRAYOGO
178160118**



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

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

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

Document Accepted 2/10/23

Access From (repository.uma.ac.id)2/10/23

LEMBAR PENGESAHAN

Judul Skripsi : Analisis Boosting Algorithm Dan HOG Dalam Klasifikasi Pada

Penyakit Daun Mangga


Nama : Rizki Suprayogo


NPM : 178160118


Fakultas : Teknik


Prodi : Informatika

Disetujui oleh
Komisi Pembimbing


Muhathir, ST, M.Kom
Pembimbing I


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


Dr. Rahmad Syah, S.Kom, M.Kom
Dekan Fakultas Teknik


Rizki Mulliono, S.Kom, M.Kom
Ka. Prodi

Tanggal Lulus : 19 Januari 2023

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah. Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.



Medan, 29 April 2023



Rizki Supravogo
178160118

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGASAKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

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

Nama : Rizki Suprayogo
NPM : 178160118
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 :

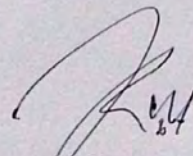
Analisis Boosting Algorithm Dan HOG Dalam Klasifikasi Pada Penyakit Daun Mangga

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 : 29 April 2023

Yang menyatakan



(Rizki Suprayogo)

RIWAYAT HIDUP

Penulis dilahirkan di Torganda Pada tanggal 04 Desember 1998 dari Bapak Sawali H S dan Ibu Nurhayani Sinaga. Penulis merupakan anak ke-tiga (3) dari lima (5) bersaudara. Tahun 2017 Penulis lulus dari SMK Swasta Pemda Rantauprapat dan pada tahun 2017 terdaftar sebagai mahasiswa Fakultas Teknik Universitas Medan Area. Selama mengikuti perkuliahan, penulis melaksanakan kerja praktek (KP) di Satuan Pendidikan Non Formal Sanggar Kegiatan Belajar Kota Binjai.



ABSTRAK

Boosting algorithm merupakan metode pembelajaran ensemble yang menggabungkan *learner* yang lemah menjadi *learner* yang kuat untuk meminimalkan kesalahan. Setiap metode mencoba mengkompensasi kelemahan dengan setiap iterasi, aturan lemah dari masing-masing *classifier* digabungkan untuk membentuk satu aturan akurasi yang kuat. *Boosting algorithm* memiliki 4 jenis metode yaitu *Adaptive Boosting* (Adaboost), Gradient Boosting, *Extreme Gradient Boosting* (XGBoost), dan *Light Gradient Boosting Machine* (LightGBM). Dikarenakan *boosting algorithm* memiliki keunikan dari masing-masing jenis metode. Maka penelitian ini akan di uji coba pada kasus penyakit daun mangga yang terdiri dari penyakit daun mangga *Capmodium* dan penyakit daun mangga *Collectricum* dengan memanfaatkan bantuan ekstraksi fitur *Histogram of Oriented Gradient* (HOG). Hasil akurasi tertinggi pada klasifikasi penyakit daun mangga dengan memanfaatkan ekstraksi fitur algoritma HOG pada XGBoost mencapai 95%, sedangkan Adaboost mencapai 85%, kemudian Gradient Boosting mencapai 89% , selanjutnya LightGBM mencapai 91%.

KATA KUNCI : Penyakit Daun Mangga, Boosting Algorithm, HOG.

ABSTRACT

Boosting algorithm is an ensemble learning method that combines weak learners to become strong learners to minimize errors. Each method tries to compensate for weaknesses with each iteration, the weak rules of each classifier combine to form one strong accuracy rule. There are 4 types of boosting algorithms, namely Adaptive Boosting (Adaboost), Gradient Boosting, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM). Because the boosting algorithm has the uniqueness of each type of method. So this research will be tested on cases of mango leaf disease consisting of Capmodium mango leaf disease and Collectricum mango leaf disease by utilizing the help of Histogram of Oriented Gradient (HOG) feature extraction. The highest accuracy results in the classification of mango leaf disease by utilizing the HOG feature extraction algorithm on XGBoost reach 95%, while Adaboost reaches 85%, then Gradient Boosting reaches 89%, then LightGBM reaches 91%.

Keywords : Mango Leaf Disease, Boosting Algorithm, HOG

KATA PENGANTAR

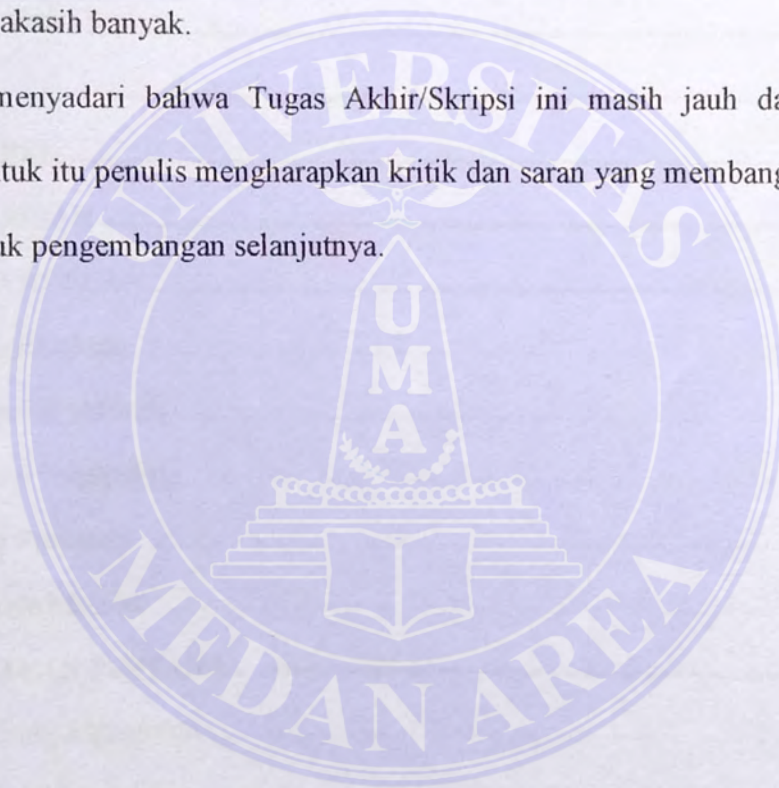
Puji dan syukur kehadirat Allah SWT Yang Maha Esa atas, rahmat, dan hidayah-Nya, sehingga saya bisa menyelesaikan penyusunan skripsi dengan judul “Analisis Boosting Algorithm Dan HOG Dalam Klasifikasi Pada Penyakit Daun Mangga”.

Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan banyak dukungan seperti dukungan motivasi, dukungan informasi materi, dan arahan, sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya:

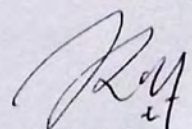
1. Allah SWT Yang Maha Esa, berkat rahmat dan hidayah-Nya skripsi ini dapat terselesaikan,
2. Orang Tua Bapak dan Ibu Saya yang telah mendukung, memberikan semangat, motivasi, dan banyak perhatian serta memenuhi segala kebutuhan saya selama masa penyusunan tugas akhir/skripsi ini,
3. Bapak Prof. Dr. 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 Dan Pembimbing II yang telah memberikan banyak masukan, kritik, saran, dan motivasi kepada penulis serta membimbing penulis dalam menyelesaikan tugas akhir/skripsi ini,
5. Ibu Susilawati, S.Kom., M.Kom., selaku Wakil Dekan Fakultas Teknik Universitas Medan Area yang telah memberikan, kritik dan saran kepada penulis dalam menyelesaikan tugas akhir/skripsi ini,

6. Bapak Rizki Muliono, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Medan Area,
 7. Bapak Muhathir ST, M.Kom., selaku Dosen Pembimbing I yang telah memberikan arahan, bimbingan, semangat, dan motivasi kepada penulis hingga penyusunan tugas akhir/skripsi ini terselesaikan,
 8. Serta semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir/skripsi ini, yang Namanya tidak bisa disebutkan satu persatu.
- Terimakasih banyak.

Penulis menyadari bahwa Tugas Akhir/Skripsi ini masih jauh dari kata sempurna, untuk itu penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk pengembangan selanjutnya.



Medan, 29 April 2023


Rizki Suprayogo
178160118

DAFTAR ISI

LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	v
RIWAYAT HIDIP	vi
ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR.....	ix
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 Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Boosting Algoritm	7
2.1.1 Adaboost	7
2.1.2 Gradient Boosting	9
2.1.3 XGBoost	10
2.1.4 LightGBM.....	11
2.2 Penyakit Tanaman Mangga Pada Daun	12
2.2.1 Capmodium Mangiferum.....	12
2.2.2 Collectroticum Ghoesporioides.....	12
2.3 Histogram of Oriented Gradients (HOG).....	12

2.4	Python	15
2.5	Confusion Matrix	15
2.6	Penelitian-Penelitian Terdahulu.....	16
BAB III METODE PENELITIAN		22
3.1	Studi Literatur	22
3.2	Spesifikasi Sistem	22
3.3	Teknik Pengumpulan Data.....	23
3.4	Pembagian Data	25
3.5	Metode Evaluasi.....	26
3.6	Arsitektur Penelitian	27
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		29
4.1	Hasil	29
4.1.1	Ekstraksi Fitur HOG	29
4.1.2	Klasifikasi Variasi Boosting	31
4.1.3	Confusion Matrix.....	34
4.1.4	Evaluasi Kinerja Algoritma.....	36
4.2	Pembahasan.....	39
BAB V KESIMPULAN DAN SARAN		43
5.1	Kesimpulan	43
5.2	Saran	43
DAFTAR PUSTAKA.....		44
LAMPIRAN-LAMPIRAN		47

DAFTAR TABEL

Tabel 2.1 Confusion Matrix	16
Tabel 2.2 <i>State of The Art</i>	16
Tabel 3.1 Perangkat Keras (<i>Hardware</i>).....	22
Tabel 3.2 Perangkat Lunak (<i>Software</i>)	23
Tabel 3.3 Pembagian Data Daun Mangga	25
Tabel 3.4 Pembagian Data Testing & Training	26
Tabel 4.1 Evaluasi Adaboost.....	36
Tabel 4.2 Evaluasi Gradient Boost.....	37
Tabel 4.3 Evaluasi XGBoost.....	37
Tabel 4.4 Evaluasi LightGBM	38
Tabel 4.5 Pembahasan Pada Penelitian	40

DAFTAR GAMBAR

Gambar 2.1 Diagram Alur Dari Algoritma HOG.....	13
Gambar 3.1 Daun <i>Collectricum</i>.....	24
Gambar 3.2 Daun <i>Capmodium</i>.....	24
Gambar 3.3 Daun Normal.....	25
Gambar 3.4 Alur Metodologi Penelitian.....	27
Gambar 4.1 Perubahan Daun <i>Capmodium</i> ke HOG	29
Gambar 4.2 Perubahan Daun <i>Collectricum</i> ke HOG.....	30
Gambar 4.3 Perubahan Daun Normal ke HOG.....	30
Gambar 4.4 Learning Curve Variasi Boosting Algorithm.....	31
Gambar 4.5 ROC variasi Boosting Algorithm	32
Gambar 4.6 Grafik Accuracy Varian Algorithm Boosting.....	33
Gambar 4.7 Confusion Matrix Adaboost	34
Gambar 4.8 Confusion Matrix Gradient Boost.....	34
Gambar 4.9 Confusion Matrix XGBoost	35
Gambar 4.10 Confusion Matrix LightGBM.....	35

BAB I PENDAHULUAN

1.1 Latar Belakang

Boosting Algorithm merupakan algoritma iteratif yang memberikan bobot yang berbeda pada distribusi training data pada setiap iterasi. Setiap iterasi *boosting* menambahkan bobot pada contoh-contoh kesalahan klasifikasi dan menurunkan bobot pada contoh klasifikasi yang benar, sehingga secara efektif dapat merubah distribusi pada data training (Ahmad, dkk., 2019). *Boosting Algorithm* adalah salah satu metode *Ensemble Learning* yang menggunakan hanya satu tipe model *base-model* dengan melakukan pembelajaran secara sekuensial secara adaptif (hasil dari sebuah *base-model* tergantung dari hasil *base-model* sebelumnya), dan kemudian digabungkan untuk mendapatkan hasil yang terbaik.

Saat ini algoritma *boosting* telah dikembangkan mulai dari *Adaptive Boosting* (adaboost) dikemukakan oleh Yoav Freund & Robert Schapire pada tahun 1995 (Bauer & Kohavi, 1999). Kemudian dikembangkan oleh Friedman pada tahun 2001 menjadi *Gradient Boosting* (Natekin & Knoll, 2013). Selanjutnya dikembangkan lagi oleh Tianqi Chen pada tahun 2016 menjadi *Xtreme Gradient Boosting* (XGBoost) (Pinata, dkk, 2020). Dan saat ini perkembangan algoritma *boosting* telah mencapai menjadi *Light Gradient Boosting Machine* (LightGBM) yang dikembangkan oleh Goulin pada tahun 2016.

Adaptative boosting (AdaBoost) merupakan algoritma pembelajaran terintegrasi yang bertujuan untuk memperoleh dan menyesuaikan bobot semua sampel pelatihan melalui pelatihan model berulang. Setiap iterasi membentuk prediktor lemah dan kemudian menggabungkan prediktor lemah ini menjadi prediktor kuat melalui pembobotan, sehingga meningkatkan kinerja model secara keseluruhan. Pada penelitian sebelumnya tentang Adaboost yaitu *Short-Term Wind Power Prediction Based On Particle Swarm Optimization-Extreme Learning Machine Model Combined With Adaboost Algorithm* bahwa model prediksi tenaga angin adaboost berkinerja lebih baik dengan tingkat akurasi 97% (Guoqing,dkk, 2021).

Gradien boosting adalah strategi pembelajaran *ensemble* yang menyatukan beberapa *learners* yang lemah untuk membangun model yang kuat oleh karena itu, akurasi prediksinya secara signifikan lebih baik daripada model tunggal. Pada penelitian sebelumnya Algoritma telah diterapkan pada prediksi *genom* (genetik) hewan dan tumbuhan yang berjudul *Predicting Phenotypes From High-Dimensional Genomes Using Gradient Boosting Decision Trees* hasil kemampuan akurasi prediksi lebih rendah daripada kinerja prediksi keseluruhan tetapi efisiensi komputasi keseluruhan lebih tinggi (Tingxi, dkk, 2022).

XGBoost adalah model ensemble menggunakan teknik boost untuk memperbaiki model pengklasifikasi lemah secara berurutan dan menjadikannya sebagai pengklasifikasi kuat. XGBoost membuat *residual* yang dipelajari secara berurutan dengan cara *Residual Fitting* sehingga dapat meningkatkan akurasi klasifikasi data. Pada penelitian sebelumnya untuk

memprediksi resiko demensia yang berjudul *Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization* hasil evaluasi model XGBoost memiliki akurasi 85,61% dan skor F1 79,28% (Seong, dkk, 2020).

LighGBM adalah algoritma berbasis pohon keputusan, yang membagi parameter di lapisan input menjadi bagian yang berbeda dan dengan demikian membangun hubungan pemetaan antara input dan output. Pada penelitian sebelumnya tentang LighGBM yang berjudul *Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River* bahwa prediksi ketinggian air sungai *Columbia* mendapatkan keakuratan model LighGBM dievaluasi secara statistik menunjukkan akurasi yang lebih tinggi di bandingkan dengan model yang lainnya (Gan, dkk, 2021)

Algoritma *machine learning* pada keluaran hanya melakukan klasifikasi pada data numerik. Sehingga dibutuhkan ekstraksi fitur untuk dapat mengubah data dari gambar menjadi data numerik. Salah satu ekstraksi fitur adalah algoritma HOG. *Histogram gradien berorientasi* (HOG) adalah ekstraksi fitur yang digunakan untuk deteksi objek dalam visi komputer dan pemrosesan gambar. Yang dapat membangun fitur dengan menghitung histogram arah gradien dari area lokal gambar. Algoritma HOG beroperasi pada sel lokal gambar, sehingga mempertahankan invarian yang baik terhadap deformasi geometris dan optik gambar, dan dapat mengekstrak fitur tekstur gambar dengan lebih akurat (Wenyu , dkk, 2019).

Berdasarkan keunikan jenis *Boosting Algorithm* dan algoritma HOG sebagai *Feature Extraction* maka dengan penelitian ini akan di uji coba pada

kasus penyakit pada daun mangga. Pada penelitian ini akan membedakan jenis-jenis penyakitnya yang terdapat pada daun mangga yaitu (a) Penyakit *mangifera (capmodium mangiferum)*, Jamur *Diplodia* menyebabkan luka tanaman mangga muda yang dicangkok. Umumnya, daun mangga yang terkena mengakibatkan daun menjadi warna hitam beledu, warna hitam disebabkan oleh jamur yang hidup dalam cairan manis. (b) *Colletotrichum Gloeosporiodes* bercak karat merah disebabkan oleh jamur *colletotrichum gloeosporiodes*, biasanya menyerang daun, ranting, bunga dan pucuk mangga sehingga membentuk bercak berwarna merah (Solikin, 2020).

Menurut (Suhendri, dkk, 2017) menyampaikan dalam karyanya klasifikasi jenis daun mangga menggunakan metode *Support Vectot Machine* (SVM). Dari hasil temuannya dengan tingkat akurasi 64,67%.

Menurut (Sarngadi, dkk, 2014) menyampaikan dalam karyanya klasifikasi kualitas kesehatan daun mangga berdasarkan warna citra daun mangga. Hasil pengamatan mereka menghasilkan klasifikasi dengan beberapa cluster yang mepresentasikan tingkat kesehatan daun, serta memiliki vektor ciri dari masing-masing cluster.

Dari penelitian terdahulu belum ditemukan algoritma boosting untuk mengklasifikasi jenis daun mangga yang terkena penyakit. Maka dari itu penelitian ini memanfaatkan algoritma boosting sebagai metode klasifikasi dan algoritma HOG sebagai ekstraksi fitur untuk mengklasifikasi jenis daun mangga berpenyakit.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diambil adalah Bagaimana kinerja variasi algoritma boosting dalam mengklasifikasi jenis penyakit daun mangga dengan bantuan ekstraksi fitur algoritma HOG ?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah Untuk mengetahui tingkat akurasi pada variasi *Boosting Algorithm* dengan bantuan ekstraksi fitur algoritma HOG dalam mengklasifikasi penyakit pada daun mangga.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat memberi sumbangan ide atau pengetahuan dalam menganalisis Boostings Alghorithm dan algoritma HOG dalam mengklasifikasi penyakit pada daun mangga.
2. Mendapatkan wawasan dan pengalaman mengenai jenis penyakit yang terdapat pada daun mangga.
3. Sebagai salah satu bahan dalam penulisan skripsi, guna memenuhi persyaratan untuk mendapatkan gelar sarjana di Program Studi Informstika Fakultas Teknik Universitas Medan Area.

1.5 Batasan Masalah

Adapun yang menjadi batasan masalah adalah sebagai berikut :

- a. Daun mangga yang terkena penyakit yaitu daun capmodium, daun collectricum dan daun normal.
- b. Jumlah daun yang digunakan sebanyak 1500 data sampel diantaranya 500 daun capmodium, 500 daun collectricum, dan 500 daun normal.
- c. Data sample diambil dengan cara memetik daunnya kemudian difoto menggunakan kamera smarphone Realme 7 pro dengan format .jpg dan backgroundnya putih menggunakan kertas HVS.
- d. Metode yang digunakan Algorithm Boosting diantaranya Adaboost, Gradient Boost, XGBoost dan LightGBM.
- e. Bantuan Ekstraksi Fitur yang digunakan adalah algoritma HOG.

BAB II TINJAUAN PUSTAKA

2.1 Boosting Algorithm

Boosting adalah teknik ensemble yang terkenal dan handal yang terbukti sering mengungguli algoritma pembelajaran lainnya. *Boosting Algorithm* adalah pendekatan machine learning yang didasarkan pada ide untuk membuat prediktor yang sangat akurat dengan menggabungkan banyak “aturan praktis” yang lemah dan tidak akurat. Algoritma boosting juga dioptimalkan berdasarkan peningkatan learning yang dikembangkan (Lihi & Erez, 2020). Adapun jenis-jenis yang akan dilakukan untuk melakukan boosting adalah sebagai berikut:

2.1.1 Adaboost

Menurut (Jifang, dkk, 2021) AdaBoost algoritma membangun beberapa pengklasifikasi lemah melalui beberapa iterasi, dan mempengaruhi bobot sampel pengklasifikasi generasi berikutnya sesuai dengan hasil klasifikasi, dan melakukan penambangan dalam sampel dengan menetapkan bobot yang berbeda untuk sampel, dan akhirnya, bobot voting untuk menghasilkan pengklasifikasi yang kuat. AdaBoost melatih pengklasifikasi lemah pertama dengan menetapkan bobot awal ke sampel pelatihan. Menurut hasil klasifikasi sampel pengklasifikasi lemah setelah pelatihan, bobot sampel diperbarui secara dinamis untuk membuat sampel yang salah diklasifikasikan menerima lebih banyak, dan kemudian berdasarkan pengklasifikasi lemah keseluruhan kesalahan uji digunakan untuk menyesuaikan bobot pengklasifikasi lemah, jadi untuk mengubah proses pelatihan pengklasifikasi terakhir, dan melatih semua pengklasifikasi satu per satu, dan mendapatkan pengklasifikasi kuat sesuai dengan bobot akhir pengklasifikasi lemah.

Berikut terdapat langkah-langkah pada algoritma adaboost yaitu :

Input : Suatu kumpulan sample penelitian dengan label $\{(x_i, y_i), \dots, (x_N, y_N)\}$, suatu *Component Learn* algoritma jumlah perputaran T.

1. *Initialize* : Bobot suatu sample pelatihan $w_i^1 = \frac{1}{N}$, untuk semua

$i = 1, \dots, N$.

2. For $t = 1, \dots, T$.

a. Gunakan *componet learn* algoritma untuk melatih suatu komponen klasifikasi h_t pada sample bobot pelatihan.

b. Hitung kesalahan pelatihannya pada

$$h_t: \varepsilon_t = \sum_{i=1}^N w_i^t y_i \neq h_t(x_i)$$

c. Tetapkan bobot untuk *component classifier*.

$$h_t = a_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

d. Update bobot sample pelatihan

$$w_i^{t+1} = \frac{w_i^t \exp\{-a_t y_i h_t(x_i)\}}{c_t}, i = 1, \dots, N$$

adalah salah satu konstanta normalisasi.

e. *Output* : $f(x) = \text{sign}(\sum_{t=1}^T a_t h_t(x))$

2.1.2 Gradient Boosting

Menurut(Fuki, dkk, 2022) Gradient boosting (GB) adalah model machine learning yang populer, algoritma GB terdiri dari beberapa pohon keputusan, yang didasarkan pada pembelajaran dasar dihasilkan secara berurutan sedemikian rupa sehingga pembelajar dasar saat ini lebih efektif daripada yang sebelumnya menjadi model keseluruhan meningkatkan secara berurutan dengan setiap iterasi. Bobot untuk hasil yang salah di klasifikasikan tidak bertambah, sebagai gantinya metode peningkatan gradien mencoba mengoptimalkan fungsi kerugian dari learning sebelumnya dengan menambahkan yang lemah untuk mengurangi fungsi kerugian.

Berikut langkah-langkah pada algoritma gradient boost adalah

- Input data $(x, y)_{i=1}^N$
- Number of iterations M
- Pilihan loss function (y, f)
- Pilihan base learner model $h(x, \theta)$

Algoritma:

1. Intialize f_0 dengan constant
2. *for* $t = 1$ to M *do*
3. Menghitung negatif gradient $g_t(x)$
4. Mencocokkan new base learner function $h(x, \theta_t)$
5. Menemukan ukuran langkah penurunan gradient terbaik

$$\rho_t = \arg \min_{\rho} \sum_{i=1}^N [y_i, f_{t-1}(x_i) + \rho h(x_i, \theta_t)]$$

6. Update function estimate
7. $f_t \leftarrow f_{t-1} + \rho_t h(x, \theta_t)$

2.1.3 XGBoost

XGBoost adalah versi lanjutan dari metode Gradient Boosting secara harfiah berarti eXtreme Gradient Boosting. Tujuan utama algoritma ini adalah untuk meningkatkan kecepatan dan efisiensi komputasi. Algoritma Gradient Boosting menghitung output pada tingkat yang lebih lambat karena mereka mengalisis kumpulan secara berurutan, oleh karena itu XGBoost digunakan untuk meningkatkan kinerja model. Metode XGBoost sering digunakan banyak orang pada kompetisi *machine learning* yang diselenggarakan oleh Kaggle pada tahun 2015. XGBoost merupakan *tree ensembles algorithm* yang terdiri atas beberapa *classification and regression tree* (CART). Algoritma XGBoost melakukan optimasi 10 kali lebih cepat dibandingkan dengan implmentasi dari Gradient boosting (Iswaya, dkk, 2019).

Menurut (Neng, dkk, 2021) langkah-langkah pada algoritma XGBoost adalah sebagai berikut :

- Data : Dataset and hyperparameters
- Initialize $f_0(x)$;
- For $k = 1, 2, \dots, M$ do
 1. Calculate $g_k = \frac{\partial L(y, f)}{\partial f}$;
 2. Calculate $h_k = \frac{\partial^2 L(y, f)}{\partial f^2}$;
 3. Menentukan struktur dengan memilih perpecahan dengan gain maksimail
$$A = \frac{1}{2} \left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G^2}{H} \right];$$
 4. Menentukan leaf weights $w^* = -\frac{G}{H}$;

5. Menentukan base learner $b(x) = \sum_{j=1}^T wI$;

6. Add trees $f_k(x) = f_{k-1}(x) + b(x)$;

End

Result: $f_k(x) = \sum_{k=0}^M f_k(x)$

2.1.4 LightGBM

LightGBM adalah penguat gradien ringan yang untuk memecahkan masalah yang memakan waktu di lingkungan sampel data dimensi tinggi dan yang besar (Zhang & Gong, 2020). LightGBM merupakan algoritma pembelajaran pohon keputusan penambah gradien yang telah banyak digunakan untuk fitur seleksi, klasifikasi dan regresi (Li, dkk, 2021)

Model LightGBM berkaitan dengan konsep Gradient Boosting, yang menggunakan orde kedua ekspansi untuk memperkirakan fungsi target. Lebih khusus lagi, fungsi target ke- i adalah diestimasi dengan menggunakan hasil belajar dari proses $(i - 1)$:

$$Obj_i = L(\eta_i - 1) + gT_i(X, \theta_i) + \frac{1}{2} hT_i^2(X, \theta_i) + \Omega(T_i) \quad (2.1)$$

$$g = \frac{\partial L(\eta_i - 1)}{\partial \eta_i - 1} \quad (2.2)$$

$$h = \frac{\partial^2 L(\eta_i - 1)}{\partial \eta_i - 1^2} \quad (2.3)$$

di mana g dan h adalah turunan pertama dan kedua dari $L(\eta_i - 1)$ terhadap $(\eta_i - 1)$ karena $L(\eta_i - 1)$ diketahui dari proses pembelajaran $(i - 1)$, fungsi target dapat ditulis sebagai:

$$Obj_i = \sum_{j=1}^J \left[(G_j \omega_j + \alpha |\omega_j| + \frac{1}{2} (H_j + \beta) \omega_j^2) \right] \quad (2.3)$$

$$G_j = \sum_{t=1}^N g^t \quad (t \in \text{leaf node } j) \quad (2.4)$$

$$H_j = \sum_{t=1}^N h^t \quad (t \in \text{leaf node } j) \quad (2.5)$$

2.2 Penyakit Tanaman Mangga Pada Daun

Daun berpenyakit memiliki bintik-bintik tidak beraturan berukuran kurang dari 5mm, sehingga bintik-bintik membentuk kombinasi menjadi daun berlubang. Pusat-pusat yang sering diamati rusak, menghasilkan bintik-bintik berlubang. Daun yang terkena penyakit akan mengering dan musim gugur serangan daun dapat menyebabkan daun layu dan rontok. Menurut (Solikin, 2020) Adapun 2 jenis penyakit mangga yang terdapat pada daun adalah sebagai berikut :

2.2.1 *Capmodium Mangiferum*

Pada penyakit daun ini terdapat adanya jamur jelaga daun yang diserang berwarna hitam seperti beledu. Warna hitam ini disebabkan oleh jamur yang hidupnya dicairan manis. Pengendalian dengan memberantas serangga yang menghasilkan cairan manis dengan *insektisida* atau tepung belerang.

2.2.2 *Collectotrichum Gloeosporioides*

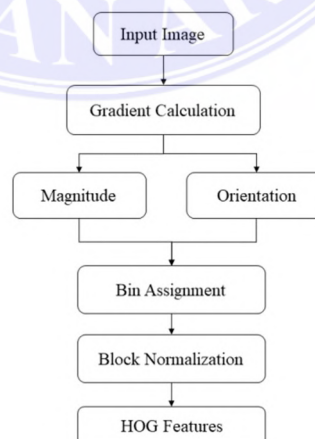
Pada penyakit daun ini terdapat adanya bercak karat merah yang menyerang daun sehingga berbentuk seperti bercak yang berwarna merah. Penyakit ini sangat mempengaruhi proses pertumbuhan. Pengendalian pemngkasan pada daun dan menyemprotkan fungisida bubuk *bordeaux* atau sulfat.

2.3 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) adalah salah satu teknik ekstraksi ciri yang dalam pemrosesan gambar yang mengklasifikasikan nilai gradien suatu piksel menurut orientasi di setiap bagian lokal dari gambar. Tampilan dan bentuk objek lokal seringkali bisa cukup baik berdasarkan distribusi gradien intensitas

lokal atau orientasi tepi, meskipun lokasi pasti dari gradien atau tepi yang sesuai tidak diketahui. (Shaojun, dkk, 2021). Hal inilah yang menjadi ide dasar dari teknik ekstraksi fitur yang diajukan oleh Dalal dan Triggs (2005). Distribusi gradien atau histogram inilah yang menjadi suatu fitur untuk pencocokan kesamaan sehingga dapat dilatihkan ke dalam suatu pembelajaran mesin (Sina, dkk., 2020).

Algoritma HOG merupakan teknik pengambilan karakteristik mencoba untuk mengekstrak informasi penting dari gambar. Bagaimana metode ini bekerja yaitu dengan mengevaluasi histogram lokal yang dinormalisasi lebih baik daripada mendistribusikan gradien gambar dalam grid yang padat. teknik ekstraksi fitur metode ini adalah distribusi lokal dari intensitas gradien setiap piksel terkandung dalam objek gambar. Dalam metode algoritma HOG pada pengenalan wajah manusia, ukuran sel berupa kumpulan atau kombinasi pixel dan blok berupa kumpulan atau kombinasi sel beserta jumlah orientation bin yang merupakan tempat untuk menampung hasil arah dan besar gradien akan mempengaruhi output fitur vektor yang dihasilkan dan juga akurasi yang dihasilkan (Sina Ghaffari, dkk., 2020).



Gambar 2.1 Diagram alur dari algoritma HOG

Pertama, gradien sumbu dihitung untuk setiap piksel pada gambar x dan y menggunakan Persamaan 1.

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y) \quad (2.6)$$

$$G_y(x, y) = I(x, y + 1) - I(x, y - 1) \quad (2.7)$$

Rumus (2.6) dan (2.7) menjelaskan bahwa gradien dilestarikan kurangi nilai piksel sebelumnya dari nilai piksel yang sesuai di belakangnya dengan sumbunya. Setelah mendapatkan nilai gradien sumbu x dan y masing-masing piksel, proses selanjutnya adalah menghitung nilai gradien dan arah dengan Persamaan.

$$M(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (2.8)$$

$$\theta(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)} \quad (2.9)$$

Nilai besaran gradien dan arah sudut gradien digunakan sebagai komponen pada proses selanjutnya. G_x dan G_y masing-masing adalah gradien sumbu x dan y. Kemudian setiap piksel pada gambar dibagi menjadi beberapa sel, setiap sel menghitung distribusi histogram gradien berorientasinya melalui proses pencocokan. Proses vote histogram gradien berorientasi pertama-tama menentukan nilai bin dengan membagi jumlah total sudut gradien dengan jumlah bin orientasi. Kemudian, untuk setiap arah, pertama-tama sudut gradien dari setiap piksel dalam sel dimasukkan ke dalam bin orientasi yang ditentukan dan kemudian nilai gradien dibagi dengan arah bin terkait.

2.4 Python

Python telah menjadi bahasa pemrograman pilihan untuk area penelitian yang selama beberapa tahun terakhir, tidak hanya karena mudah digunakan tetapi juga dukungan komunitas yang baik. Python adalah bahasa pemrograman tingkat tinggi, lintas platform, dan ditafsirkan yang berfokus pada keterbacaan kode. Sejumlah besar perpustakaan berkualitas tinggi tersedia dan dukungan untuk segala jenis komputasi ilmiah dipastikan. Karakteristik ini menjadikan Python alat yang tepat untuk banyak proyek penelitian. Pengoptimalan peran penting dalam banyak bidang ilmiah, seperti teknik, analisis data, dan pembelajaran mendalam. Bidang-bidang ini berkembang pesat dan konsepnya digunakan untuk berbagai tujuan, misalnya memperoleh wawasan dari kumpulan data besar atau menyesuaikan model prediksi yang akurat. Setiap kali suatu algoritma harus menangani sejumlah besar data, implementasi yang efisien dalam bahasa pemrograman yang sesuai adalah menggunakan python (Julian, dkk., 2020).

2.5 Confusion Matrix

Untuk memahami metrik yang digunakan, metrik tersebut dijelaskan terlebih dahulu bahwa TP (True Positive), TN (True Negative), FP (False Positive) dan FN (False Negatif) menurut Tabel 3.3. TP singkatan dari karena data positif diprediksi positif, TN didefinisikan sebagai data negatif diprediksi sebagai negatif. Meskipun FN sebaliknya TP adalah data positif yang diprediksi negatif dan FP, kebalikan dari TN, yaitu data negatif diprediksi sebagai positif.

Tabel 2.1 Confusion Matrix

		Kelas sebenarnya	
		Positif	Fasle Positive
Prediksi	Positif	True Positive	False Positive
	Negatif	False Negative	True Negative

2.6 Penelitian – penelitian terdahulu

Penelitian ini didasari dari sebuah penelitian terdahulu, baik dari jenis penelitian maupun teori yang digunakan pada penjelasannya dibawah ini sebagai berikut :

Tabel 2.2 *State of The Art*

No	Judul, Penulis, Tahun	Hasil Review
1.	Short-Term Wind Power Prediction Based On Particle Swarm Optimization-Extreme Learning Machine Model Combined With Adaboost Algorithm, Guoqing An, Ziyao Jiang , Xin Cao , Yufei Liang ,Yuyang Zhao ,	Pada kumpulan data sampel besar yang berisi data multi-musim, model prediksi tenaga angin Adaboost PSO-ELM dapat mengikuti perubahan kecepatan angin dan arah di setiap musim dan membuat prediksi, yang lebih unggul dari model tradisional lainnya. Performa prediksi model ini membuktikan bahwa model ini memiliki ketahanan dan kemampuan generalisasi yang baik serta dapat memberikan dasar yang lebih handal untuk pengiriman jaringan listrik.

	Zheng Li, (Member, Ieee), Weichao Dong, And Hexu Sun ^{1,2} , (Senior Member, Ieee), 2021.	
2.	Transformer Fault Diagnosis Based on Multi-Class AdaBoost Algorithm, Jifang Li , Genxu Li, Chen Hai, And Mengbo Guo, 2021.	Model diagnosis kesalahan IPSO-SVM yang disempurnakan oleh algoritma AdaBoost dapat secara efektif dan akurat mengidentifikasi jenis kesalahan transformator. Dibandingkan dengan algoritma SVM dan AdaBoost tradisional, ini menunjukkan akurasi klasifikasi yang lebih tinggi.
3.	Predicting Phenotypes From High-Dimensional Genomes Using Gradient Boosting Decision Trees, Tingxi Yu, Li Wang, Wuping Zhang, Guofang Xing, Jiwan Han, Fuzhong Li, And Chunqing Cao, 2022.	Ditemukan bahwa kecuali untuk sifat GW, akurasi prediksi GB lebih rendah daripada kinerja prediksi keseluruhan dari lima model Bayesian, tetapi efisiensi komputasi keseluruhan GB lebih tinggi daripada model Bayesian.
4.	eFL-Boost: Efficient Federated Learning for	Dalam hal kinerja prediksi, peneliti mengevaluasi kinerja tiga skema untuk beberapa kumpulan data publik, dan

	<p>Gradient Boosting Decision Trees, Fuki Yamamoto, Seiichi Ozawa, (Senior Member, Ieee), And Lihua Wang, 2022.</p>	<p>pengaruh jumlah pemilik yang berpartisipasi dan jumlah data untuk setiap pemilik data pada akurasi diselidiki. Dalam hampir semua kasus, eFL-Boost mengungguli TFL dan F-GBDT-G dan memiliki kinerja yang sebanding dengan FederBoost.</p>
5.	<p>Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization, Seong-Eun Ryu, Dong-Hoon Shin, And Kyungyong Chung, 2020.</p>	<p>Dibandingkan dengan model klasifikasi lainnya, model yang diusulkan menunjukkan kinerja terbaik. Ini membuktikan bahwa XGBoost berdasarkan ekstraksi variabel turunan dapat menghasilkan kinerja yang efektif untuk prediksi risiko demensia memiliki akurasi 85,61%.</p>
6.	<p>Comparation Analysis Of Ensemble Technique With Boosting(Xgboost) And Bagging(Randomforest) For Classify Splice Junction Dna Sequence Category, Iswaya Maalik ,Wisnu Ananta Kusuma , and Sri Wahjuni, 2019.</p>	<p>Penelitian ini menunjukkan bahwa metode ensemble baik boosting maupun bagging mampu menangani klasifikasi dengan cara yang baik, ketika hyperparameter ditentukan dengan tepat. Tingkat akurasi xgboost secara keseluruhan lebih unggul. Namun, kelemahan xgboost adalah proses pelatihannya membutuhkan lebih banyak waktu untuk diselesaikan.</p>

7.	<p>A Data-Driven Method for Power System Transient Instability Mode Identification Based on Knowledge Discovery and XGBoost Algorithm, Neng Zhang , Huimin Qian, Yuchao He, Lirong Li, And Chaoyun Sun, 2021.</p>	<p>Hasil eksperimen mengungkapkan bahwa dibandingkan dengan pendekatan lain, pendekatan yang diusulkan berdasarkan KODAMA dan XGBoost dapat mencapai akurasi evaluasi tertinggi 95,53%.</p>
8.	<p>Identification of Extremely Similar Animal Fibers Based on Matched Filter and HOG-SVM, Wenyu Xing,Na Deng, Binjie Xin, Yiwen Liu, Yang Chen, And Zhengye Zhang, 2019</p>	<p>Dalam metode ini, gambar serat asli disaring dengan filter yang cocok untuk mendapatkan tekstur serat gambar biner tanpa latar belakang dan noise. Kemudian, algoritma HOG operator fitur digunakan untuk mengekstrak fitur tekstur dari gambar keluaran filter, dan SVM digunakan untuk melatih dan mengklasifikasikan set fitur dimensi tinggi. Metode ini dapat memperoleh gambar tekstur yang jelas dan tidak bersuara tanpa pra-pemrosesan gambar yang rumit.</p>
9.	<p>Analysis and Comparison of FPGA-Based</p>	<p>Dalam penelitian ini, kami meninjau metode yang digunakan untuk mengimplementasikan histogram algoritma</p>

	<p>Histogram of Oriented Gradients Implementations, Sina Ghaffari, Parastoo Soleimani, Kin Fun Li, (Senior Member, IEEE) And David W. Capson, (Senior Member, IEEE), 2020</p>	<p>gradien berorientasi pada FPGA dalam dekade terakhir (2010-2019). Beberapa teknik yang dilaporkan terkait dengan langkah-langkah individual dari algoritme, dan beberapa memengaruhi keseluruhan algoritme. Kami juga meninjau berbagai metode penyederhanaan dan implementasi perangkat keras dari fitur yang dimodifikasi yang didasarkan pada algoritma HOG asli. Setelah itu, kami membandingkan karya terbaru mengenai kecepatan, akurasi, dan pemanfaatan sumber daya dari desain. Diamati bahwa metode yang berfokus pada pengoptimalan jalur data desain memiliki efek paling besar pada kecepatan sirkuit.</p>
<p>10.</p>	<p>Deteksi Penyakit Pada Tanaman Mangga Dengan Citra Digital : Tinjauan Literatur Sistematis (SLR), Solikin, 2020</p>	<p>Untuk hasil penelitian (Research Objective) yang sesuai yang terkait dengan penelitian ini adalah : 1) Menentukan metode gambar digital yang dapat digunakan dengan optimal untuk membedakan karakteristik penyakit tanaman mangga.</p> <p>2) Metode klasifikasi yang bisa digunakan secara efektif untuk mengklasifikasikan jenis penyakit pada tanaman mangga.</p>

11	<p>Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River,</p> <p>Min Gan, Shunqi Pan, Yongping Chen, Chen Cheng, Haidong Pan and Xian Zhu</p>	<p>Bahwa prediksi ketinggian air sungai <i>Columbia</i> mendapatkan keakuratan model LightGBM dievaluasi secara statistik menunjukkan akurasi yang lebih tinggi</p>
12	<p>A Multi-Organ Fusion and LightGBM Based Radiomics Algorithm for High-Risk Esophageal Varices Prediction in Cirrhotic Patients,</p> <p>Lijuan Li, Yiken Lin, Dexin Yu, Zhuyun Liu, Yanjing Gao , And Jianping Qiao,</p>	<p>Hasil multi-organ dan radiomik berbasis LightGBM dan kerangka kerja pembelajaran mesin untuk membedakan EV risiko tinggi dari EV risiko rendah. Biomarker esofagus terbukti lebih unggul daripada biomarker esofagus hati dan limpa dalam prediksi risiko EV. LightGBM memiliki kinerja yang lebih tinggi daripada metode pemilihan fitur lainnya.</p>

BAB III

METODE PENELITIAN

3.1 Studi Literatur

Pada tahap studi literatur ini bertujuan untuk mempelajari teori - teori maupun metode yang akan digunakan dan berkaitan pada penelitian seperti metode algoritma HOG dan Boosting Alghoritm. Teori dan metode dikumpulkan dari beberapa jurnal.

3.2 Spesifikasi Sistem

Spesifikasi sistem merupakan dokumen yang berfungsi sebagai dasar bagi rekayasa perangkat keras, perangkat lunak, database dan manusia. Spesifikasi sistem menggambarkan fungsi dan kinerja dari sebuah sistem berbasis komputer serta batasan yang mengatur pengembangnya. Berikut merupakan perangkat keras dan perangkat lunak yang akan digunakan dalam penelitian :

Tabel 3.1 Perangkat Keras (Hardware)

No	Hardware	Spesifikasi
1	Laptop	Acer Aspire E 14
2	Processor	Intel inside Core i3-5005u (2.0 GHz, 3MB L3 Cache)
3	Ram	4gb DDR3
4	Kamera	Smartphone Realme 7 Pro

Tabel 3.2 Perangkat Lunak (Software)

No	Software	Spesifikasi
1	Os	Microsoft Windows 10 Enterprise Version: 10.0.17763 N/A Build 17763
2	Tools Pemograman	Jupyter Notebook (conda)
3	Bahasa Pemograman	Python
4	Kamera	Picsart Versi 18.4.0

3.3 Teknik Pengumpulan Data

Peneliti melakukan pengumpulan data dengan cara mengambil data sampel yang digunakan dalam bentuk gambar. Gambar yang diambil dalam penelitian ini adalah daun mangga. Data sampel daun mangga dalam penelitian ini berjumlah 1500 data yang terdiri dari 500 data penyakit daun *Capmodium Mangiferum*, 500 data penyakit daun *Collectotrichum Gloeosporioides*, dan 500 daun normal (sehat). Kamera yang digunakan untuk mengambil gambar sample daun mangga menggunakan smartphone realme 7 pro dengan format gambar jpg. Jarak pengambilan gambar per sampel berbeda-beda diantaranya 7cm, 8cm, 9cm, 10cm, 11cm,12cm, dan 13cm. background gambar berwarna putih menggunakan kertas HVS.



Gambar 3.1 Daun Collectricum

Pada gambar 3.1 data sampel memiliki karakteristik seperti bercak merah kemerahan di daerah tertentu yang terkena penyakit, sehingga daun menjadi rusak yang berdampak bolong-bolong pada daun mangga.



Gambar 3.2 Daun Capmodium

Pada gambar 3.2 data sampel memiliki karakteristik seperti bintik-bintik hitam di sekitar daun tertentu yang terkena penyakit, maka daun perlahan-lahan akan rusak.



Gambar 3.3 Daun Normal

Pada gambar 3.3 data sampel memiliki karakteristik daun yang sehat normal, dan tidak terkena penyakit apapun.

3.4 Pembagian Data

Data yang digunakan pada penelitian ini sebanyak 1500 data daun mangga yang terdiri dari 3 bagian yaitu daun capmodium, daun collectricum, dan daun normal (sehat). Data daun capmodium berjumlah 500 data, daun collectricum berjumlah 500 data, dan daun normal berjumlah 500 data.

Terdapat data training dan data test dengan masing-masing presentase sebanyak 80% dan 20% . Adapun rincian pembagian jumlah data citra dapat dilihat pada tabel 3.3.

Tabel 3.3 Pembagian Data Daun Mangga

Kelompok Data	Jumlah Data
Daun Capmodium	500
Daun Collectricum	500
Daun Normal	500
Total	1500

Tabel 3.4 Pembagian Data Testing & Training

Pembagian Data Keseluruhan untuk Test & Training	Jumlah Data
Training 80%	1200
Testing 20%	300

3.5 Metode Evaluasi

Metode evaluasi yang akan digunakan adalah *Accuracy*, *Precision*, *Recall*, dan *F1-score*. *Accuracy* adalah ukuran kinerja yang paling intuitif dan hanyalah rasio pengamatan yang diprediksi dengan benar terhadap total pengamatan. *Precision* adalah rasio pengamatan positif yang diprediksi dengan benar total pengamatan positif yang diprediksi. *Recall* adalah rasio pengamatan positif yang di prediksi dengan benar semua pengamatan di kelas yang sebenarnya. *F1-score* adalah rata-rata tertimbang dari *precision* dan *recall* maka skor ini memperhitungkan positif palsu dan negatif palsu.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (3.1)$$

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

$$Recall / Sensitivity = \frac{TP}{TP+FN} \quad (3.3)$$

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

$$Specificity = \frac{TN}{TN+FP} \quad (3.5)$$

Keterangan :

TP = *true positive*,

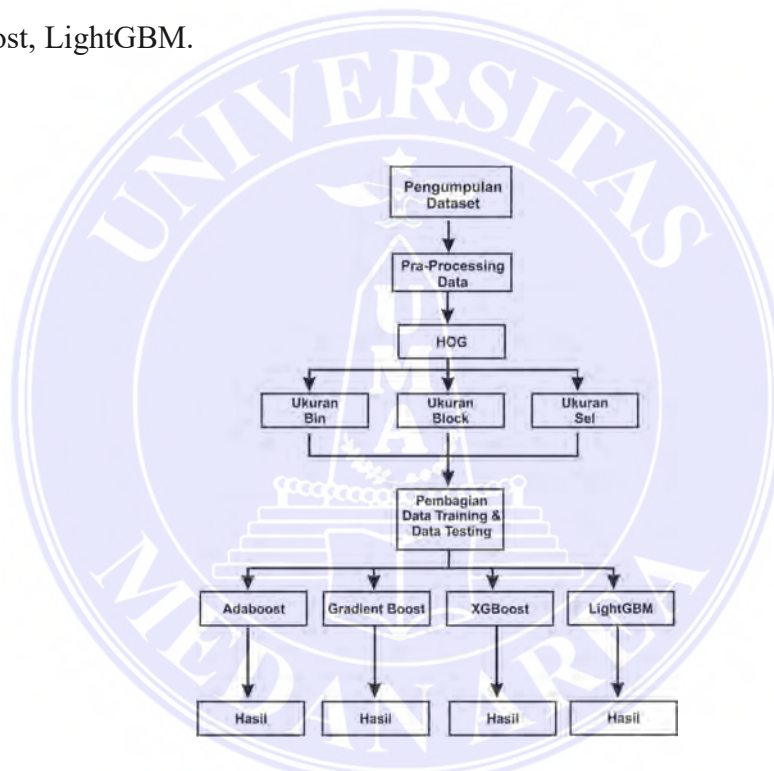
FP = *false positive*

FN = *false negative*,

TN = *true negative*

3.6 Arsitektur Penelitian

Perlu dilakukan langkah-langkah analisis data dengan menggunakan tiga tahapan, yaitu pre-processing, ekstraksi fitur menggunakan algoritma HOG dan klasifikasi Boosting Algoritm yang terdiri dari Adaboost, Gradient Boosting, XGBoost, LightGBM.



Gambar 3.4 Alur metodolgi penelitian

Dataset yang sudah dikumpulkan sebanyak 1500 sampel akan di pra-processing yaitu melakukan *resize* gambar yang bertujuan untuk mengubah dimensi gambar tersebut seperti ukuran lebar,tinggi, dan aspek ratio. Maka ukuran gambar yang akan digunakan berukuran (282 x 400) pixel dan (391, 293) pixel. Setiap gambar berukuran besar, pemrosesan akan membutuhkan jumlah memori besar sehingga banyak memakan waktu yang cukup lama pada saat mengubah visual

gambar menjadi HOG. Sehingga digunakan meresize gambar agar tidak banyak memakan waktu.

Kemudian mengubah visual gambar menjadi HOG sebagai fitur ekstraksi membuat ukuran bin, ukuran sel, dan ukuran blok. Dalam algoritma HOG gambar dibagi menjadi beberapa blok, dan setiap blok dibagi menjadi beberapa sel. Sebagai contoh, setiap blok dapat berisi empat sel, dan setiap sel dapat berisi 16 (4×4) piksel.

Langkah selanjutnya adalah proses klasifikasi dengan membuat kombinasi model tetapi memilih model dengan bobot tertinggi untuk klasifikasi. Setiap model yang dibuat memiliki atribut berupa nilai bobot. diklasifikasikan menggunakan varian boosting (Adaboost, Gradient Boost, XGBoost, dan LightGBM) untuk memberikan hasil klasifikasi.

Setelah hasil klasifikasi sudah keluar maka peneliti akan menganalisa varian boosting (Adaboost, Gradient boost, XGBoost, dan LightGBM) yang memiliki hasil terbaik dengan cara membandingkannya. Hasil nilai yang tertinggi akan menjadi acuan metode yang terbaik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis penelitian yang telah dilakukan, maka dapat disimpulkan bahwa klasifikasi dengan menggunakan metode *Boosting Algorithm* (adaboost, gradient boost, dan XGBoost) dan ekstraksi fitur Histogram of Oriented Gradient (HOG) untuk mengklasifikasi citra penyakit daun mangga didapatkan hasil XGBoost mempunyai nilai akurasi yang paling baik sebanyak 93% sedangkan adaboost sebanyak 87%, gradient boost sebanyak 89% dan Lightgbm sebanyak 87%.

5.2 Saran

Masih banyak kekurangannya yang ada pada penelitian ini, saran yang dapat penulis berikan demi pengembangan selanjutnya adalah :

- a. Ekstraksi fitur dapat dianalisis ulang dengan menggunakan ekstraksi fitur lainnya seperti Surf, Kaze, Brisk, dan lain-lain
- b. Pengembangan teknologi untuk mencapai hasil yang lebih maksimal dengan menyempurnakan seluruh jenis penyakit pada daun mangga.
- c. Pengujian dengan Machine Learning atau Deep Learning lainnya sebagai pengembang penelitian selanjutnya.

DAFTAR PUSTAKA

- Ahmad, R. A. , Muhammad, K. I., Dwi, K., Mohammad, R. F., Fatma, I. (2019). Teknik Bagging Dan Boosting Pada Algoritma CART Untuk Klasifikasi Masa Studi Mahasiswa. *Jurnal Sains dan Informatika*.
- Aziz, N., Akhir, E. A., Aziz, I. A., Jaafar, J., Hasan, M. H., & Abas, A. N. (2020). A Study on Gradient Boosting Algorithms for Development of AI Monitoring and Prediction System. *International Conference on Computing Intelligence (ICCI)*.
- Bauer, E., & Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms : Bagging, Boosting, and Variants. *Machine Learning*.
- Fuki, Y., Seiichi, O., & Lihua, W. (2022). eFL-Boost: Efficient Federated Learning for Gradient Boosting Decision Trees. *IEEE Access*.
- Gan, M., Pan, S., Chen, Y., Cheng, C., Pan, H., & Zhu, X. (2021). Application of the Machine Learning LighGBM Model to the Prediction on the Water Levels of the Lower Columbia River. *Marine Science and Engineering*.
- Guoqing, A., Ziyao, J., Xin, C., Yufei, L., Yuyang, Z., Zheng, L., Weichao, D., & Hexu, S. (2021). Short-Term Wind Power Prediction Based On Particle Swarm Optimization-Extreme Learning Machine Model Combined With Adaboost Algorithm. *IEEE Access*.
- Iswaya, M. S., Wisnu, A. K., & Sri, W. (2019). Comparison Analysis Of Ensemble Technique With Boosting(Xgboost) And Bagging(Randomforest) For Classify Splice Junction Dna Sequence Category. *Jurnal Penelitian Pos dan Informatika*.
- Jifang, L. , Genxu, L. , Chen, H.1 , & Mengbo, G. (2021). Transformer Fault Diagnosis Based on Multi-Class AdaBoost Algorithm. *IEEE Access*.

- Julian, B., & Kalyanmoy, D. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*.
- Linggar, M. C., & Adi, W. (2022). Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes. *Jurnal Masyarakat Informatika*.
- Lih, D., & Erez, S. (2020). BoostLR: A Boosting-Based Learning Ensemble for Label Ranking Tasks. *IEEE Access*.
- Li, L., Lin, Y., Yu, D., Liu, Z., Gao, Y., & Qiao, J. (2021). A Multi-Organ Fusion and LightGBM Based Radiomics Algorithm for High-Risk Esophageal Varices Prediction in Cirrhotic Patients. *IEEE Access*.
- Natekin, A., & Knoll, A. (2013). Gradient Boosting Machines, a Tutorial. *Frontiers in Neurorobotics*.
- Neng, Z., Huimin, Q., Yuchao, H., Lirong, L., & Chaoyun, S. (2021). A Data-Driven Method for Power System Transient Instability Mode Identification Based on Knowledge Discovery and XGBoost Algorithm. *IEEE Access*.
- Pinata, N. N., Sukarsa, I. M., & Rusjyanthi, N. D. (2020). Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python. *Journal Ilmiah Merpati*.
- Seong, E. R., Dong, H. S., & Kyungyong, C. (2020). Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization. *IEEE Access*
- Shaojun, G., Feng, L., Xiaohu, Y., Chunrong, Z., Li, C., & Tongsheng, S. (2021). HSPOG: An Optimized Target Recognition Method Based on Histogram of Spatial Pyramid Oriented Gradients. *Tsinghua Science And Technology*.
- Solikin. (2020). Deteksi Penyakit Pada Tanaman Mangga Dengan Citra Digital : Tinjauan Literatur Sistematis (SLR). *Bina Insani Ict Journal*.

- Sina, G. , Parastoo, S., Kin, F. L., & David, W. C. (2020). Analysis and Comparison of FPGA-Based Histogram of Oriented Gradients Implementations. *IEEE Access*.
- Suhendri, Muharam, F. M., & Aelani, K. (2017). Implementasi Support Vector Machine (Svm) Untuk Klasifikasi Jenis Daun Mangga Menggunakan Metode Gray Level Co-Occurrence Matrix. *Jurnal Ilmiah Manajemen Informatika dan Komputer*.
- Tingxi, Y., Li, W., Wuping, Z., Guofang, X., Jiwan, H., Fuzhong, L., & Chunqing, C. (2022). Predicting Phenotypes From High-Dimensional Genomes Using Gradient Boosting Decision Trees. *IEEE Access*.
- Wenyu, X., Na, D., Binjie, X., Yiwen, L., Yang C., & Zhengye, Z. (2019). Identification of Extremely Similar Animal Fibers Based on Matched Filter and HOG-SVM. *IEEE Access*.
- Zhang, D., & Gong, Y. (2020). The Comparison of LightGBM and XGBoost. *IEEE Access*.

LAMPIRAN-LAMPIRAN

Lampiran 1. Tahap Awal Library Bahasa Pemograman Python Pada Penelitian

```
import os
import numpy as np
import pandas as pd
import seaborn as sea
import tensorflow as tf
import seaborn as sns
from tensorflow.keras.applications.mobilenet import MobileNet
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.mobilenet import
preprocess_input
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
import lightgbm as lgb
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
confusion_matrix
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
from matplotlib import pyplot
import matplotlib.gridspec as gridspec
from sklearn.model_selection import GridSearchCV,
cross_val_score, StratifiedKFold, learning_curve
from sklearn.metrics import jaccard_score
from sklearn.metrics import roc_curve
import yellowbrick
from yellowbrick.classifier import PrecisionRecallCurve,
ROCAUC, ConfusionMatrix
from sklearn.metrics import classification_report,
accuracy_score
from yellowbrick.model_selection import LearningCurve,
FeatureImportances
```

Lampiran 2. Proses Pemograman Pemanggilan Data Pada Penelitian

```

sea.set_style("darkgrid")
from google.colab import drive
drive.mount('/content/drive/')
path = "drive/MyDrive/hog/"
for dirname, _, filenames in os.walk(path):
    for filename in filenames:
        print(os.path.join(dirname, filename))
classes = ["capmodium", "collectricum", "normal"]
path = "drive/MyDrive/hog/"

file_path = [os.path.join(path, "capmodium/hog_0001.jpg"),
             os.path.join(path, "collectricum/hog_0501.jpg"),
             os.path.join(path, "normal/hog_1001.jpg")]

fig = plt.figure(figsize=(10, 12))
gs = gridspec.GridSpec(nrows=3, ncols=2, figure=fig)

for i in range(3):
    y, x = i//2, i%2
    ax = fig.add_subplot(gs[y,x])
    ax.imshow(image.load_img(file_path[i]))
    ax.axis("off")
    ax.title.set_text(classes[i])

```

Lampiran 3. Proses Pemograman Ekstraksi Fitur Pada Penelitian

```
model = MobileNet(input_shape=(224,224,3), include_top=True)
model.summary()
vector = model.get_layer("reshape_2").output
feature_extraction = tf.keras.Model(model.input, vector)
X_list = []
Y_list = []

for f in range(3):
    folder_path = os.path.join(path, classes[f])
    for file in os.listdir(folder_path):
        file_path = os.path.join(folder_path, file)

        # check file extension, skip file if not jpg
        if not(file.endswith(".jpg")):
            continue

        img = image.load_img(file_path, target_size=(224,224))
        img_arr = image.img_to_array(img)
        img_arr_b = np.expand_dims(img_arr, axis=0)
        input_img = preprocess_input(img_arr_b)
        feature = feature_extraction.predict(input_img)

        X_list.append(feature.ravel())
        Y_list.append(f)
```


Lampiran 4. Proses Pemograman Pembagian Data Training dan Data Testing Pada Penelitian

```
X = np.asarray(X_list, dtype=np.float32)
Y = np.asarray(Y_list, dtype=np.float32)

for s in range(100):
    X, Y = shuffle(X, Y)

print("Shape of feature matrix X")
print(X.shape)
print("\nShape of label matrix Y")
print(Y.shape)

class_types, counts = np.unique(Y, return_counts=True)

print("\nClass labels")
print(class_types)
print("\nClass counts")
print(counts)
train_X, test_X, train_Y, test_Y = train_test_split(X, Y,
                                                    test_size=0.2,
                                                    stratify=Y,
                                                    random_state=0)

print("Shape of train_X")
print(train_X.shape)
print("\nShape of test_X")
print(test_X.shape)
```

Lampiran 5. Proses Pemograman Algoritma Adaboost

```

#ADABOOST

ABclassifier = AdaBoostClassifier(n_estimators=50,
learning_rate=1, random_state=0)
ABclassifier.fit(train_X, train_Y)
prediction_A = ABclassifier.predict(test_X)
ABAcc = accuracy_score(prediction_A, test_Y)
print('... AdaBoost Accuracy:'+'\033[1m
{:.2f}%'.format(ABAcc*100)+' ...')

# --- AdaBoost Classification Report ---
print('\n\033[1m'+'.: Classification Report'+'\033[0m')
print('*' * 25)
print(classification_report(test_Y, prediction_A))

# --- Performance Evaluation ---
print('\n\033[1m'+'.: Performance Evaluation'+'\033[0m')
print('*' * 26)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2,
figsize=(14, 10))

# --- AdaBoost Confusion Matrix ---
abcmatrix = ConfusionMatrix(ABclassifier, ax=ax1,
cmap='Blues',
title='AdaBoost Confusion Matrix')
abcmatrix.fit(train_X, train_Y)
abcmatrix.score(test_X, test_Y)
abcmatrix.finalize()

# --- AdaBoost ROC AUC ---
abrocauc = ROCAUC(ABclassifier, classes=['capmodium',
'collectricum', 'normal' ], ax=ax2,
title='AdaBoost ROC AUC Plot')
abrocauc.fit(train_X, train_Y)
abrocauc.score(test_X, test_Y)
abrocauc.finalize()

# --- AdaBoost Learning Curve ---
ablcl = LearningCurve(ABclassifier, ax=ax3, title='AdaBoost
Learning Curve')
ablcl.fit(train_X, train_Y)
ablcl.finalize()
plt.savefig('Adaboost.png')
plt.tight_layout();

```

Lampiran 6. Proses Pemograman Algoritma Gradient Boosting

```

#GRADIENT BOOST
gbc = GradientBoostingClassifier(n_estimators=33,
learning_rate=0.5, max_features=2, max_depth=2,
random_state=0)
gbc.fit(train_X, train_Y)
prediction_g = gbc.predict(test_X)
GBAcc = accuracy_score(prediction_g, test_Y)
print('... GradientBoosting Accuracy:'+'\033[1m
{: .2f}%'.format(GBAcc*100)+' ...')

# --- GradinetBoosting Classification Report ---
print('\n\033[1m'+'.: Classification Report'+'\033[0m')
print('*' * 25)
print(classification_report(test_Y, prediction_g))

# --- Performance Evaluation ---
print('\n\033[1m'+'.: Performance Evaluation'+'\033[0m')
print('*' * 26)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2,
figsize=(14, 10))

# --- GradientBoosting Confusion Matrix ---
abcmatrix = ConfusionMatrix(gbc, ax=ax1, cmap='Blues',
title='GradientBoosting Confusion
Matrix')
abcmatrix.fit(train_X, train_Y)
abcmatrix.score(test_X, test_Y)
abcmatrix.finalize()

# --- GradientBoosting ROC AUC ---
abrocauc = ROCAUC(gbc, classes=['capmodium', 'collectricum',
'normal' ], ax=ax2,
title='GradientBoosting ROC AUC Plot')
abrocauc.fit(train_X, train_Y)
abrocauc.score(test_X, test_Y)
abrocauc.finalize()

# --- GradientBoosting Learning Curve ---
ablcl = LearningCurve(gbc, ax=ax3, title='GradientBoosting
Learning Curve')
ablcl.fit(train_X, train_Y)
ablcl.finalize()
plt.savefig('GradienBOSST.png')
plt.tight_layout();

```

Lampiran 7. Proses Pemograman Algoritma XGBoost

```

#XGBOOST
xgb_clf =
XGBClassifier(random_state=0,learning_rate=1,n_estimators=44,
n_jobs=1)
xgb_clf.fit(train_X, train_Y)
prediction_x = xgb_clf.predict(test_X)

XGBAcc = accuracy_score(prediction_x, test_Y)
print('... XGBoost Accuracy:'+'\033[1m
{:.2f}%'.format(XGBAcc*100)+' ...')

# --- XGBoost Classification Report ---
print('\n\033[1m'+'.: Classification Report'+'\033[0m')
print('*' * 25)
print(classification_report(test_Y, prediction_x))

# --- Performance Evaluation ---
print('\n\033[1m'+'.: Performance Evaluation'+'\033[0m')
print('*' * 26)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2,
figsize=(14, 10))

# --- XGBoost Confusion Matrix ---
abcmatrix = ConfusionMatrix(xgb_clf, ax=ax1, cmap='Blues',
title='XGBoost Confusion Matrix')
abcmatrix.fit(train_X, train_Y)
abcmatrix.score(test_X, test_Y)
abcmatrix.finalize()

# --- XGBoost ROC AUC ---
abrocauc = ROCAUC(xgb_clf, classes=['capmodium',
'collectricum', 'normal' ], ax=ax2,
title='XGBoost ROC AUC Plot')
abrocauc.fit(train_X, train_Y)
abrocauc.score(test_X, test_Y)
abrocauc.finalize()

# --- XGBoost Learning Curve ---
ablcl = LearningCurve(xgb_clf, ax=ax3, title='XGBoost Learning
Curve')
ablcl.fit(train_X, train_Y)
ablcl.finalize()
plt.savefig('XGBoost.png')
plt.tight_layout();

```

Lampiran 8. Proses Pemograman Algoritma LightGBM

```

#LIGHTGBM
lgbm = lgb.LGBMClassifier(learning_rate=1, max_depth=-1,
n_estimators=4)
lgbm.fit(train_X, train_Y)
prediction_l = lgbm.predict(test_X)

LGAcc = accuracy_score(prediction_l, test_Y)
print('... LightGBM Accuracy:'+'\033[1m
{: .2f}%'.format(LGAcc*100)+' ...')

# --- LightGBM Classification Report ---
print('\n\033[1m'+'.: Classification Report'+'\033[0m')
print('*' * 25)
print(classification_report(test_Y, prediction_l))

# --- Performance Evaluation ---
print('\n\033[1m'+'.: Performance Evaluation'+'\033[0m')
print('*' * 26)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2,
figsize=(14, 10))

# --- LightGBM Confusion Matrix ---
abcmatrix = ConfusionMatrix(lgbm, ax=ax1, cmap='Blues',
title='LightGBM Confusion Matrix')
abcmatrix.fit(train_X, train_Y)
abcmatrix.score(test_X, test_Y)
abcmatrix.finalize()

# --- LightGBM ROC AUC ---
abrocauc = ROCAUC(lgbm, classes=['capmodium', 'collectricum',
'normal' ], ax=ax2,
title='LightGBM ROC AUC Plot')
abrocauc.fit(train_X, train_Y)
abrocauc.score(test_X, test_Y)
abrocauc.finalize()

# --- LightGBM Learning Curve ---
ablcl = LearningCurve(lgbm, ax=ax3, title='LightGBM Learning
Curve')
ablcl.fit(train_X, train_Y)
ablcl.finalize()

plt.savefig('LightGBM.png')

plt.tight_layout();

```

Lampiran 9. Proses Pemograman Perubahan Citra Ke Hog

```

import numpy as np
import pandas as pd
import cv2
import cv2 as cv
from skimage import color
import matplotlib.pyplot as plt
%matplotlib inline
import cv2
import cv2 as cv
from skimage.io import imread, imshow
from skimage.transform import resize
from skimage.feature import hog
from skimage import data, exposure
import os
for dirname, _, filenames in os.walk('../TA/daun'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
print('Done!')
import time
start = time.time()
capmodium = imread('../TA/daun/capmodium/PicsArt_07-06-03.15.49.jpg')
imshow(capmodium)
print(capmodium.shape)
end = time.time()
print("waktu komputerasi ",end - start , "per detik")
collectricum = imread('../TA/daun/collectricum/PicsArt_07-07-01.37.15.jpg')
imshow(collectricum)
print(collectricum.shape)
normal = imread('../TA/daun/normal/PicsArt_07-10-05.38.06.jpg')
imshow(normal)
print(normal.shape)
rc1 = resize(capmodium, (282,400))
imshow(rc1)
print(rc1.shape)
rc2 = resize(collectricum, (282,400))
imshow(rc2)
print(rc2.shape)
rc3 = resize(normal, (282,400))
imshow(rc3)
print(rc3.shape)
fd, hc1 = hog(rc1, orientations=9, pixels_per_cell=(8, 8),

```

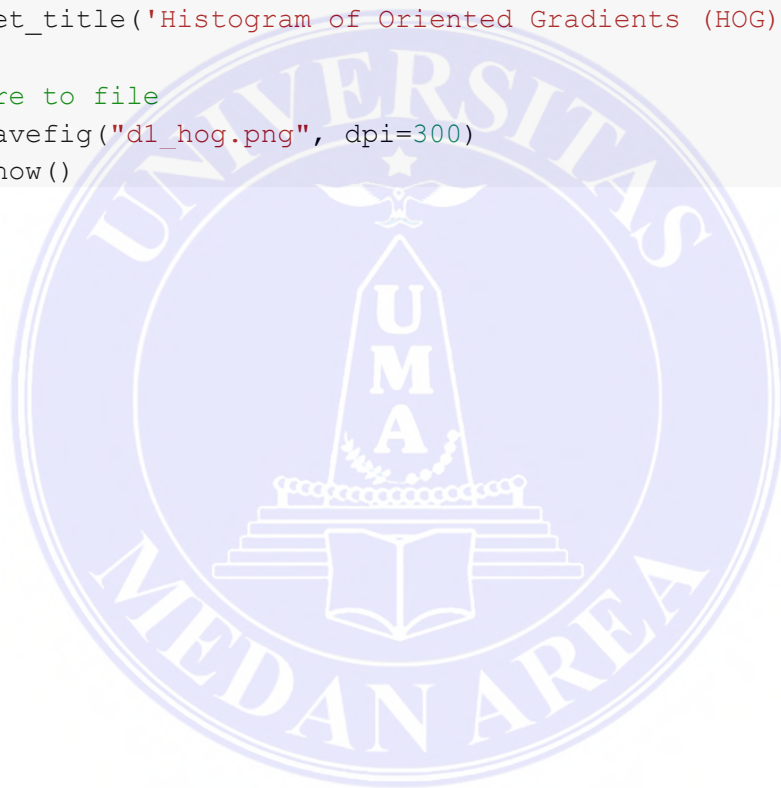
```
        cells_per_block=(2, 2), visualize=True,
multichannel=True)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 7),
sharex=True, sharey=True)

ax1.imshow(rc3, cmap=plt.cm.gray)
ax1.set_title('Daun Normal')

# Rescale histogram for better display
hc3_rescaled = exposure.rescale_intensity(hc3, in_range=(0,
50))

ax2.imshow(hc3_rescaled, cmap=plt.cm.gray)
ax2.set_title('Histogram of Oriented Gradients (HOG)')

# store to file
plt.savefig("d1_hog.png", dpi=300)
plt.show()
```



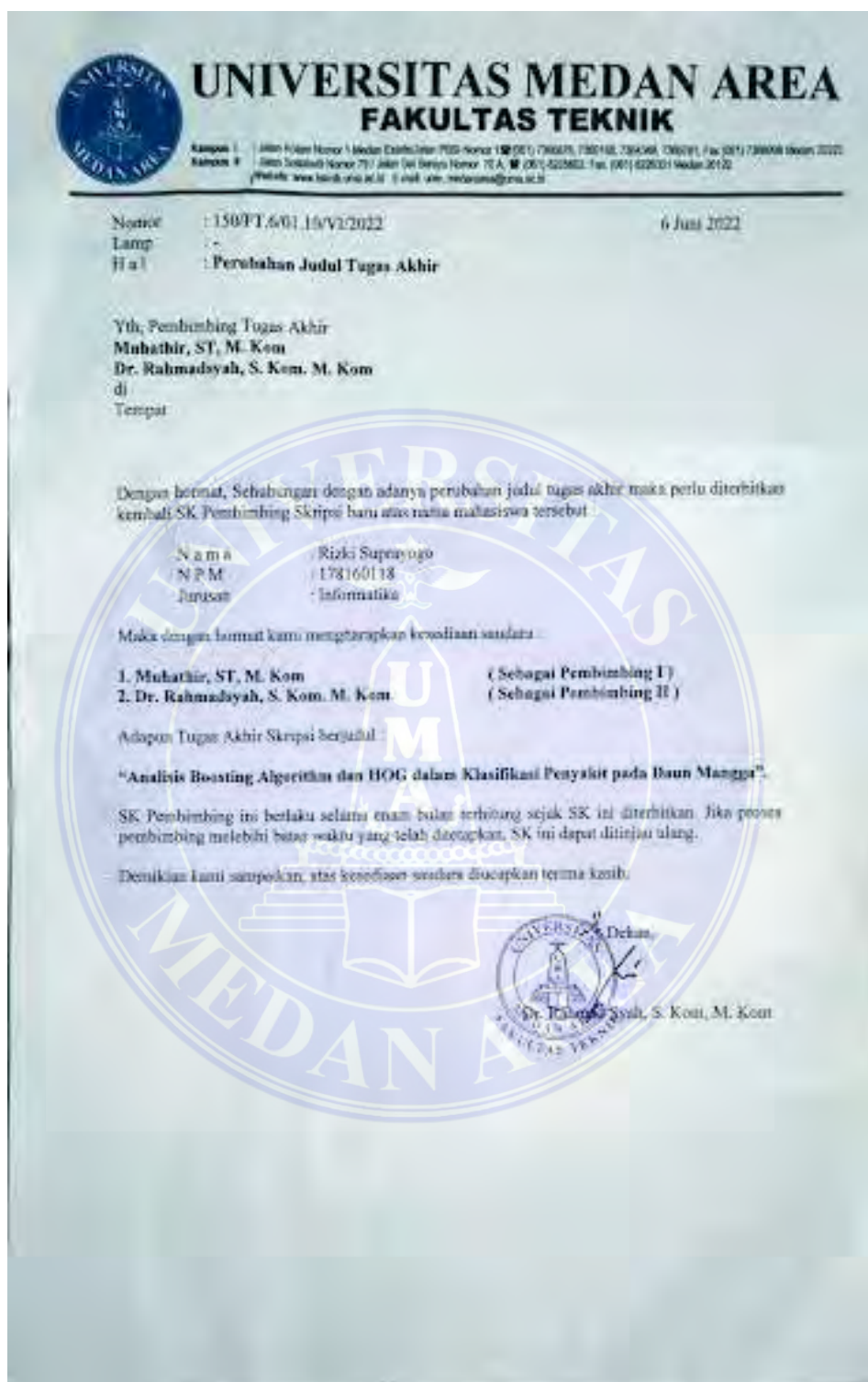
Lampiran 10. Data Daun



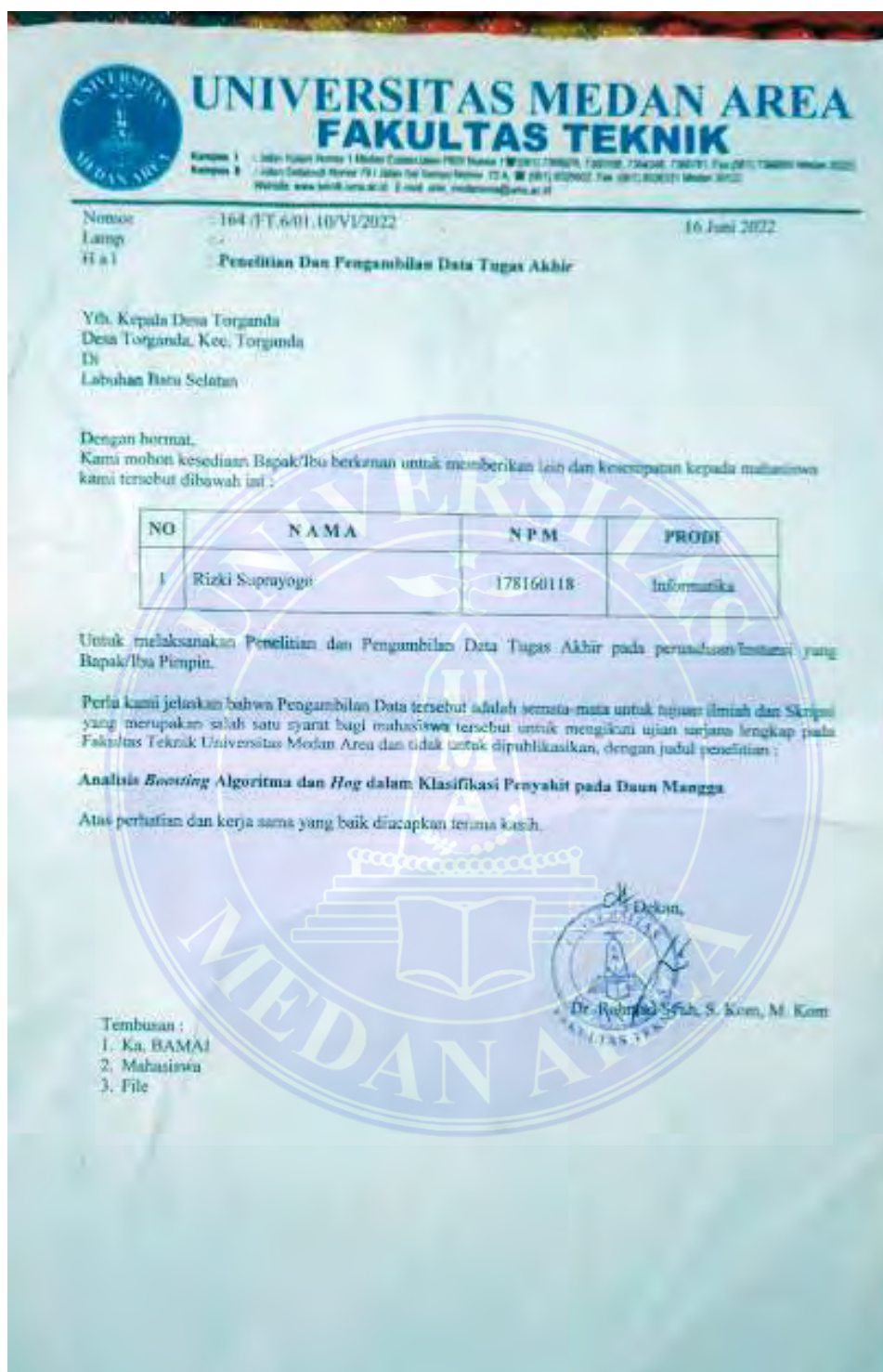
Keseluruhan data dapat di download melalui

https://drive.google.com/drive/folders/1QH9-9oOc_l4vfTlG9Xnn1eE7tWWiK66C?usp=drive_link

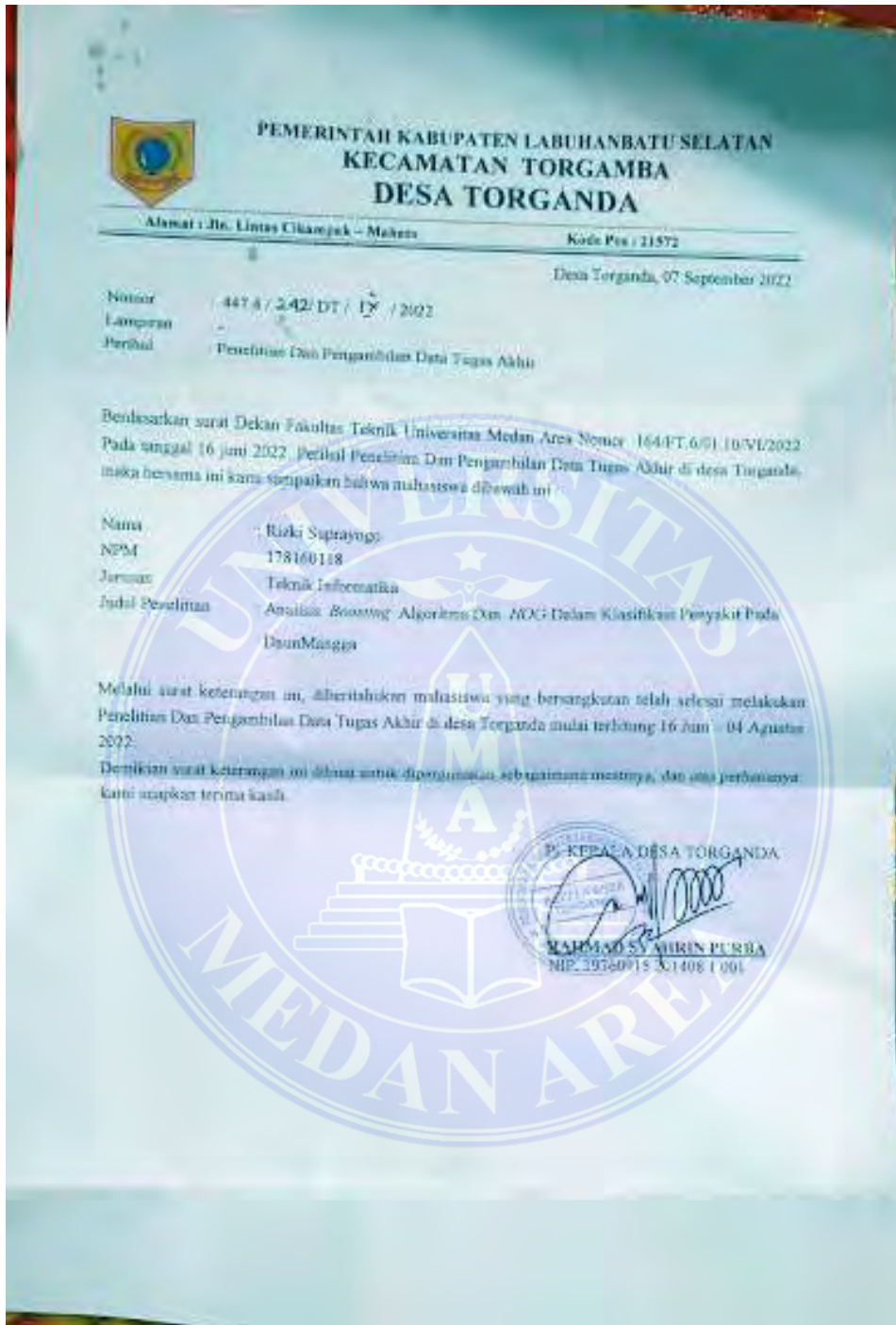
Lampiran 11. SK Dosen Pembimbing



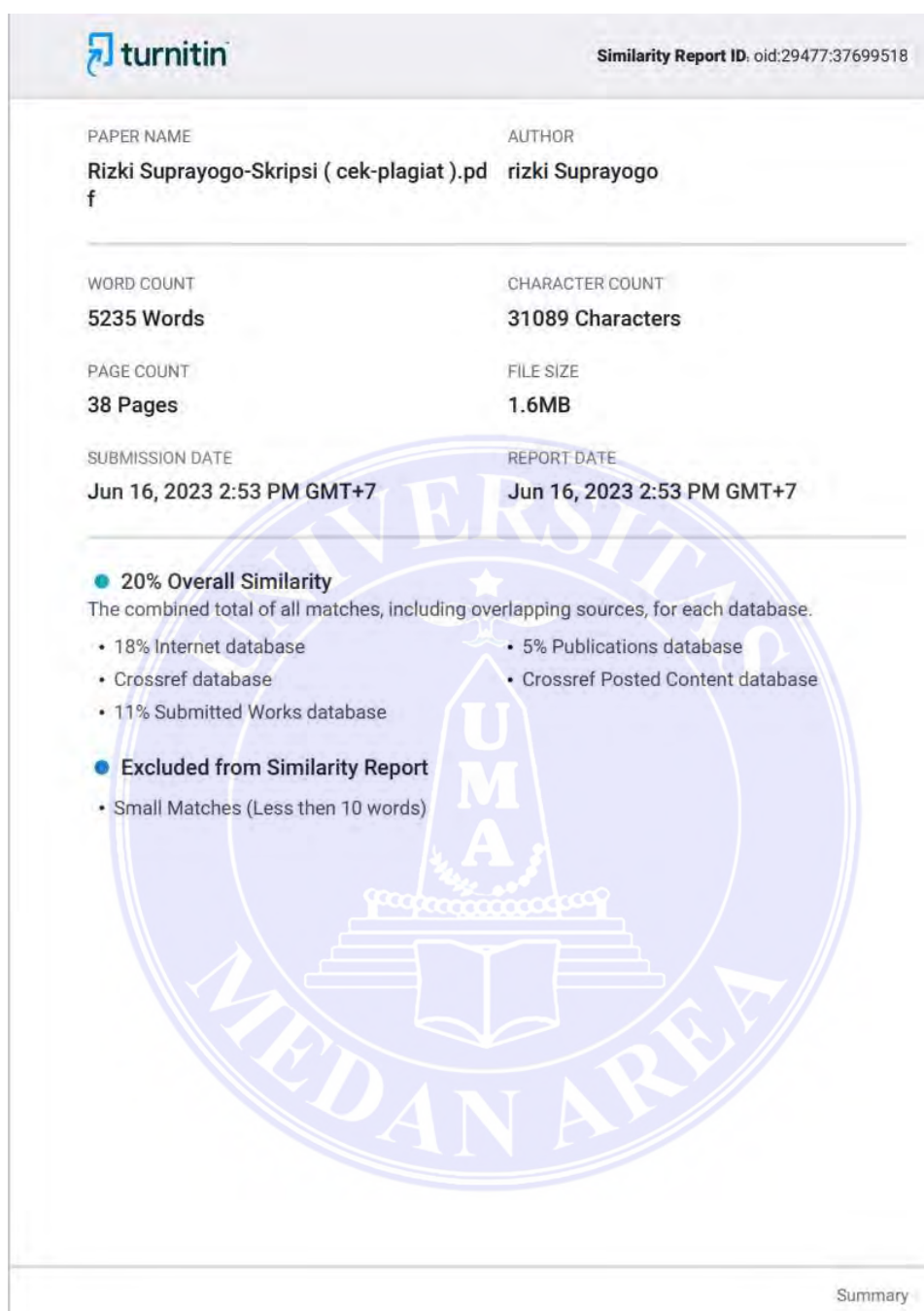
Lampiran 12. Surat Riset



Lampiran 13. Surat Selesai Riset



Lampiran 14. Turnitin



The image shows a Turnitin Similarity Report for a document titled "Rizki Suprayogo-Skripsi (cek-plagiat).pdf" by the author "rizki Suprayogo f". The report includes the following statistics:

PAPER NAME	AUTHOR
Rizki Suprayogo-Skripsi (cek-plagiat).pdf	rizki Suprayogo f

WORD COUNT	CHARACTER COUNT
5235 Words	31089 Characters

PAGE COUNT	FILE SIZE
38 Pages	1.6MB

SUBMISSION DATE	REPORT DATE
Jun 16, 2023 2:53 PM GMT+7	Jun 16, 2023 2:53 PM GMT+7

20% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.

- 18% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 11% Submitted Works database

Excluded from Similarity Report

- Small Matches (Less than 10 words)

The report is overlaid with a large, semi-transparent watermark of the Universitas Medan Area logo, which features a central emblem with the letters "UMA" and the text "UNIVERSITAS MEDAN AREA" around the perimeter.

Summary