JURNAL RESPONSIF, Vol. 5 No.2 Agustus 2023, pp. 407~415
E-ISSN**:** 2685-6964

PERBANDINGAN ALGORITMA *MACHINE LEARNING* UNTUK KLASIFIKASI *DRY BEAN DATASET*

Gregian Bayu Anugrah[1], Raisya Nadzira Zahirahtush Shafa [2], Adang Kurniawan[3]

[1]Universitas Adhirajasa Reswara Sanjaya
e-mail: ryanbayuanugrah@gmail.com

[2]Universitas Adhirajasa Reswara Sanjaya
e-mail: raisyanadzira82@gmail.com

[3]Universitas Adhirajasa Reswara Sanjaya
*****e-mail korespondensi: adangkurniawan99@gmail.com

Abstrak

Indonesia merupakan negara agraris dengan mayoritas penduduknya bekerja pada sektor pertanian yang memiliki peran penting dalam meningkatkan perekonomian negara dan memenuhi kebutuhan pangan masyarakat. *Dry Bean* memegang peranan penting pada sektor pertanian, pengolahan makanan, dan juga ekspor-impor. Memungkinkan para pemangku kepentingan untuk mengoptimalkan produksi, memastikan kontrol kualitas, meningkatkan daya saing pasar, mendorong inovasi, dan memenuhi permintaan konsumen dengan memanfaatkan *dry bean* yang tersedia. Dataset *Dry* Bean merupakan salah satu data publik yang ada pada web UCI *Machine Learning Repository.* Tujuan penelitian ini adalah untuk mengetahui performa yang paling baik diantara enam algoritma yang diuji dengan menggunakan model *rapid miner* dan *confussion matrix.* Enam algoritma klasifikasi yang diuji tersebut yaitu *Naive Bayes, Neural Network, Decision Tree, Random Forest*, *K-Nearest Neighbors (*K-NN), *Linear Discriminant Analysis.* Perbedaan dalam penelitian ini adalah terletak pada objek dan waktu penelitian, literatur dan teori yang digunakan serta hasil penelitian. Berdasarkan hasil penelitian dapat diketahui bahwa algoritma *Neural Network* memiliki nilai akurasi performa paling tinggi sebesar 92,68% dan nilai akurasi performa terendah adalah algoritma *K-Nearest Neighbors (K-NN)* dengan nilai 78,32%.

Kata Kunci: *Algoritma, Machine Learning, Dry Bean, Dataset*.

Abstract

The majority of people in Indonesia, an agricultural nation, are employed in the agriculture industry, which plays a significant part in boosting the nation's economy and providing for the needs of its citizens. In the agriculture industry, food processing, and export-import trade, dry beans are crucial. Utilizing the available dry beans enables stakeholders to maximize output, guarantee quality control, improve market competitiveness, spur innovation, and satisfy consumer demand. One of the publicly accessible datasets on the UCI Machine Learning Repository website is the Dry Bean dataset. The goal of this study was to compare the six algorithms utilizing the quick miner and confusion matrix models to find which performed the best. The six algorithms under test for categorization were Naive Bayes, Neural Network, Decision Tree, Random Forest, K-Nearest Neighbors (K-NN), Linear Discriminant Analysis. The difference in this study lies in the object and time of research, the literature and theory used and the results of the research. Based on the research results, it can be seen that the Neural Network algorithm has the highest performance accuracy value of 92.68% and the lowest performance accuracy value is the K-Nearest Neighbors (K-NN) algorithm with a value of 78.32%.

Keywords*: Algorithm, Machine Learning, Dry Bean, Dataset*.

Naskah diterima 31 Juli 2023; direvisi 25 Agustus 2023; diterbitkan 31 Agustus 2023

1. Pendahuluan

Perkembangan teknologi memberikan banyak dampak positif pada berbagai sektor. Semakin banyaknya *software,* sistem, teknologi baru yang mendukung dan membantu dalam mengolah data. Muncul suatu kebutuhan untuk dapat menghasilkan informasi dari data yang telah tersedia. Setiap informasi yang tersedia menjadi hal yang penting dalam menentukan setiap keputusan pada situasi tertentu. Hal ini menyebabkan penyediaan informasi menjadi sarana untuk dianalisa dan diringkas menjadi suatu pengetahuan yang bermanfaat ketika pengambilan suatu keputusan dilakukan (Ardiyansyah et al., 2018).

Indonesia merupakan negara agraris dengan mayoritas penduduknya bekerja pada sektor pertanian yang memiliki peran penting dalam meningkatkan perekonomian negara dan memenuhi kebutuhan pangan masyarakat (Setiadi et al., 2020). Tanaman pangan adalah semua jenis tanaman yang di dalamnya terdapat karbohidrat, protein, mineral, dan vitamin sebagai sumber energi manusia (Kaunang et al., 2018). Tanaman pangan juga sebagai tanaman utama yang dikonsumsi oleh manusia sebagai asupan energi untuk kelangsungan hidup tubuh manusia.

Saat ini *Dry Bean* sebagai salah satu jenis tanaman pangan memegang peranan penting pada sektor pertanian, pengolahan makanan, dan juga ekspor-impor. Memungkinkan para pemangku kepentingan untuk mengoptimalkan produksi, memastikan kontrol kualitas, meningkatkan daya saing pasar, mendorong inovasi, dan memenuhi permintaan konsumen dengan memanfaatkan *dry bean* yang tersedia (Mehta et al., 2023). Tidak sulit menemukan data mengenai perkembangan pertumbuhan *dry bean*. Dataset *Dry* Bean merupakan salah satu data publik yang ada pada web UCI *Machine Learning Repository* dapat digunakan sebagai acuan untuk melakukan penelitian.

Salah satu cara mendapatkan informasi atau pola dari kumpulan data yang besar adalah dengan menggunakan teknik-teknik dalam *data mining*. *Data mining* adalah proses menelusuri pengetahuan baru, pola dan tren yang dipilih dari jumlah data yang besar yang disimpan dalam repositori atau tempat penyimpanan dengan menggunakan teknik pengenalan pola serta statistik dan teknik matematika (Widiastuti et al., 2007). Penelitian ini menggunakan algoritma klasifikasi. Membutuhkan sebuah data *training* untuk menemukan sebuah pola. Kemudian dari data *training* tersebut akan diketahui performa pada setiap algoritma klasifikasi. Sehingga dapat ditentukan performa yang terbaik di antara algoritma yang digunakan (Ardiyansyah et al., 2018).

Machine learning merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan komputer untuk belajar dari data dan pengalaman tanpa harus secara eksplisit diprogram. *Machine Learning* telah membawa perubahan besar dalam berbagai bidang, termasuk klasifikasi data. Salah satu tugas penting dalam *machine learning* adalah klasifikasi, yaitu memprediksi kategori atau label dari suatu data berdasarkan informasi yang ada. Tujuannya untuk mengembangkan algoritma dan model komputer yang dapat mengidentifikasi pola dalam data dan membuat keputusan atau prediksi berdasarkan pola tersebut (Murphy, 2012).

Terdapat banyak algoritma atau metode yang dapat di gunakan dalam *machine learning.* Pilihan algoritma tergantung pada karakteristik dataset, ukuran data, dan tujuan dari klasifikasi. Performa algoritma dapat bervariasi tergantung pada berbagai faktor, termasuk *tuning parameter, preprocessing data*, dan jumlah data penelitian yang tersedia. Adanya algoritma *machine learning* dapat membantu membandingkan antara satu algoritma dengan algoritma lainnya yang lebih akurat. Pemilihan algoritma yang tepat dapat mempengaruhi kinerja model klasifikasi yang dihasilkan (Lestari et al., 2023).

Hal yang membedakan penelitian ini dengan sebelumnya adalah terdapat enam algoritma klasifikasi yaitu *Naive Bayes, Neural Network, Decision Tree, Random Forest*, *K-Nearest Neighbors* (K-NN), *Linear Discriminant Analysis.* Penelitian ini bertujuan untuk mengetahui performa yang paling baik diantara enam algoritma tersebut dengan menggunakan model *rapid miner* dan *confussion matrix.*

2. Metode Penelitian

Metode penelitian adalah cara yang digunakan oleh peneliti dalam pengumpulan data penelitiannya (Arikunto, 2006). Penelitian yang dilakukan menggunakan enam algoritma untuk menentukan performa paling terbaik. *Software* yang digunakan dalam penelitian ini adalah *rapid miner*, dataset *dry bean* diolah melalui UCI *Machine Learning Repository.*

Tahapan penelitian ini yaitu pengumpulan data, pengolahan data, metode yang diusulkan, pengujian metode dan hasil penelitian. Berikut beberapa metode yang digunakan dalam penelitian ini
.

Naïve Bayes

Algoritma Naive Bayes merupakan metode klasifikasi yang didasarkan pada teorema Bayes dengan asumsi "naif" bahwa fitur-fitur yang digunakan untuk klasifikasi adalah independen satu sama lain. Meskipun asumsi ini jarang benar dalam dunia nyata, algoritma ini tetap efektif dalam banyak kasus dan sering digunakan dalam pengenalan pola dan analisis teks.

Rumus Naive Bayes untuk mengklasifikasikan suatu data ke dalam kelas tertentu adalah sebagai berikut :

$$P(C|X)=P(X)P(X|C)\cdot P(C) \quad (1)$$

Keterangan :

- $P(C|X)$ adalah probabilitas kelas C, diberikan data X.
- $P(X|C)$ adalah probabilitas data X, diberikan kelas C. Ini melibatkan asumsi naif bahwa semua fitur adalah independen satu sama lain, sehingga dapat dihitung sebagai $P(x1|C)\times P(x2|C)\times\ldots\times P(xn|C)$, dengan P1,P2,…,$x1$,$x2$,…,$xn$ adalah fitur-fitur dari data X.
- $P(C)$ adalah probabilitas kelas C sebelum melihat data X.
- $P(X)$ adalah probabilitas data X.

Algoritma Naive Bayes bekerja dengan menghitung probabilitas untuk setiap kelas dan memilih kelas dengan probabilitas tertinggi sebagai prediksi akhir (Duda & Hart, 2006).

Neural Network

Algoritma Neural Network merupakan model matematika yang terinspirasi oleh cara kerja otak manusia. Ini terdiri dari jaringan neuron buatan (unit pemrosesan informasi) yang terhubung satu sama lain dan memiliki kemampuan untuk belajar dari data. Setiap neuron menerima input, mengalikan input dengan bobot tertentu, menjumlahkannya, dan kemudian melewatkan outputnya melalui fungsi aktivasi untuk menghasilkan output akhir. Selama pelatihan, bobot-bobot ini disesuaikan agar jaringan dapat melakukan prediksi dengan lebih baik.

Rumus kalkulasi pada setiap neuron dalam jaringan adalah sebagai berikut :

1. Input ke Neuron :

$$\sum=1+z=\sum i=1n(xi\times wi)+b \quad (2)$$

Keterangan :

- z adalah total input ke neuron.
- xi adalah input ke-i dari neuron.
- wi adalah bobot ke-i dari input tersebut.
- b adalah bias, yaitu nilai yang ditambahkan ke total input. Bias ini memungkinkan jaringan untuk belajar memprediksi nilai ketika semua inputnya adalah nol.

2. Output dari Neuron setelah melewati fungsi aktivasi :

$$a=f(z) \quad (3)$$

Keterangan :

- a adalah output dari neuron setelah melewati fungsi aktivasi.
- $(.)f(.)$ adalah fungsi aktivasi yang memetakan total input (z) ke output (a). Fungsi aktivasi ini memberikan non-linearitas pada jaringan, yang memungkinkan jaringan untuk memodelkan hubungan kompleks dalam data.

Jaringan saraf tiruan terdiri dari beberapa lapisan, seperti lapisan input, lapisan tersembunyi (hidden layer), dan lapisan output. Ketika data melewati jaringan dari lapisan input ke lapisan output melalui lapisan tersembunyi, proses kalkulasi dan pembelajaran terjadi melalui metode seperti *feedforward* (meneruskan informasi dari input ke output) dan *backpropagation* (menyesuaikan bobot berdasarkan kesalahan prediksi) (Ian Goodfellow, Yoshua Bengio, 2016).

Decision Tree

Algoritma *Decision Tree* merupakan metode pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. Pohon keputusan berisi serangkaian keputusan hirarki yang dibuat berdasarkan fitur-fitur dari data input, sehingga akhirnya menghasilkan keputusan atau prediksi.

Proses pembuatan pohon keputusan dimulai dengan membagi data training menjadi subset-subset yang lebih kecil berdasarkan fitur-fitur tertentu. Setiap pemisahan ini berusaha untuk mengidentifikasi fitur yang paling informatif sehingga membantu dalam memisahkan data ke dalam kelas-kelas yang berbeda atau dalam mengestimasi nilai regresi.

Proses ini berlangsung secara berulang hingga seluruh data telah dikelompokkan dengan baik atau hingga mencapai kriteria penghentian tertentu.

Rumus yang digunakan untuk menghitung keuntungan (gain) informasi dari setiap pemisahan dan memilih fitur terbaik untuk melakukan pemisahan adalah:

1. Gini Impurity (untuk klasifikasi):

Gini1−∑=1 ; 2Gini(*p*)=1−∑*i*=1*Kpi*2 (4)

Keterangan :

- *p* adalah distribusi probabilitas dari kelas pada set data yang dipertimbangkan.
- *K* adalah jumlah kelas yang ada.
- *pi* adalah probabilitas dari kelas ke-i dalam distribusi.

2. Entropy (untuk klasifikasi) :

Entropy=−∑=1log2
Entropy(*p*)=−∑*i*=1*Kpi*log2(*pi*) (5)

Keterangan :

- *p*, *K*, *pi* memiliki arti yang sama seperti dalam rumus Gini Impurity.

3. Mean Squared Error (untuk regresi):

MSE=1∑=1(2MSE(*y*)=*N*1∑*i*=1*N*(*yi*−*y*¯)2 (6)

Keterangan :

- *y* adalah nilai target pada set data yang dipertimbangkan.
- *N* adalah jumlah data dalam set tersebut.
- *yi* adalah nilai target pada data ke-i.
- *y*¯ adalah rata-rata dari nilai target pada set data.

Pada setiap langkah pemisahan, algoritma memilih fitur yang memberikan penurunan gini impurity atau entropy yang paling besar (untuk klasifikasi) atau penurunan MSE yang paling besar (untuk regresi) (Tan et al., 2016).

Random Forest

Algoritma *Random Forest* adalah metode pembelajaran mesin yang berdasarkan konsep "*ensemble learning*" (pembelajaran gabungan). Konsep menggabungkan beberapa pohon keputusan (decision trees) untuk meningkatkan kinerja prediksi dan mengurangi *overfitting*. Setiap pohon keputusan dibangun berdasarkan subset acak dari data latihan dan subset acak dari fitur-fitur yang tersedia.

Proses pembuatan *Random Fores*t melibatkan beberapa langkah:

1. Memilih subset acak dari data latihan dengan penggantian (bootstrapping).
2. Memilih subset acak dari fitur-fitur yang tersedia untuk digunakan dalam pembangunan pohon keputusan.
3. Membangun pohon keputusan pada subset data dan subset fitur yang dipilih.
4. Mengulangi langkah 1 hingga 3 untuk membangun sejumlah pohon keputusan yang diinginkan.
5. Ketika melakukan prediksi, setiap pohon memberikan hasil prediksi, dan hasil akhir diambil berdasarkan mayoritas atau rata-rata prediksi dari semua pohon.

Rumus untuk prediksi pada Random Forest (dalam kasus klasifikasi) adalah sebagai berikut :

Prediksi=mode(Prediksi(1),Prediksi(2),…, Prediksi(……))
Prediksi=mode(Prediksi(t1),Prediksi(t2),…, Prediksi(tn)) (7)

Keterangan :

- Prediksi(ti) adalah prediksi dari pohon keputusan i.
- mode(…)mode(…) mengambil nilai yang paling sering muncul di antara semua prediksi pohon (Breiman, 2001).

K-*Nearest Neighbors* (K-NN)

Menurut (Christopher & Nasrabadi, 2006) Algoritma K-NN (K-*Nearest Neighbors*) adalah metode pembelajaran mesin untuk tugas klasifikasi dan regresi. Metode ini bekerja dengan mencari k titik data terdekat dari data uji yang ingin diprediksi, kemudian menggunakan mayoritas kelas (dalam klasifikasi) atau rata-rata nilai (dalam regresi) dari tetangga terdekat tersebut untuk melakukan prediksi.

Langkah-langkah umum dalam algoritma K-NN adalah sebagai berikut :

1. Hitung jarak antara data uji yang ingin diprediksi dengan setiap data latihan menggunakan metrik jarak tertentu, seperti Euclidean distance atau Manhattan distance.
2. Pilih k data latihan dengan jarak terdekat dengan data uji.

3. Dalam klasifikasi, tentukan kelas mayoritas dari K tetangga terdekat sebagai prediksi kelas untuk data uji. Dalam regresi, hitung rata-rata nilai dari K tetangga terdekat sebagai prediksi nilai untuk data uji.

Rumus yang digunakan untuk menghitung jarak antara dua titik data (misalnya, titik data x dan y) dalam Euclidean distance adalah :

$$Distance(x,y)=\sum i=1n(xi-yi)2 \quad (8)$$

Keterangan :
- xi adalah nilai fitur ke-i dari data x.
- yi adalah nilai fitur ke-i dari data y.
- n adalah jumlah fitur dalam data.

Rumus yang digunakan untuk menghitung jarak dalam *Manhattan distance* (*Manhattan* atau *City Block distance*) adalah:

$$Distance(x,y)=\sum i=1n|xi-yi| \quad (9)$$

Linear Discriminant Analysis

Algoritma *Linear Discriminant Analysis* (LDA) merupakan metode pembelajaran mesin yang digunakan untuk tugas klasifikasi. LDA mencoba untuk menemukan kombinasi linear dari fitur-fitur yang memaksimalkan pemisahan antara kelas data. Dengan kata lain, LDA mencari proyeksi dari data ke sebuah ruang baru sehingga data dari kelas yang berbeda terpisah dengan jelas dan sejauh mungkin, sementara data dari kelas yang sama mendekati satu sama lain (Duda & Hart, 2006).

Langkah-langkah umum dalam algoritma Linear Discriminant Analysis adalah sebagai berikut :

1. Menghitung vektor rata-rata dari setiap kelas data.
2. Menghitung matriks scatter antar kelas (between-class scatter matrix) yang mengukur seberapa jauh rata-rata kelas berbeda satu sama lain.
3. Menghitung matriks scatter dalam kelas (within-class scatter matrix) yang mengukur seberapa jauh setiap data dari rata-rata kelasnya.
4. Menghitung vektor eigen dan nilai eigen dari invers matriks scatter dalam kelas.
5. Memilih vektor eigen dengan nilai eigen tertinggi yang sesuai dengan kelas data yang diinginkan.
6. Proyeksikan data ke ruang baru menggunakan vektor eigen yang dipilih.
7. Melakukan klasifikasi dengan metode k-NN atau metode lainnya pada data yang telah diproyeksikan ke ruang baru.

Rumus yang digunakan untuk menghitung matriks scatter antar kelas dan matriks scatter dalam kelas adalah:

1. Matriks Scatter Antar Kelas (Between-Class Scatter Matrix) :

$$SB=\sum i=1KNi(mi-m)(mi-m)T \quad (10)$$

Keterangan :
- K adalah jumlah kelas yang ada.
- Ni adalah jumlah data dalam kelas ke-i.
- mi adalah vektor rata-rata dari kelas ke-i.
- m adalah vektor rata-rata dari seluruh data.

2. Matriks Scatter Dalam Kelas (Within-Class Scatter Matrix) :

$$SW=\sum i=1K\sum x\in Ci(x-mi)(x-mi)T \quad (11)$$

Keterangan :
- x adalah vektor data dalam kelas ke-i.
- Ci adalah kelas ke-i.
- mi adalah vektor rata-rata dari kelas ke-i.

Metode Pengujian dan Pengolahan Data

Menurut (Christopher & Nasrabadi, 2006) *Confusion Matrix* (Matriks Konfusi) adalah alat evaluasi kinerja yang umum digunakan dalam tugas klasifikasi untuk mengukur seberapa baik model klasifikasi dapat memprediksi kelas data yang benar. Berisi informasi tentang jumlah prediksi yang benar dan salah yang dilakukan oleh model untuk setiap kelas. Matriks konfusi berisi empat istilah utama:

1. *True Positive (TP*): Jumlah data positif yang benar diprediksi sebagai positif oleh model.
2. *True Negative (TN)*: Jumlah data negatif yang benar diprediksi sebagai negatif oleh model.
3. *False Positive (FP):* Jumlah data negatif yang salah diprediksi sebagai positif oleh model (biasanya disebut juga dengan istilah "*Type I error*").
4. *False Negative (FN):* Jumlah data positif yang salah diprediksi sebagai negatif oleh model (biasanya disebut juga dengan istilah "*Type II error*").

Berikut adalah representasi umum dari *Confusion Matrix* :

Actual Positive (Actual 1) Actual Negative (Actual 0) Predicted Positive (Predicted 1) True Positive (TP) False Positive (FP) Predicted Negative (Predicted 0) False Negative (FN) True Negative (TN) Predicted Positive (Predicted 1) Predicted Negative (Predicted 0) Actual Positive (Actual 1) True Positive (TP) False Negative (FN) Actual Negative (Actual 0) False Positive (FP) True Negative (TN)

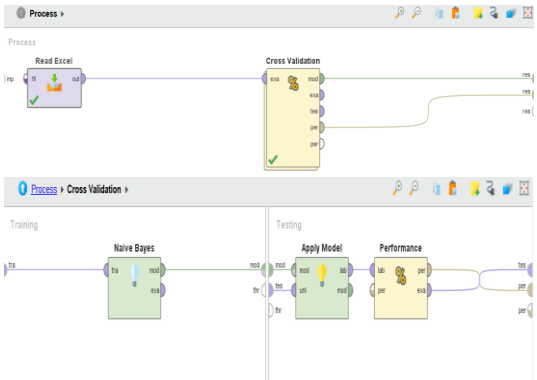
Rumus untuk menghitung beberapa metrik evaluasi klasifikasi berdasarkan matriks konfusi adalah sebagai berikut :

1. Akurasi (*Accuracy*)
   Akurasi=TP+TN+FP+FNTP+TN (12)
2. Presisi (*Precision*) atau *Positive Predictive Value* (PPV):
   Presisi=TP+FPTP (13)
3. Recall (Sensitivitas) atau *True Positive* Rate (TPR): *Recall*=TP+FNTP (14)
4. F1-Score : (15)
   F1Score=2×Presisi×RecallPresisi+Recall
   F1-score=Presisi+Recall2×Presisi×Recall

3. Hasil dan Pembahasan

Penelitian dilakukan dengan menguji dan menganalisa menggunakan aplikasi *rapid miner* terhadap enam algoritma yaitu *Naive Bayes, Neural Network, Decision Tree, Random Forest*, *K-Nearest Neighbors* (K-NN), *Linear Discriminant Analysis* maka diperoleh hasil sebagai berikut :

3.1 Pengujian perhitungan akurasi algoritma *Naïve Bayes*



Gambar 1. *Model Rapid Miner Naïve Bayes*

Berdasarkan gambar 1 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang belum pernah dilihat sebelumnya dengan menggunakan algoritma *Naïve Bayes*.

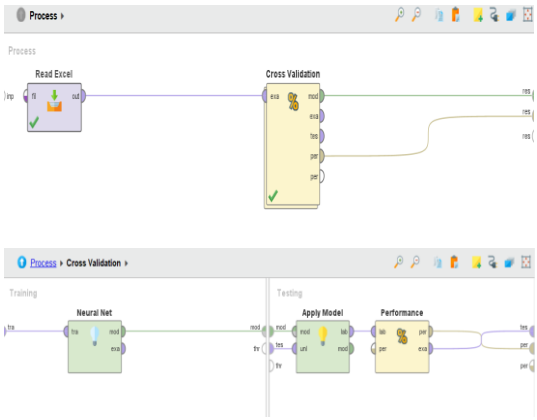
Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 1. Hasil Perhitungan Akurasi *Confussion Matrix Naïve Bayes*

Accuracy : 89,81% +/- 0,97% (micro average: 89,81%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HOROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1908	6	0	1	0	44	101	92.62%
pred. BARBUNYA	20	1083	0	117	2	15	4	97.27%
pred. BOMBAY	0	1	522	3	0	0	0	99.24%
pred. CALI	0	174	0	1475	35	10	0	87.07%
pred. HOROZ	1	7	0	27	1842	66	9	94.36%
pred. SIRA	79	51	0	7	38	2290	328	81.99%
pred. DERMOSAN	19	0	0	0	11	211	3104	92.80%
class recall	94.13%	81.92%	100.00%	90.49%	95.54%	86.87%	87.54%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *Naïve Bayes*, didapatkan hasil akurasi sebesar 89,81%.

3.2 Pengujian perhitungan akurasi algoritma *Neural Network*



Gambar 2. *Model Rapid Miner Neural Network*

Berdasarkan gambar 2 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang belum pernah dilihat sebelumnya dengan menggunakan algoritma *Neural Network*.

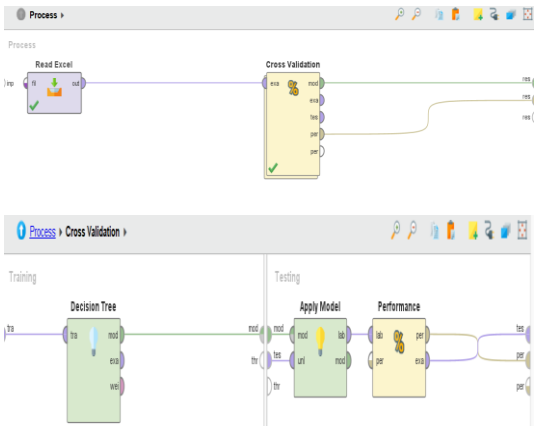
Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 2. Hasil Perhitungan Akurasi *Confussion Matrix Neural Network*

Accuracy : 92.68% +/- 0.72% (micro average: 92.68%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HORROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1926	13	0	2	0	31	51	95.21%
pred. BARBUNYA	10	1208	1	43	5	9	0	94.67%
pred. BOMBAY	0	0	519	0	0	0	0	100.00%
pred. CALI	0	73	2	1544	28	6	0	93.41%
pred. HORROZ	0	3	0	28	1835	33	5	96.38%
pred. SIRA	59	24	0	13	43	2342	250	85.76%
pred. DERMOSAN	32	1	0	0	17	215	3240	92.44%
class recall	95.02%	91.38%	99.43%	94.72%	95.18%	88.85%	91.37%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *Neural Network*, didapatkan hasil akurasi sebesar 92,68%.

3.3 Pengujian perhitungan akurasi algoritma *Decision Tree*



Gambar 3. *Model Rapid Miner Decision Tree*

Berdasarkan gambar 3 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang belum pernah dilihat sebelumnya dengan menggunakan algoritma *Decision Tree*.

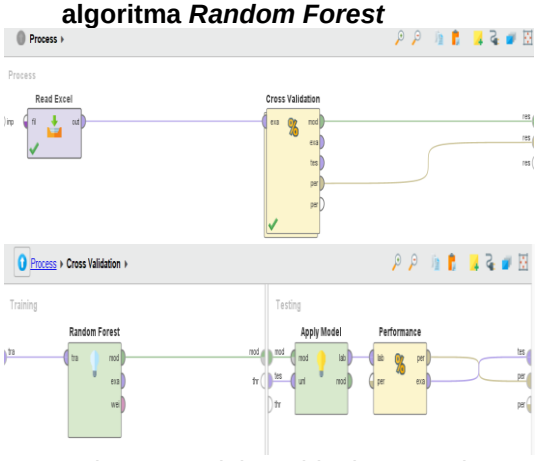
Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 3. Hasil Perhitungan Akurasi *Confussion Matrix Decision Tree*

Accuracy : 86.06% +/- 2.06% (micro average: 86.06%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HORROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1801	9	0	1	0	11	40	96.72%
pred. BARBUNYA	8	470	2	50	2	3	0	87.85%
pred. BOMBAY	0	1	520	0	0	0	0	99.81%
pred. CALI	0	761	0	1527	28	9	0	65.68%
pred. HORROZ	1	10	0	34	1814	37	2	95.57%
pred. SIRA	126	71	0	18	63	2261	184	83.03%
pred. DERMOSAN	91	0	0	0	21	315	3320	88.60%
class recall	88.85%	35.55%	99.62%	93.68%	94.09%	85.77%	93.63%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *Decision Tree*, didapatkan hasil akurasi sebesar 86,06%.

3.4 Pengujian perhitungan akurasi algoritma *Random Forest*



Gambar 4. *Model Rapid Miner Random Forest*

Berdasarkan gambar 4 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang

belum pernah dilihat sebelumnya dengan menggunakan algoritma *Random Forest*.

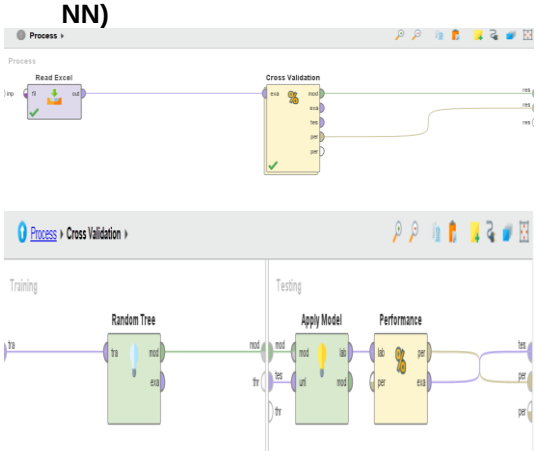
Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 4. Hasil Perhitungan Akurasi *Confussion Matrix Random Forest*

Accuracy : 88.80% +/- 0.88% (micro average: 88.80%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HOROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1812	10	0	2	0	14	42	96.38%
pred. BARBUNYA	5	892	1	80	3	3	0	90.65%
pred. BOMBAY	0	1	521	0	0	0	0	99.81%
pred. CALI	0	349	0	1506	35	9	0	79.30%
pred. HOROZ	1	5	0	25	1801	23	3	96.93%
pred. SIRA	125	65	0	17	63	2202	149	84.01%
pred. DERMOSAN	84	0	0	0	26	385	3352	87.13%
class recall	89.39%	67.47%	99.81%	92.39%	93.41%	83.54%	94.53%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *Random Forest*, didapatkan hasil akurasi sebesar 88,80%.

3.5 Pengujian perhitungan akurasi algoritma *K-Nearest Neighbors* (K-NN)



Gambar 5. *Model Rapid Miner K-Nearest Neighbors (K-NN)*

Berdasarkan gambar 5 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang belum pernah dilihat sebelumnya dengan menggunakan algoritma *K-Nearest Neighbors (*K-NN).

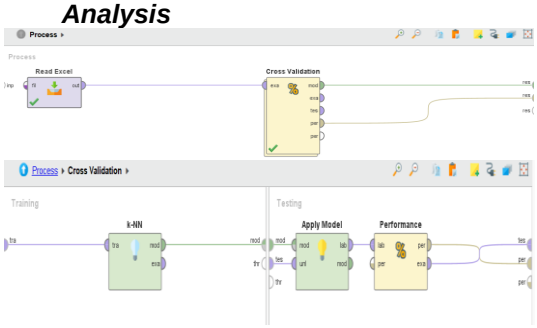
Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 5. Hasil Perhitungan Akurasi *Confussion Matrix K-Nearest Neighbors (*K-NN)

Accuracy : 78.32% +/- 0.77% (micro average: 73.82%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HOROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1303	0	0	1	4	111	183	81.34%
pred. BARBUNYA	2	623	0	432	125	9	0	52.31%
pred. BOMBAY	0	0	520	1	0	0	0	99.81%
pred. CALI	1	486	0	1085	87	1	0	65.28%
pred. HOROZ	23	169	0	105	1345	185	4	73.46%
pred. SIRA	386	43	0	6	315	2038	226	67.62%
pred. DERMOSAN	312	1	0	0	52	292	3133	82.66%
class recall	64.28%	47.13%	99.62%	66.56%	69.76%	77.31%	88.35%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *K-Nearest Neighbors (*K-NN), didapatkan hasil akurasi sebesar 78,32%.

3.6 Pengujian perhitungan akurasi algoritma *Linear Discriminant Analysis*



Gambar 6. *Model Rapid Miner Linear Discriminant Analysis*

Berdasarkan gambar 6 proses yang terjadi dimulai dari membaca dataset berbentuk *excel* kemudian diukur dan di

validasi (*cross validation)* seberapa baik model akan berkinerja pada data yang belum pernah dilihat sebelumnya dengan menggunakan algoritma *Linear Discriminant Analysis*.

Algoritma tersebut menghasilkan penilaian berdasarkan pola yang telah diidentifikasi selama proses pelatihan model, sehingga muncul perhitungan performa dari model yang di gunakan, kinerja atau kemampuan model atau sistem dalam menghasilkan hasil yang diharapkan atau akurat.

Tabel 6. Hasil Perhitungan Akurasi *Confussion Matrix Linear Discriminant Analysis*

Accuracy : 87.42% +/- 0.98% (micro average: 87.42%)								
	true SEKER	true BARBUNYA	true BOMBAY	true CALI	true HOROZ	true SIRA	true DERMOSAN	class precision
pred. SEKER	1806	13	0	2	0	5	28	97.41%
pred. BARBUNYA	19	1035	5	5	12	7	6	95.04%
pred. BOMBAY	0	0	501	0	0	0	0	100.00%
pred. CALI	0	85	16	1466	67	1	0	89.66%
pred. HOROZ	0	2	0	11	1712	5	2	98.85%
pred. SIRA	173	187	0	146	124	2550	681	66.05%
pred. DERMOSAN	29	0	0	0	13	68	2829	96.26%
class recall	89.10%	78.29%	95.96%	89.94%	88.80%	96.74%	79.78%	

Hasil pengujian dataset *Dry Bean* menggunakan *Rapid Miner (Cross Validation*) dan perhitungan *Confussion Matrix* algoritma *Linear Discriminant Analysis*, didapatkan hasil akurasi sebesar 87,42%.

4. Kesimpulan

Penelitian yang dilakukan menggunakan dataset *dry bean* yang diperoleh dari UCI *Machine Learning Repository* dengan membandingkan enam algoritma klasifikasi. Hasil akurasi yang diperoleh dari masing-masing algoritma yaitu *Naive Bayes* sebesar 89,91%*, Neural Network* sebesar 92,68%*, Decision Tree* sebesar 86,06%*, Random Forest* sebesar 88,80%, *K-Nearest Neighbors (*K-NN) sebesar 78,32%, dan *Linear Discriminant Analysis* sebesar 87,42%.

Menggunakan model *rapid miner* dan *confussion matrix* diperoleh hasil bahwa nilai akurasi performa terbaik dan terbesar yaitu algoritma *Neural Network* sebesar 92,68% dan nilai akurasi performa terendah adalah algoritma *K-Nearest Neighbors (K-NN)* dengan nilai 78,32%.

Referensi

Ardiyansyah, Rahayuningsih, P. A., & Maulana, R. (2018). Analisis Perbandingan Algoritma Klasifikasi Data Mining Untuk Dataset Blogger Dengan Rapid Miner. *Jurnal Khatulistiwa Informatika*, *VI*(1), 20–28.

Arikunto, S. (2006). *Prosedur Penelitian Suatu Pendekatan Praktek*. PT Rineka Cipta.

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.Ian Goodfellow, Yoshua Bengio, dan A. C. (2016). *Deep Learning*.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Duda, R. O., & Hart, P. E. (2006). Pattern classification. John Wiley & Sons.

Kaunang, F. J., Rotikan, R., & Tulung, G. S. (2018). Pemodelan Sistem Prediksi Tanaman Pangan Menggunakan Algoritma Decision Tree. *Cogito Smart Journal*, 4(1), 213-218.

Lestari, I., Akbar, M., & Intan, B. (2023). Perbandingan Algoritma Machine Learning Untuk klasifikasi Amenorrhea. *Journal of Computer and Information Systems Ampera*, *4*(1), 32–43. https://doi.org/10.51519/journalcisa.v4i1.371

Mehta, A., Sengupta, P., Garg, D., Singh, H., & Diamand, Y. S. (2023). *Benchmarking the Effectiveness of Classification Algorithms and SVM Kernels for Dry Beans*. http://arxiv.org/abs/2307.07863

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Setiadi, T., Noviyanto, F., Hardianto, H., Tarmuji, A., Fadlil, A., & Wibowo, M. (2020). Implementation of *Naïve Bayes* method in food crops planting recommendation. *Int. J. Sci. Technol. Res*, 9(02), 4750-4755.

Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

Widiastuti, D. (2012). Analisa Perbandingan Algoritma Svm, Naive Bayes, Dan Decision Tree Dalam Mengklasifikasikan Serangan (Attacks) Pada Sistem Pendeteksi Intrusi.