KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 6, No. 1, Januari (2025), pp. 343-356

Klasifikasi Kualitas Produk Mesin Pertanian Berdasarkan Evaluasi Kinerja Algoritma Random Forest

Irma Hakim[1], Asdi Asdi[2], Mhd. Dicky Syahputra Lubis[3], Mely Novasari Harahap[4], Lokot Ridwan Batu Bara[5]*
[1]Universitas Muhammadiyah Makassar, Makassar, Indonesia
[2]Universitas Muhammadiyah Makassar, Makassar, Indonesia
[3]Universitas Tjut Nyak Dhien, Medan, Indonesia
[4]STAI UISU Pematangsiantar, Pematangsiantar, Indonesia
[5]Universitas Asahan, Kisaran, Indonesia
E-mail: campus_gardenia@yahoo.co.id[1], asdi@unismuh.ac.id[2], dickylubis@utnd.ac.id[3], melynovasariharahap7675@gmail.com[4], lokotridwan36@gmail.com[5]

Abstract

This study aims to classify product quality in the agricultural industry using the Random Forest algorithm. The data used includes various inspection result parameters, such as dimensions, weight, product color, quality status, defect image, inspection time, temperature, machine speed, and indicator lights. The model is developed to classify products into "good" and "defective" categories, and is evaluated based on accuracy metrics and confusion matrix analysis. The results show that the Random Forest model is able to achieve an accuracy of 85% in classifying product quality. Based on the confusion matrix, the model has a perfect prediction rate for good quality products (100% precision) and several misclassifications in the defect category. Feature importance analysis shows that the parameters of inspection time, machine temperature, and defect image are the most significant factors in determining product quality. This study proves that the Random Forest algorithm can be a reliable tool to support the product quality inspection process in the agricultural industry, with further integration into IoT-based systems, this approach can improve the efficiency of the inspection process, reduce manual errors, and ensure more consistent product quality standards.

Keywords: *Classification, Random Forest, Industrial Technology, Product Quality, Agricultural Machinery*

Abstrak

Penelitian ini bertujuan untuk melakukan klasifikasi kualitas produk dalam industri pertanian dengan menggunakan algoritma Random Forest. Data yang digunakan mencakup berbagai parameter hasil inspeksi, seperti dimensi, berat, warna produk, status kualitas, citra defek, waktu inspeksi, suhu, kecepatan mesin, dan lampu indikator. Model dikembangkan untuk mengklasifikasikan produk ke dalam kategori "baik" dan "cacat", serta dievaluasi berdasarkan metrik akurasi dan analisis confusion matrix. Hasil penelitian menunjukkan bahwa model Random Forest mampu mencapai akurasi sebesar 85% dalam mengklasifikasikan kualitas produk. Berdasarkan confusion matrix, model memiliki tingkat prediksi sempurna terhadap produk berkualitas baik (precision 100%) dan beberapa kesalahan klasifikasi pada kategori cacat. Analisis feature importance menunjukkan bahwa parameter waktu inspeksi, suhu mesin, dan citra defek merupakan faktor paling signifikan dalam menentukan kualitas produk. Studi ini membuktikan bahwa algoritma Random Forest dapat menjadi alat yang andal untuk mendukung proses inspeksi kualitas produk dalam industri pertanian, dengan integrasi lebih lanjut ke dalam sistem berbasis IoT, pendekatan ini dapat meningkatkan efisiensi proses inspeksi,

ISSN: 2720-992X

mengurangi kesalahan manual, dan memastikan standar kualitas produk yang lebih konsisten.

Kata Kunci: *Klasifikasi, Random Forest, Teknologi Industri, Kualitas Produk, Mesin Pertanian*

1. Pendahuluan

Pertanian merupakan salah satu sektor strategis yang memiliki peran vital dalam memenuhi kebutuhan pangan dunia [1]–[5]. Sebagai tulang punggung ketahanan pangan, sektor ini terus berkembang untuk menjawab tantangan global, seperti pertumbuhan populasi dan perubahan iklim [6]–[8]. Salah satu upaya untuk meningkatkan efisiensi dan produktivitas di sektor pertanian adalah dengan memanfaatkan teknologi modern, termasuk mesin pertanian [9]. Penggunaan mesin pertanian secara luas memungkinkan peningkatan hasil panen sekaligus pengurangan beban kerja manusia. Namun, kualitas mesin yang digunakan memainkan peran penting dalam menentukan keberhasilan proses produksi. Mesin pertanian berkualitas rendah dapat menyebabkan berbagai masalah serius, seperti kerusakan alat selama operasional, penurunan produktivitas hasil panen, hingga peningkatan biaya pemeliharaan [10]. Akibatnya, masalah ini dapat menghambat keberlanjutan usaha pertanian dan merugikan para petani. Oleh karena itu, inspeksi kualitas produk mesin pertanian menjadi langkah penting yang tidak dapat diabaikan. Inspeksi ini bertujuan untuk memastikan bahwa mesin yang digunakan mampu beroperasi secara optimal, tahan lama, dan mendukung efisiensi proses produksi [11].

Saat ini, metode inspeksi kualitas produk masih banyak bergantung pada proses manual atau pendekatan konvensional berbasis aturan sederhana. Meskipun metode ini telah lama digunakan, efektivitasnya sering kali terbatas. Proses manual membutuhkan waktu yang cukup lama dan melibatkan penilaian subjektif dari operator, sehingga rentan terhadap kesalahan manusia. Kesalahan ini tidak hanya memengaruhi akurasi hasil inspeksi tetapi juga dapat meningkatkan biaya operasional dan menurunkan efisiensi produksi. Pendekatan ini menjadi kurang relevan dalam menghadapi kebutuhan industri modern yang menuntut kecepatan dan presisi tinggi. Sementara itu, beberapa metode otomatis yang telah diterapkan, seperti inspeksi berbasis threshold sederhana atau algoritma statistik tradisional, juga memiliki keterbatasan. Metode ini umumnya hanya mampu menangani pola sederhana dalam data inspeksi dan sering kali gagal mengidentifikasi pola-pola kompleks yang terdapat dalam data multidimensi. Pada bidang industri pertanian, data inspeksi mesin sering kali bersifat heterogen dan kompleks, mencakup berbagai parameter seperti suhu, berat, kecepatan, dan citra visual, yang sulit diolah dengan pendekatan konvensional. Keterbatasan ini menciptakan celah penelitian (*research gap*) yang perlu diisi melalui pengembangan metode yang lebih cerdas, cepat, dan akurat.

Penelitian ini mengusulkan penggunaan algoritma *Random Forest* sebagai solusi untuk mengatasi keterbatasan metode inspeksi kualitas produk yang ada saat ini. Algoritma *Random Forest* merupakan salah satu pendekatan pembelajaran mesin yang terkenal dengan kemampuannya dalam menangani data yang bersifat multidimensi dan kompleks [12]–[14]. Kemampuannya membangun sejumlah pohon keputusan (*decision trees*) secara acak dan menggabungkan hasilnya, algoritma ini mampu memberikan prediksi yang lebih andal dan akurat dibandingkan metode tradisional [15]. Salah satu keunggulan utama dari algoritma *Random Forest* adalah kemampuannya untuk mengenali pola-pola kompleks dalam data, baik untuk variabel numerik maupun kategorikal [16]. Hal ini memungkinkan model untuk menangani data inspeksi yang sering kali terdiri atas berbagai jenis variabel, seperti suhu, kecepatan mesin, berat, atau bahkan data berbasis citra. Selain itu, algoritma ini juga dilengkapi dengan fitur analisis *feature importance* yang sangat berguna untuk mengidentifikasi faktor-faktor kritis yang paling memengaruhi kualitas produk [17].

Informasi ini dapat menjadi dasar untuk pengambilan keputusan strategis dalam perbaikan kualitas produk di masa mendatang. Dibandingkan dengan metode konvensional atau pendekatan otomatis berbasis algoritma statistik sederhana, penggunaan *Random Forest* menawarkan berbagai keunggulan, seperti tingkat akurasi yang lebih tinggi, fleksibilitas dalam pengaturan parameter model, dan ketahanan terhadap *overfitting* [18].

Beberapa penelitian-penelitian sebelumnya yang melatarbelakangi dilakukan nya penelitian ini diantaranya: Penelitian dengan memanfaatkan algoritma *Random Forest* untuk memprediksi hasil panen padi di desa Minanga, Sulawesi Barat. Kriteria yang digunakan dan berdasarkan data yang ada sebelumnya seperti: luas lahan, jenis pupuk, jumlah bibit, hama dan gulma, curah hujan, sistem penanaman padi yang digunakan (jajar legowo), serta pengendalian hama dan gulma. Hasil nya, bahwa variabel importance yang memiliki nilai paling tinggi yakni variabel luas lahan, dengan nilai akurasi sebesar 95,11% [19]. Selanjutnya penelitian untuk menganalisis produktivitas padi di Kabupaten Batang pada periode 2018-2022 menggunakan algoritma *Random Forest*. Berbagai variabel seperti indeks vegetasi, curah hujan, suhu udara, dan jenis tanah digunakan untuk memprediksi hasil panen. Hasil penelitian menunjukkan bahwa model memiliki tingkat akurasi 91%, dengan AUC dalam kategori "*Excellent Classification*". *Analisis feature importance* mengungkapkan bahwa indeks vegetasi dan curah hujan adalah faktor utama yang memengaruhi produktivitas [20]. Penelitian berikutnya menggunakan algoritma *Random Forest* untuk klasifikasi pencemaran udara di Jakarta. Hasil nya performa model dari algoritma *Random Forest* mendapatkan nilai precision, *recall*, dan F1-score yang sempurna yaitu 100% disemua kelas dan AUC juga sebesar 100%, lalu pada data test pada nilai precision untuk setiap kelas juga sangat tinggi yaitu 99%, dan AUC sebesar 99,96% [21]. Selanjutnya penelitian dengan memanfaatkan *Random Forest* untuk klasifikasi kualitas benih jagung. Berdasarkan 1.417 sampel benih jagung, diperoleh 695 sampel benih jagung berlabel Baik dan 722 sampel berlabel Buruk. Hasil klasifikasi pada penelitian ini berupa akurasi sebesar 100% [22]. Penelitian berikutnya menggunakan algoritma *Random Forest* untuk klasifikasi tingkat risiko diabetes. Berdasarkan hasil seleksi rangking variabel, teridentifikasi bahwa terdapat 300 pasien dengan risiko diabetes rendah dan 20 pasien dengan risiko diabetes tinggi. Selain itu, model yang digunakan menunjukkan kemampuan klasifikasi yang sangat baik, dengan tingkat akurasi mencapai 98%. Nilai *Area Under Curve* (AUC) yang diperoleh adalah 100%, yang termasuk dalam kategori "Klasifikasi Sangat Baik," karena model mampu membedakan secara sempurna antara kelas positif dan negatif [23].

Berdasarkan hal tersebut, maka penelitian ini bertujuan mengevaluasi model klasifikasi kualitas produk mesin pertanian berdasarkan hasil inspeksi menggunakan algoritma *Random Forest*, karena algoritma tersebut mampu melakukan klasifikasi dengan baik. Secara khusus, penelitian ini bertujuan untuk melihat akurasi prediksi kualitas, mengidentifikasi parameter inspeksi yang paling signifikan, dan mengusulkan pendekatan yang dapat diintegrasikan ke dalam sistem inspeksi otomatis berbasis teknologi industri 4.0. Sehingga, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam meningkatkan efisiensi dan efektivitas proses inspeksi kualitas di sektor pertanian.

2. Metodologi Penelitian

2.1. Data Penelitian

Pada penelitian ini, dataset penelitian yang digunakan adalah data dummy berdasarkan dataset produk dan hasil inspeksi. Data ini disimulasikan untuk merepresentasikan berbagai parameter inspeksi kualitas produk mesin pertanian, seperti dimensi, berat, warna produk, status kualitas, citra defek, waktu inspeksi, suhu, kecepatan mesin, dan lampu indikator. Penggunaan data dummy bertujuan untuk menguji efektivitas algoritma *Random Forest* dalam mengklasifikasikan kualitas mesin pertanian tanpa harus

bergantung pada data asli yang mungkin memiliki keterbatasan akses atau jumlah yang terbatas, dengan data dummy ini, penelitian dapat mengevaluasi performa model dalam mengenali pola-pola yang ada dalam data inspeksi serta mengidentifikasi variabel-variabel yang paling berpengaruh terhadap klasifikasi kualitas produk. Selain itu, data dummy memungkinkan validasi awal terhadap model sebelum diterapkan pada data nyata. Hasil analisis dari model ini dapat memberikan gambaran tentang seberapa baik algoritma *Random Forest* bekerja dalam mengklasifikasikan kualitas produk mesin pertanian, yang nantinya dapat diterapkan dalam skenario industri pertanian yang sesungguhnya untuk meningkatkan efisiensi inspeksi dan kontrol kualitas. Tabel 1 berikut ini akan menyajikan dataset penelitian yang terdiri dari 100 record sebagai sampel.

Tabel 1. Data Produk dan Hasil Inspeksi

No	ID Produk	Jenis Produk	Dimensi (cm)	Berat (kg)	Warna Produk	Status Kualitas	Citra Defek	Waktu Inspeksi (menit)	Suhu (°C)	Kecepatan Mesin (RPM)	Lampu Indikator
1	P001	Gearbox	15x12x10	5,2	Silver	Baik	0	0,25	24	3200	1
2	P002	Motor	20x18x15	7,8	Hitam	Cacat	1	0,30	28	2800	0
3	P003	Roda	25x25x10	4,3	Biru	Baik	0	0,20	22	3000	1
4	P004	Komponen Elektronik	10x8x5	1,5	Merah	Baik	0	0,15	23	3500	1
5	P005	Panel Kontrol	30x20x8	6,0	Putih	Cacat	1	0,18	26	3300	0
6	P006	Kabel	40x5x3	0,5	Hitam	Baik	0	0,10	25	3600	1
7	P007	Motor	18x14x12	6,2	Abu-abu	Baik	0	0,22	27	2900	1
8	P008	Gearbox	16x14x12	5,6	Silver	Cacat	1	0,30	30	3200	0
9	P009	Panel Kontrol	32x22x10	7,5	Hitam	Baik	0	0,25	24	3100	1
10	P010	Komponen Elektronik	12x8x5	1,3	Merah	Cacat	1	0,12	28	3300	0
11	P011	Roda	28x28x12	5,0	Biru	Baik	0	0,23	25	3000	1
12	P012	Gearbox	14x10x8	4,8	Silver	Baik	0	0,20	23	3100	1
13	P013	Motor	21x18x14	7,0	Hitam	Cacat	1	0,27	26	2800	0
14	P014	Komponen Elektronik	13x9x6	1,7	Merah	Baik	0	0,16	25	3200	1
15	P015	Panel Kontrol	33x24x9	7,2	Putih	Baik	0	0,20	22	3300	1
16	P016	Kabel	45x6x4	0,6	Hitam	Baik	0	0,09	27	3500	1
17	P017	Motor	19x15x13	6,4	Abu-abu	Cacat	1	0,25	28	3000	0
18	P018	Gearbox	17x13x10	5,4	Silver	Baik	0	0,22	24	3200	1
19	P019	Roda	26x26x11	4,6	Biru	Cacat	1	0,28	25	3100	0
20	P020	Komponen Elektronik	11x7x5	1,4	Merah	Baik	0	0,14	24	3500	1
…	…	…	…	…	…	…	…	…	…	…	…
100	P100	Gearbox	18x44x16	3,7	Merah	Baik	1	0,99	32	2708	0

Dataset penelitian yang disajikan pada Tabel 1 terdiri dari berbagai variabel yang digunakan untuk menganalisis kualitas produk mesin pertanian. ID Produk berfungsi sebagai kode unik untuk membedakan setiap produk, sementara Jenis Produk mengidentifikasi kategori mesin pertanian yang diuji. Dimensi (cm) dan Berat (kg) mencerminkan ukuran fisik produk, sedangkan Warna Produk dapat digunakan sebagai indikator standar kualitas. Status Kualitas menunjukkan hasil evaluasi produk, seperti "Baik" atau "Cacat". Citra Defek (0 = Tidak, 1 = Ada) merepresentasikan apakah terdapat cacat visual yang terdeteksi. Waktu Inspeksi (menit) mencatat durasi inspeksi, sedangkan Suhu (°C) dan Kecepatan Mesin (RPM) menjadi faktor lingkungan dan operasional yang dapat memengaruhi kualitas mesin. Selain itu, Lampu Indikator (0 = Mati, 1 = Menyala) digunakan sebagai indikator kondisi mesin. Semua variabel ini diolah menggunakan algoritma *Random Forest* untuk mengidentifikasi pola dan melihat tingkat akurasi klasifikasi kualitas produk mesin pertanian.

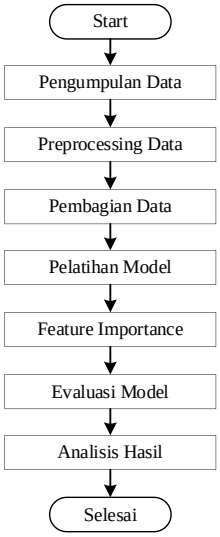
2.2. Algoritma *Random Forest*

Algoritma *Random Forest* merupakan metode machine learning yang menggabungkan hasil dari beberapa pohon keputusan (*decision trees*) untuk meningkatkan akurasi prediksi dan mengurangi risiko *overfitting* [24]. Algoritma ini dapat digunakan untuk tugas klasifikasi maupun regresi. Pohon keputusan menggunakan konsep pohon *N-aray* untuk menyelesaikan masalah

tersebut. Masing-masing node daun di pohon sesuai dengan label kelas. Fitur diwakili pada node internal yang disebut node keputusan yang membantu menghasilkan prediksi dari serangkaian pemisahan berbasis fitur. *Random Forest* menggunakan kumpulan pohon keputusan untuk membangun hutan yang mengatasi keterbatasan algoritma pohon keputusan yang memiliki kecenderungan untuk overfit data pelatihan. *Random Forest* membangun beberapa pohon keputusan acak dan menggabungkannya bersama untuk membuat model lebih akurat dan kurang sensitif terhadap data pelatihan. Model *Random Forest* dilatih dengan *Bagging method* yang merupakan kombinasi dari dua langkah, bootstrap dan agregasi.

2.3. Tahapan Penelitian

Tahapan penelitian merujuk pada serangkaian langkah yang diikuti dalam proses penelitian untuk mencapai tujuan yang telah ditetapkan. Proses ini bersifat iteratif dan mungkin perlu diulang atau disesuaikan berdasarkan temuan dan perkembangan selama penelitian berlangsung. Setiap tahap memiliki peran penting dalam memastikan penelitian dilakukan secara sistematis dan ilmiah. Tahapan yang dilakukan pada penelitian ini dapat dilihat pada gambar 1 berikut ini.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1 dapat dijelaskan bahwa tahapan penelitian dimulai dengan Pengumpulan Data, yakni menggunakan dataset Produk dan Hasil Inspeksi yang dapat dilihat pada Tabel 1. Tahapan selanjutnya adalah Preprocessing Data dengan menghapus data duplikat atau tidak relevan, Menangani data kosong (missing values) jika ada, dan normalisasi atau standarisasi data. Berikutnya adalah tahapan Pembagian Data (*Data Splitting*) berarti membagi dataset menjadi data latih (*training set*) dan data uji (*testing set*). Tahapan Pelatihan Model berarti melatih algoritma *Random Forest* dengan data latih. Selanjutnya dilakukan *Feature importance* (pemilihan fitur penting) untuk menentukan seberapa besar pengaruh setiap fitur (variabel independen) terhadap prediksi model. Tahapan berikutnya adalah Evaluasi Model yang berarti menghitung akurasi, *precision*, *recall*, dan *F1-score* menggunakan data uji serta menganalisis hasil menggunakan *Confusion Matrix*. Sedangkan tahapan Analisis Hasil berarti menentukan performa model dan faktor utama yang memengaruhi kualitas produk.

3. Hasil dan Pembahasan

3.1. Pengumpulan Data

Pengumpulan data dilakukan dengan memanfaatkan data dummy yang disusun berdasarkan dataset produk dan hasil inspeksi. Data ini disimulasikan untuk

menggambarkan berbagai parameter dalam inspeksi kualitas produk mesin pertanian, seperti ukuran, berat, warna, status kualitas, citra cacat, waktu inspeksi, suhu, kecepatan mesin, serta lampu indikator. Penggunaan data dummy bertujuan untuk mengevaluasi efektivitas algoritma *Random Forest* dalam mengklasifikasikan kualitas mesin pertanian tanpa ketergantungan pada data asli yang mungkin sulit diakses atau jumlahnya terbatas. Dataset yang digunakan dalam penelitian ini berupa file excel dan dapat dilihat pada Tabel 1. Pada langkah awal, akan dilakukan import *library* yang dibutuhkan dalam Python.

```
# Import Library
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
```

Gambar 2. Coding import library

Gambar 2 berisi kode program untuk mengimpor berbagai pustaka (*libraries*) yang digunakan dalam proses analisis dan pemodelan. *import pandas as pd* berarti mengimpor pandas untuk manipulasi dan analisis data dalam bentuk DataFrame. *import seaborn as sns* berarti mengimpor seaborn untuk visualisasi data yang lebih menarik dan informatif. *import matplotlib.pyplot as plt* berarti mengimpor *matplotlib* untuk membuat grafik dan plot. *from sklearn.model_selection import train_test_split* berarti mengimpor fungsi *train_test_split* dari *scikit-learn*, yang digunakan untuk membagi dataset menjadi data *training* dan *testing*. *from sklearn.ensemble import RandomForestClassifier* berarti mengimpor *Random Forest Classifier*, yaitu model berbasis *ensemble learning* yang terdiri dari banyak *decision tree* untuk tugas klasifikasi. *from sklearn.metrics import classification_report, accuracy_score, confusion_matrix* berarti mengimpor beberapa metrik evaluasi model seperti: *classification_report* (menampilkan metrik evaluasi seperti *precision*, *recall*, *f1-score*, dan *support*), *accuracy_score* (menghitung tingkat akurasi model), *confusion_matrix* (menampilkan matriks kebingungan untuk analisis performa model klasifikasi). *from sklearn.preprocessing import LabelEncoder* berarti mengimpor *LabelEncoder*, yang digunakan untuk mengonversi nilai kategori (teks) menjadi nilai numerik agar bisa digunakan dalam algoritma *machine learning*. Selanjutnya dataset akan dibaca agar dapat di proses selanjutnya. Kode programnya dapat dilihat pada Gambar 3 berikut ini:

```
# Load dataset
file_path = r'D:\\Produk dan Hasil Inspeksi.xlsx'
data = pd.read_excel(file_path, sheet_name='Sheet1')
```

Gambar 3. Coding membaca data penelitian

Gambar 3 merupakan kode program untuk memuat dataset dari file Excel ke dalam *DataFrame Pandas*. *file_path = r'D:\\Produk dan Hasil Inspeksi.xlsx'* berarti menyimpan lokasi file Excel yang akan dibaca. Tanda *r* sebelum string menunjukkan bahwa itu adalah *raw string*, yang mencegah karakter *escape* seperti *\n* atau *\t* berfungsi dalam *path*. *data = pd.read_excel(file_path, sheet_name='Sheet1')* berarti menggunakan fungsi *pd.read_excel()* untuk membaca file Excel yang berada di lokasi *file_path*. Parameter *sheet_name='Sheet1'* menunjukkan bahwa data yang diambil berasal dari lembar kerja bernama *Sheet1*. Hasil pembacaan ini disimpan dalam variabel data sebagai *DataFrame Pandas*, yang kemudian bisa digunakan untuk analisis lebih lanjut.

3.2. Preprocessing Data

Tahapan ini merupakan persiapan data sebelum digunakan dalam analisis atau pemodelan *machine learning*. Proses ini mencakup pembersihan data, transformasi, dan penyusunan ulang agar lebih siap digunakan. Beberapa langkah preprocessing yang dilakukan penelitian ini meliputi: menghapus data yang tidak relevan atau tidak memiliki kontribusi terhadap analisis, dan normalisasi untuk menyelaraskan skala data numerik. Kode programnya dapat dilihat pada Gambar 4 berikut ini:

```
# Data preprocessing
# Drop unnecessary columns
columns_to_drop = ['No', 'ID Produk', 'Dimensi (cm)']  # Remove irrelevant columns
data = data.drop(columns=columns_to_drop)
```

Gambar 4. Coding preprocessing data

Gambar 4 merupakan kode program preprocessing data dengan menghapus kolom yang dianggap tidak relevan. *columns_to_drop = ['No', 'ID Produk', 'Dimensi (cm)']* berarti daftar kolom yang akan dihapus adalah *"No", "ID Produk"*, dan *"Dimensi (cm)"*. Kode *data = data.drop(columns=columns_to_drop)* berarti menggunakan *fungsi drop ( )* dari *pandas* untuk menghapus kolom-kolom yang tidak diperlukan dari *DataFrame*, sedangkan *Parameter columns=columns_to_drop* memastikan bahwa hanya kolom dalam daftar tersebut yang dihapus. Setelah eksekusi kode ini, *DataFrame* data akan memiliki lebih sedikit kolom, hanya menyisakan kolom yang dianggap relevan untuk analisis lebih lanjut. Selanjutnya melakukan *encoding* variabel kategori menggunakan *LabelEncoder*, yang mengonversi nilai kategori (*string*) menjadi angka agar bisa digunakan dalam model *machine learning*, dapat dilihat pada Gambar 5 berikut ini:

```
# Encode categorical variables
label_encoders = {}
for col in ['Jenis Produk', 'Warna Produk', 'Status Kualitas']:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le
```

Gambar 5. Encoding variabel kategori

Berdasarkan Gambar 5, *label_encoders = {}* merupakan dictionary untuk menyimpan objek *LabelEncoder* dari setiap kolom. *for col in ['Jenis Produk', 'Warna Produk', 'Status Kualitas']* berarti melakukan iterasi pada tiga kolom kategori yang akan dikodekan. *le = LabelEncoder ()* berarti membuat objek *LabelEncoder*. Kode *data[col] = le.fit_transform (data[col])* berarti mengubah nilai kategori dalam kolom menjadi angka dengan *fit_transform( )*. Sedangkan kode *label_encoders[col] = le* berarti menyimpan *encoder* untuk digunakan nanti (misalnya untuk *inverse transform* atau decoding kembali). Selanjutnya dilakukan pendefinisian fitur (X) dan target (y) untuk pemodelan, yang dapat dilihat pada Gambar 6 berikut ini:

```
# Define features and target
X = data.drop(columns=['Status Kualitas'])  # Features
y = data['Status Kualitas']  # Target
```

Gambar 6. Coding definisi fitur dan target

Penjelasan Gambar 6, bahwa Kode *X = data.drop(columns=['Status Kualitas'])* berarti mengambil semua kolom kecuali "Status Kualitas" sebagai fitur (variabel independen). Kode *y = data['Status Kualitas']* berarti menetapkan "Status Kualitas" sebagai target (variabel dependen) yang akan diprediksi oleh model. Berdasarkan pemisahan ini, model

machine learning nantinya dapat dilatih menggunakan X untuk memprediksi nilai y. Tahapan ini merupakan persiapan sebelum proses *feature selection* atau pelatihan model.

3.3. Pembagian Data

Pembagian data merupakan langkah penting dalam *machine learning*, di mana dataset dibagi menjadi dua bagian utama: data pelatihan (*training data*) dan data pengujian (*testing data*). Tujuan dari pembagian ini adalah untuk melatih model menggunakan data pelatihan dan kemudian menguji kemampuannya untuk generalisasi dengan menggunakan data pengujian yang belum pernah dilihat sebelumnya oleh model. Pembagian data pada penelitian ini dapat dilihat pada Gambar 7 berikut ini:

```
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 7. Coding Pembagian Data

Berdasarkan Gambar 7 dapat dijelaskan bahwa *X_train* merupakan data pelatihan yang akan digunakan untuk melatih model. *X_test* merupakan data pengujian yang akan digunakan untuk menguji kinerja model setelah dilatih. *y_train* merupakan label atau target yang sesuai dengan data pelatihan (*X_train*). *y_test* merupakan label atau target yang sesuai dengan data pengujian (*X_test*). *X* merupakan data input (fitur atau variabel independen) yang akan digunakan untuk melatih model. *y* merupakan target atau label (variabel dependen) yang ingin diprediksi oleh model. *test_size=0.2* berarti menentukan proporsi data yang akan digunakan untuk data pengujian, dalam hal ini 20% dari total data akan digunakan sebagai data pengujian, sementara 80% sisanya digunakan untuk data pelatihan. Kode *random_state=42* berarti menetapkan nilai acak (*seed*) untuk memastikan bahwa pembagian data ini dapat diulang dengan hasil yang konsisten. Ini berguna saat kita ingin memastikan pembagian yang sama setiap kali kode dijalankan.

3.4. Pelatihan Model

Pada model *Random Forest*, pelatihan melibatkan proses pembentukan banyak pohon keputusan (*decision trees*) dari subset data pelatihan yang berbeda. Setiap pohon membuat keputusan yang berbeda, dan prediksi akhir dibuat berdasarkan hasil mayoritas dari pohon-pohon tersebut (dalam klasifikasi). Kode program pelatihan model dapat dilihat pada Gambar 8 berikut ini:

```
# Train Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

Gambar 8. Coding Pelatihan Model *Random Forest*

Berdasarkan Gambar 8, kode program *RandomForestClassifier(n_estimators=100, random_state=42)* berarti menginisialisasi model *Random Forest* dengan 100 pohon keputusan (parameter *n_estimators=100*), yang artinya model akan membangun 100 pohon untuk membuat prediksi. Parameter *random_state=42* memastikan bahwa pembagian data dan pemilihan fitur acak dapat diulang dengan hasil yang sama setiap kali model dilatih, sehingga memberikan konsistensi dalam eksperimen. Fungsi *fit(X_train, y_train)* kemudian melatih model dengan data pelatihan, di mana *X_train* adalah fitur input (data yang digunakan untuk memprediksi) dan *y_train* adalah label target (jawaban yang benar). Proses ini memungkinkan model untuk mempelajari pola dalam data dan menghasilkan model klasifikasi yang dapat memprediksi label dari data baru berdasarkan keputusan yang dibuat oleh pohon-pohon dalam hutan (*forest*).

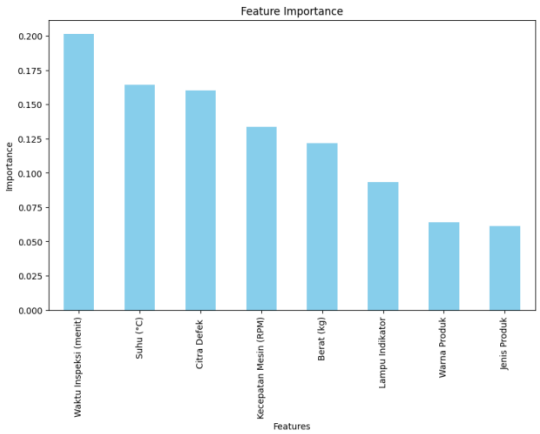
3.5. *Feature Importance*

Tahapan ini dilakukan untuk menentukan seberapa besar pengaruh setiap fitur (variabel independen) terhadap prediksi model. *Feature importance* dihitung berdasarkan seberapa sering suatu fitur digunakan dalam pembagian (*splitting*) di setiap pohon keputusan dan seberapa besar pengaruhnya dalam mengurangi *impurity* (Ketidakmurnian). Fitur dengan nilai *importance* yang lebih tinggi dianggap lebih berkontribusi dalam keputusan model dibandingkan fitur dengan nilai yang lebih rendah. Kode programnya dapat dilihat pada Gambar 9 berikut ini:

```
# Feature Importance
feature_importance = pd.Series(rf_model.feature_importances_, index=X.columns).sort_values(ascending=False)
plt.figure(figsize=(10, 6))
feature_importance.plot(kind='bar', color='skyblue')
plt.title('Feature Importance')
plt.ylabel('Importance')
plt.xlabel('Features')
plt.show()
```

Gambar 9. *Coding Feature Selection*

Kode program tersebut bertujuan untuk menganalisis dan memvisualisasikan *Feature importance* menggunakan model *Random Forest (rf_model).* Kode *rf_model.feature_importances_* berarti mengambil bobot kepentingan setiap fitur dari model *Random Forest. pd.Series(..., index=X.columns)* berarti mengubah hasil tersebut menjadi *Series* dengan fitur sebagai indeks. Kode *.sort_values(ascending=False)* berarti mengurutkan fitur dari yang paling penting hingga yang paling tidak penting. Kode *plt.figure(figsize=(10,6))* berarti mengatur ukuran grafik. *feature_importance.plot(kind='bar', color='skyblue')* berarti membuat grafik batang untuk menampilkan kepentingan fitur. Kode *plt.title('Feature importance')* berarti menambahkan judul grafik. Kode *plt.ylabel('Importance') & plt.xlabel('Features')* berarti memberikan label sumbu Y dan X. Sedangkan kode *plt.show()* berarti menampilkan grafik. Hasil *Features Importance* dapat dilihat pada Gambar 10 berikut ini:



Gambar 10. Hasil *Feature importance*

Gambar 10 merupakan visualisasi *Feature importance* dari model *Random Forest*, yang menunjukkan tingkat kepentingan setiap fitur dalam memprediksi target. Sumbu X mewakili fitur-fitur dalam dataset, sedangkan sumbu Y menunjukkan nilai kepentingan (importance) dari masing-masing fitur. Pada grafik, terlihat bahwa "Waktu Inspeksi (menit)" adalah fitur paling berpengaruh dalam model, diikuti oleh "Suhu (°C)" dan "Citra Defek". Sementara itu, "Warna Produk" dan "Jenis Produk" memiliki pengaruh paling rendah. Informasi ini dapat digunakan untuk *Feature Selection*, di mana fitur dengan kepentingan rendah dapat dihapus untuk meningkatkan efisiensi dan akurasi model.

3.6. Evaluasi Model

Setelah model dilatih menggunakan data pelatihan melalui metode *.fit()*, langkah berikutnya adalah evaluasi model, yaitu menguji kinerja model menggunakan data yang belum pernah dilihat sebelumnya (data pengujian).

```
# Make predictions
y_pred = rf_model.predict(X_test)
```

Gambar 11. Coding membuat prediksi

Pada Gambar 11, kode *rf_model.predict(X_test)* digunakan untuk memprediksi label dari data pengujian (*X_test*) dengan model yang telah dilatih sebelumnya. Hasil prediksi ini (*y_pred*) akan digunakan untuk mengevaluasi kinerja model dengan membandingkannya dengan label sebenarnya dari data pengujian (*y_test*), menggunakan metrik seperti akurasi, *precision*, *recall*, *F1-score*, atau *confusion matrix*.

```
# Evaluate model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:\n")
print(classification_report(y_test, y_pred))
```

Gambar 12. Coding Evaluasi Model

Kode program pada Gambar 12 digunakan untuk mengevaluasi kinerja model klasifikasi dengan mengukur akurasi dan menampilkan *classification report*, yang mencakup metrik evaluasi lainnya. Pertama, *accuracy_score(y_test, y_pred)* menghitung akurasi model dengan membandingkan label sebenarnya dari data uji (*y_test*) dengan hasil model prediksi (*y_pred*), di mana akurasi didefinisikan sebagai jumlah prediksi yang benar dibagi total jumlah sampel. Hasilnya kemudian ditampilkan dalam format desimal dua angka dengan *print(f"Accuracy: {accuracy:.2f}")*. Selanjutnya, *classification_report(y_test, y_pred)* digunakan untuk mencetak laporan klasifikasi yang mencakup metrik seperti *precision*, *recall*, *F1-score*, dan *support* untuk setiap kelas dalam data. *Precision* mengukur seberapa banyak prediksi positif yang benar, *recall* menunjukkan seberapa baik model menemukan semua sampel positif, dan *F1-score* adalah rata-rata harmonik dari precision dan *recall*. Metrik ini membantu dalam mengevaluasi performa model lebih mendalam daripada hanya mengandalkan akurasi, terutama jika data tidak seimbang. Hasilnya dapat dilihat pada gambar *Classification Report* berikut ini:

```
Accuracy: 0.85

Classification Report:

              precision    recall  f1-score   support

           0       0.77      1.00      0.87        10
           1       1.00      0.70      0.82        10

    accuracy                           0.85        20
   macro avg       0.88      0.85      0.85        20
weighted avg       0.88      0.85      0.85        20
```

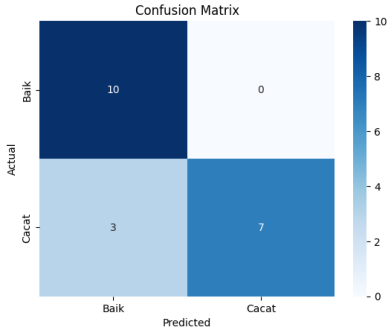
Gambar 13. *Classification Report*

Gambar 13 ini menunjukkan hasil evaluasi model klasifikasi dengan akurasi sebesar 85% serta *classification report* yang mencakup metrik *precision*, *recall*, *F1-score*, dan *support* untuk dua kelas, yaitu kelas 0 (Baik) dan kelas 1 (Cacat), dengan masing-masing memiliki 10 sampel data uji. Precision untuk kelas 0 adalah 0.77, artinya dari semua

prediksi kelas 0, 77% benar, sedangkan *recall*-nya 1.00 (100%), menunjukkan bahwa semua sampel sebenarnya kelas 0 berhasil teridentifikasi. Sebaliknya, kelas 1 memiliki precision 1.00 (100% atau semua prediksi kelas 1 benar), tetapi *recall* hanya 0.70 (70%), yang berarti ada 30% data kelas 1 yang salah diklasifikasikan sebagai kelas 0. *F1-score* adalah rata-rata harmonik dari *precision* dan *recall*, dengan nilai 0.87 (87%) untuk kelas 0 dan 0.82 (82%) untuk kelas 1. *Macro average* 0.85 (85%) menunjukkan rata-rata metrik antar kelas, sedangkan *weighted average* 0.85 (85%) mempertimbangkan jumlah sampel di setiap kelas. Berdasarkan laporan ini, model cukup baik dalam mengklasifikasikan kelas 0 tetapi masih memiliki kelemahan dalam mendeteksi kelas 1, yang terlihat dari *recall* yang lebih rendah.

3.7. Analisis Hasil

Analisis hasil model klasifikasi pada penelitian ini dilakukan berdasarkan *Confusion Matrix*, karena *confusion matrix* memberikan gambaran yang lebih rinci tentang bagaimana model mengklasifikasikan setiap kelas dibandingkan dengan label sebenarnya. Berdasarkan *confusion matrix*, kita dapat menghitung metrik evaluasi utama seperti *precision*, *recall*, *F1-score*, dan akurasi, yang telah ditampilkan sebelumnya dalam *classification report*. Hasil *Confusion Matrix* dapat dilihat pada Gambar 14 berikut ini:



Gambar 14. Confusion Matrix

Gambar 14 merupakan Confusion Matrix yang menunjukkan performa model klasifikasi dalam membedakan antara dua kelas, yaitu "Baik" dan "Cacat" berdasarkan prediksi dan data aktual. Baris menunjukkan nilai aktual, sedangkan kolom menunjukkan hasil prediksi model. Terdapat 10 sampel yang sebenarnya Baik dan diprediksi dengan benar sebagai Baik (*True Positives*, TP), sementara tidak ada sampel Baik yang salah diklasifikasikan sebagai Cacat (*False Negatives*, FN = 0). Untuk kelas Cacat, model berhasil mengidentifikasi 7 sampel dengan benar (*True Negatives*, TN), tetapi salah mengklasifikasikan 3 sampel Cacat sebagai Baik (*False Positives*, FP). Berdasarkan *confusion matrix* ini, dapat disimpulkan bahwa model memiliki kinerja yang sangat baik dalam mengidentifikasi objek yang Baik (100% *recall* pada kelas Baik), tetapi masih membuat kesalahan dalam mendeteksi beberapa sampel Cacat (hanya 70% *recall* pada kelas Cacat, karena 3 dari 10 sampel Cacat salah diklasifikasikan sebagai Baik). Untuk meningkatkan performa model, terutama dalam mendeteksi kategori Cacat, dapat dilakukan penyesuaian *threshold*, resampling data untuk menyeimbangkan kelas, atau mencoba model dengan arsitektur yang lebih kompleks.

4. Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa algoritma *Random Forest* mampu mengklasifikasikan kualitas produk mesin pertanian dengan tingkat akurasi yang cukup tinggi, yaitu sebesar 85%. Analisis menggunakan Confusion Matrix menunjukkan bahwa model ini memiliki tingkat presisi sempurna dalam mengidentifikasi produk

berkualitas baik, namun masih mengalami beberapa kesalahan klasifikasi pada kategori cacat. Faktor-faktor yang paling berpengaruh dalam proses klasifikasi adalah waktu inspeksi, suhu mesin, dan citra defek, sebagaimana ditunjukkan oleh analisis *feature importance*. Penerapan *Random Forest* dalam klasifikasi kualitas produk menunjukkan bahwa metode ini dapat diandalkan untuk meningkatkan efisiensi dan akurasi dalam proses inspeksi. Dibandingkan dengan metode inspeksi manual yang sering kali subjektif dan rentan terhadap kesalahan manusia, pendekatan berbasis kecerdasan buatan ini mampu mengurangi kesalahan dan memastikan standar kualitas produk yang lebih konsisten. Sehingga, penggunaan teknologi seperti ini dapat menjadi solusi efektif dalam industri pertanian untuk meningkatkan produktivitas dan mengurangi risiko kerugian akibat penggunaan mesin berkualitas rendah. Untuk pengembangan lebih lanjut, integrasi model ini ke dalam sistem berbasis *Internet of Things* (IoT) dapat menjadi langkah strategis guna meningkatkan efisiensi pemantauan kualitas produk secara otomatis dan *real-time*, dengan mengombinasikan teknologi ini dengan sensor dan sistem pemrosesan data yang lebih canggih, diharapkan dapat tercipta sistem inspeksi yang lebih akurat, cepat, dan dapat diterapkan dalam skala industri yang lebih luas. Oleh karena itu, penelitian di masa depan dapat difokuskan pada pengujian model ini dalam berbagai kondisi operasional serta peningkatan teknik feature engineering guna memperoleh hasil klasifikasi yang lebih optimal.

Daftar Pustaka

[1] Y. Jararweh, S. Fatima, M. Jarrah, and S. AlZu'bi, 'Smart and sustainable agriculture: Fundamentals, enabling technologies, and future directions', *Computers and Electrical Engineering*, vol. 110, no. September, p. 108799, 2023, doi: 10.1016/j.compeleceng.2023.108799.

[2] P. C. Pandey and M. Pandey, 'Highlighting the role of agriculture and geospatial technology in food security and sustainable development goals', *Sustainable Development*, vol. 31, no. 5, pp. 3175–3195, 2023, doi: 10.1002/sd.2600.

[3] Michael Alurame Eruaga, 'Policy strategies for managing food safety risks associated with climate change and agriculture', *International Journal of Scholarly Research and Reviews*, vol. 4, no. 1, pp. 21–32, 2024, doi: 10.56781/ijsrr.2024.4.1.0026.

[4] I. Hakim, A. Asdi, and T. Afriliansyah, 'Implementasi Algoritma Komputasi Linear Regression untuk Optimasi Prediksi Hasil Pertanian', *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 5, no. 3, pp. 1423–1434, 2024, doi: 10.30645/kesatria.v5i3.460.

[5] A. O. Hidayat, I. W. Ayu, and M. Wildan, 'Kajian Literatur: Dampak Kebijakan Pemerintah Dalam Bidang Pertanian Untuk Kesejahteraan Ekonomi Petani', *Jurnal Riset Kajian Teknologi dan Lingkungan*, vol. 7, no. 1, pp. 241–245, 2024, doi: 10.58406/jrktl.v7i1.1693.

[6] A. Ghosh, A. Kumar, and G. Biswas, 'Chapter 1 - Exponential population growth and global food security: challenges and alternatives', in *Bioremediation of Emerging Contaminants from Soils*, Prasann Kumar, A. L. Srivastav, V. Chaudhary, E. D. van Hullebusch, and R. Busquets, Eds., 2024, pp. 1–20. doi: 10.1016/B978-0-443-13993-2.00001-3.

[7] D. K. Pandey and R. Mishra, 'Towards sustainable agriculture: Harnessing AI for global food security', *Artificial Intelligence in Agriculture*, vol. 12, no. June, pp. 72–84, 2024, doi: 10.1016/j.aiia.2024.04.003.

[8] A. A. Chandio, Y. Jiang, A. Amin, M. Ahmad, W. Akram, and F. Ahmad, 'Climate change and food security of South Asia: fresh evidence from a policy perspective using novel empirical analysis', *Journal of Environmental Planning and Management*, vol. 66, no. 1, pp. 169–190, 2023, doi:

10.1080/09640568.2021.1980378.
[9] J. Saputri Mendrofa, M. W. Zendrato, N. Halawa, E. E. Zalukhu, and N. K. Lase, 'Peran Teknologi dalam Meningkatkan Efisiensi Pertanian', *Tumbuhan: Publikasi Ilmu Sosiologi Pertanian Dan Ilmu Kehutanan*, vol. 1, no. 3, pp. 01–12, 2024, doi: 10.62951/tumbuhan.v1i3.111.
[10] K. Ruslan, 'Produktivitas Tanaman Pangan dan Hortikultura. Makalah Kebijakan No. 37', *Center for Indonesian Policy Studies*, no. 37, pp. 1–48, 2021. [Online]. Available: www.cips-indonesia.org
[11] H. C. Mayana, M. Mulyadi, N. Nofriadi, D. F. Z, and E. K. Putri, 'Rancang Bangun Mesin Penepung Pakan Ternak Multifungsi Untuk Kelompok Tani Berkah Mandiri', *Jurnal Teknik Mesin*, vol. 17, no. 2, pp. 209–215, 2024, doi: 10.30630/jtm.17.2.1653.
[12] H. Mirzadeh and H. Omranpour, 'Extended *Random Forest* for multivariate air quality forecasting', *International Journal of Machine Learning and Cybernetics*, 2024, doi: 10.1007/s13042-024-02329-7.
[13] M. Al-Imran *et al.*, 'Evaluating Machine Learning Algorithms For Breast Cancer Detection: A Study On Accuracy And Predictive Performance', *The American Journal of Engineering and Technology*, vol. 6, no. 9, pp. 22–33, 2024, doi: 10.37547/tajet/Volume06Issue09-04.
[14] J. Fisher, S. Allen, G. Yetman, and L. Pistolesi, 'Assessing the influence of landscape conservation and protected areas on social wellbeing using *Random Forest* machine learning', *Scientific Reports*, vol. 14, no. 1, pp. 1–16, 2024, doi: 10.1038/s41598-024-61924-4.
[15] I. Maulita, C. R. A. Widiawati, and A. M. Wahid, 'Analisis Komparatif Linear Regression, *Random Forest*, dan Gradient Boosting untuk Prediksi Banjir', *Jurnal Pendidikan Dan Teknologi Indonesia*, vol. 4, no. 8, pp. 369–379, 2024, doi: 10.52436/1.jpti.599.
[16] S. Sobari, A. I. Purnamasari, A. Bahtiar, and K. Kaslani, 'Meningkatkan Model Prediksi Kelulusan Santri Tahfidz di Pondok Pesantren Al-Kautsar Menggunakan Algoritma *Random Forest*', *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 13, no. 1, pp. 732–738, 2025, doi: 10.23960/jitet.v13i1.5704.
[17] N. Pratama, E. Setyati, J. Santoso, and A. P. Rini, '*Random Forest* Classifier Dengan Grid Search Untuk Klasifikasi Hasil Psikotes Pada Rekrutmen Dosen dan Tenaga Kependidikan', *Konvergensi*, vol. 20, no. 1, pp. 18–24, 2024, doi: 10.30996/konv.v20i1.10874.
[18] Z. Zhou, C. Qiu, and Y. Zhang, 'A comparative analysis of linear regression, neural networks and *Random Forest* regression for predicting air ozone employing soft sensor models', *Scientific Reports*, vol. 13, no. 1, pp. 1–23, 2023, doi: 10.1038/s41598-023-49899-0.
[19] N. Nur, F. Wajidi, S. Sulfayanti, and W. Wildayani, 'Implementasi Algoritma *Random Forest* Regression untuk Memprediksi Hasil Panen Padi di Desa Minanga', *Jurnal Komputer Terapan*, vol. 9, no. 1, pp. 58–64, 2023, doi: 10.35143/jkt.v9i1.5917.
[20] A. R. Masdian, N. Bashit, and F. Hadi, 'Analisis Produktivitas Padi Menggunakan Algoritma Machine Learning *Random Forest* Di Kabupaten Batang Tahun 2018 - 2022', *Elipsoida : Jurnal Geodesi dan Geomatika*, vol. 6, no. 1, pp. 43–51, 2023, doi: 10.14710/elipsoida.2023.19023.
[21] R. Firdaus *et al.*, 'Implementasi Algoritma *Random Forest* Untuk Klasifikasi Pencemaran Udara di Wilayah Jakarta Berdasarkan Jakarta Open Data', *Jurnal FASILKOM (teknologi inFormASi dan ILmu KOMputer)*, vol. 14, no. 2, pp. 520–525, 2024, doi: 10.37859/jf.v14i2.7669.
[22] M. Zahara, A. Putra, F. Candra, and E. Prakasa, 'Klasifikasi Kualitas Varietas

Benih Jagung Bima 20 Menggunakan Metode *Random Forest*', *Jurnal Teknlogi Informatika dan Komputer*, vol. 10, no. 2, pp. 367–385, 2024, doi: 10.37012/jtik.v10i2.2177.

[23] A. Setiawan, Z. H. Nst, Z. Khairi, and L. Efrizoni, 'Klasifikasi Tingkat Risiko Diabetes Menggunakan Algoritma *Random Forest*', *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 7, no. 2, pp. 263–271, 2024, doi: 10.36595/jire.v7i2.1259.

[24] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, 'An improved *Random Forest* based on the classification accuracy and correlation measurement of decision trees', *Expert Systems with Applications*, vol. 237, no. Part B, p. 121549, 2024, doi: 10.1016/j.eswa.2023.121549.

[25] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, 'A hybrid approach for Bangla sign language recognition using deep transfer learning model with *Random Forest* classifier', *Expert Systems with Applications*, vol. 213, no. Part B, p. 118914, 2023, doi: 10.1016/j.eswa.2022.118914.