

**Machine Learning untuk Pengendalian Mutu: Studi Kasus Prediksi
Kecacatan Cetakan Catur dengan Algoritma Random Forest**

**Disusun guna memenuhi tugas mata kuliah
Kecerdasan Buatan**

Dosen Pengampu:

Ronal, M.Kom.

Rizki Yustisia Sari, S.T., M.T.



Disusun Oleh:

Tedy Saputra Anugrah Akbar

NIM : 122490032

**PROGRAM STUDI REKAYASA INSTRUMENTASI DAN AUTOMASI
FAKULTAS TEKNOLOGI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan makalah ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Ucapan terima kasih kami sampaikan kepada bapak Ronal sebagai dosen pengampu mata kuliah Kecerdasan Buatan yang telah membantu memberikan arahan dan pemahaman dalam penyusunan makalah ini.

Kami menyadari bahwa dalam penyusunan makalah ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan makalah ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Lampung Selatan, 20 Mei 2025

Tedy Saputra Anugrah Akbar

DAFTAR ISI

BAB I PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	5
1.3 Tujuan Penulisan.....	5
BAB II PEMBAHASAN.....	7
2.1 Perancangan Sistem	7
2.1.1. Random Forest	9
2.2 Tahapan Preprocessing Data untuk Meningkatkan Akurasi Klasifikasi .	13
2.2.1. Data Cleaning.....	13
2.2.2. Transformasi Data Kategorikal	13
2.2.3. Normalisasi atau Standarisasi Data.....	15
2.3 Evaluasi Performa Model Random Forest	15
2.4 Implementasi dan Uji Coba Sistem.....	19
BAB III PENUTUP	23
3.1 Kesimpulan	23
3.2 Saran	24

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) telah membawa dampak signifikan dalam berbagai bidang industri, termasuk dalam bidang manufaktur dan sistem otomasi. Salah satu penerapan yang cukup menjanjikan adalah pemanfaatan AI untuk mendeteksi kualitas produk secara otomatis dalam lini produksi. Dalam konteks industri cetakan plastik, kemampuan sistem untuk mendeteksi cacat produk sejak dini sangat krusial agar proses produksi tetap efisien dan hasil produksi tetap memenuhi standar kualitas yang diharapkan.

Proses pencetakan biji catur, meskipun terlihat sederhana, tetap membutuhkan pengawasan kualitas yang ketat. Produk cacat dapat disebabkan oleh berbagai faktor seperti suhu pemanasan yang tidak stabil, tekanan yang tidak sesuai, atau posisi cetakan yang miring. Jika pengawasan kualitas masih dilakukan secara manual, maka terdapat risiko human error, keterlambatan deteksi, serta tidak konsistennya hasil penilaian. Oleh karena itu, dibutuhkan sistem yang mampu mengidentifikasi kecacatan produk secara otomatis berdasarkan parameter proses yang tersedia.

Dalam implementasinya, sistem pencetakan biji catur ini dirancang untuk mendeteksi kualitas hasil cetakan berdasarkan tiga kategori, yaitu “Cacat Berat”, “Cacat Ringan”, dan “Bagus”, dengan memanfaatkan data dari sensor suhu, tekanan, dan kemiringan. Algoritma Random Forest digunakan sebagai metode klasifikasi karena memiliki keunggulan dalam menangani data yang kompleks dan heterogen, serta tahan terhadap overfitting. Random Forest merupakan algoritma ansambel yang terdiri atas banyak decision tree yang dilatih dengan subset data yang berbeda (bagging), sehingga memberikan prediksi yang lebih stabil dan akurat.

Sebelum dilakukan proses klasifikasi, data yang diperoleh dari hasil produksi akan melalui tahap preprocessing untuk memastikan kualitas data tersebut. Tahapan preprocessing mencakup pembersihan data (missing value dan outlier), transformasi

data kategorikal menjadi numerik, serta normalisasi atau standarisasi nilai numerik. Proses ini penting agar algoritma AI dapat bekerja secara optimal dan tidak bias terhadap fitur tertentu yang memiliki rentang nilai lebih besar.

Sistem ini juga dilengkapi dengan fitur evaluasi performa model menggunakan metode validasi silang (cross-validation) dan metrik evaluasi seperti akurasi, F1-score, dan confusion matrix. Dengan adanya sistem ini, diharapkan proses quality control dalam industri pencetakan biji catur dapat berjalan secara otomatis, efisien, dan konsisten.

Lebih lanjut, sistem ini tidak hanya menghasilkan output klasifikasi, tetapi juga dapat diekspor dalam bentuk file (misalnya Excel) dan divisualisasikan dalam bentuk grafik atau matriks, sehingga memudahkan proses analisis lanjutan. Selain itu, pengguna juga diberikan fasilitas untuk melakukan "Tes Prediksi", yaitu memasukkan parameter secara manual untuk melihat hasil klasifikasi tanpa harus melalui proses produksi secara langsung.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem klasifikasi kualitas hasil cetakan biji catur menggunakan algoritma Random Forest berbasis data sensor suhu, tekanan, dan kemiringan?
2. Bagaimana tahapan preprocessing data dilakukan untuk meningkatkan akurasi klasifikasi pada model Random Forest?
3. Bagaimana cara mengevaluasi performa model klasifikasi yang dihasilkan agar dapat digunakan secara efektif dalam proses pengendalian kualitas produksi biji catur?

1.3 Tujuan Penulisan

Tujuan dari penulisan makalah ini adalah:

1. Untuk mengetahui tentang perancangan sistem klasifikasi kualitas hasil cetakan biji catur menggunakan algoritma Random Forest berbasis data sensor suhu, tekanan, dan kemiringan.
2. Untuk memahami tentang tahapan preprocessing data seperti pembersihan data (data cleaning), normalisasi, dan encoding yang diperlukan sebelum data digunakan dalam proses pelatihan model klasifikasi.
3. Untuk mengetahui tentang metode evaluasi performa model klasifikasi, seperti akurasi, F1-score, dan confusion matrix, serta pemanfaatannya dalam pengambilan keputusan terhadap kualitas produk hasil cetakan biji catur.

BAB II PEMBAHASAN

2.1 Perancangan Sistem

Proses manufaktur biji catur berbahan plastik atau resin umumnya melibatkan tahapan pemanasan material, pengepresan atau pencetakan ke dalam cetakan (mold), dan pendinginan hingga produk mengeras. Selama proses ini, kualitas produk sangat dipengaruhi oleh parameter fisis seperti suhu pemanasan, tekanan saat pengepresan, dan posisi cetakan. Ketidaksesuaian nilai-nilai parameter tersebut dapat menghasilkan produk cacat, seperti permukaan kasar, bentuk tidak simetris, atau detail cetakan yang tidak tercetak dengan baik.

Untuk mengatasi permasalahan tersebut, sistem klasifikasi kualitas berbasis sensor dan pembelajaran mesin dikembangkan. Sistem ini memungkinkan proses pengambilan keputusan yang lebih akurat dan cepat dalam mengidentifikasi produk cacat secara otomatis.

Pada sistem yang dikembangkan, digunakan mikrokontroler ESP32 sebagai pusat pengendali untuk membaca data dari beberapa sensor, yaitu:

- Sensor suhu (DS18B20): untuk mengukur suhu cetakan selama proses pemanasan.
- Sensor tekanan (MPX5700AP): untuk mendeteksi tekanan dalam sistem pneumatic saat proses pengepresan.
- Sensor kemiringan (MPU6050): untuk memastikan posisi cetakan tetap stabil dan tidak miring selama proses cetak berlangsung.

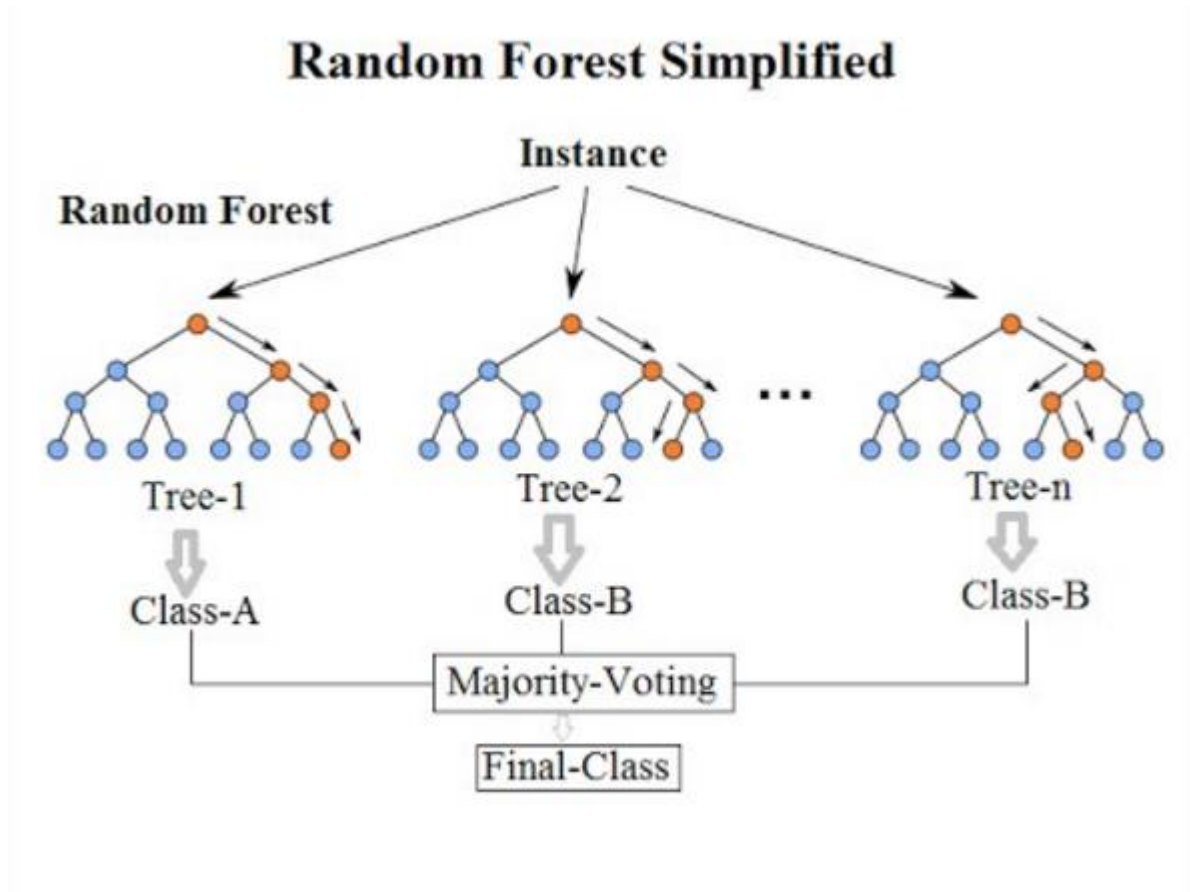
Data yang dikumpulkan dalam data Dummy tersebut kemudian dikirim ke Sistem python dan diproses dalam algoritma Random Forest. Data historis ini selanjutnya digunakan untuk melatih model klasifikasi berbasis Random Forest guna

membedakan antara produk yang memenuhi standar kualitas (baik) dan yang tidak (cacat).

Model klasifikasi ini dikembangkan dalam platform Python dan diintegrasikan ke dalam antarmuka GUI menggunakan PyQt5, sehingga pengguna dapat memantau hasil klasifikasi secara langsung. Selain itu, sistem juga menyediakan fitur ekspor data ke Excel dan visualisasi pohon keputusan, yang memudahkan pengguna dalam menganalisis pola dan penyebab produk cacat.

Penerapan sistem ini diharapkan dapat meningkatkan efisiensi produksi dengan cara mengurangi keterlibatan manusia dalam proses inspeksi manual serta mempercepat proses deteksi produk gagal. Sistem ini juga dapat dikembangkan lebih lanjut untuk digunakan dalam berbagai proses manufaktur lain yang memiliki karakteristik serupa.

2.1.1. Random Forest



Gambar 1. Random Forest

Random Forest adalah algoritma machine learning yang kuat dan populer untuk tugas klasifikasi dan regresi. Algoritma ini merupakan pengembangan dari metode decision tree yang menggabungkan beberapa pohon keputusan untuk mencapai hasil yang lebih akurat dan stabil.

Random Forest juga merupakan teknik ensemble learning yang menciptakan "hutan" dari banyak decision tree dan menggabungkan hasil prediksi mereka. Konsep utama di balik Random Forest adalah:

- Ensemble Learning: Menggabungkan banyak model sederhana (pohon keputusan) untuk mendapatkan model yang lebih baik
- Bagging (Bootstrap Aggregating): Mengambil sampel acak dengan pengembalian dari data pelatihan
- Random Feature Selection: Memilih subset acak dari fitur pada setiap split node

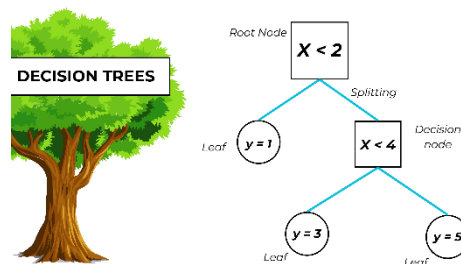
1) Cara Kerja Random Forest:

a. . Pembentukan Pohon (Tree Building)

Algoritma membuat sejumlah n pohon keputusan (biasanya ratusan hingga ribuan)

Untuk setiap pohon:

- Mengambil sampel bootstrap dari data pelatihan (sampling acak dengan pengembalian).
- Pada setiap node split, hanya subset acak dari fitur yang dipertimbangkan (biasanya \sqrt{p} fitur, di mana p adalah jumlah total fitur).
- Pohon tumbuh tanpa pruning (pemangkasan) hingga mencapai kedalaman maksimum atau sampai setiap node leaf berisi sampel minimum tertentu.



Gambar 2. Decision Tree

b. Prediksi (Inference)

- Untuk klasifikasi: Setiap pohon memberikan satu "suara" untuk kelas prediksi, dan kelas dengan suara terbanyak menjadi hasil akhir

- Untuk regresi: Output akhir adalah rata-rata prediksi dari semua pohon.

2) Komponen Matematis:

a. Sampling Bootstrap

Jika dataset pelatihan memiliki N sampel, algoritma mengambil N sampel secara acak dengan pengembalian

b. Kriteria Split

Biasanya menggunakan:

- Gini impurity atau Information Gain (Entropy) untuk klasifikasi
Gini Impurity adalah ukuran ketidakaturan atau impurity dari suatu himpunan data. Makin rendah nilai Gini, makin homogen (seragam) datanya.

Rumus Gini:

$$\text{Gini} = 1 - \sum_{i=1}^n p_i^2$$

di mana:

- p_i = proporsi data pada kelas ke-i
- n = jumlah total kelas

Entropy mengukur ketidakpastian (chaos) dari suatu himpunan data. Information Gain adalah pengurangan ketidakpastian setelah data di-split berdasarkan fitur tertentu.

Rumus Entropy:

$$\text{Entropy} = - \sum_{i=1}^n p_i \log_2(p_i)$$

Rumus Information Gain:

$$\text{Information Gain} = \text{Entropy}(S) - \sum_{j=1}^k \frac{|S_j|}{|S|} \cdot \text{Entropy}(S_j)$$

di mana:

- S = himpunan data sebelum split
- S_j = himpunan data hasil split ke- j

- Mean Squared Error (MSE) untuk regresi

MSE digunakan dalam regresi, bukan klasifikasi. Ia mengukur rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi.

Rumus MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

di mana:

- y_i = nilai aktual
- \hat{y}_i = nilai prediksi
- n = jumlah sampel

c. Agregasi Hasil

- Majority voting untuk klasifikasi: kelas final = mode(hasil dari semua pohon)
- Averaging untuk regresi: nilai final = mean(hasil dari semua pohon)

d. Parameter Penting dalam Random Forest

- `n_estimators`: Jumlah pohon dalam forest
- `max_features`: Jumlah fitur yang dipertimbangkan pada setiap split
- `max_depth`: Kedalaman maksimum pohon
- `min_samples_split`: Jumlah minimum sampel yang diperlukan untuk split node
- `min_samples_leaf`: Jumlah minimum sampel di leaf node

2.2 Tahapan Preprocessing Data untuk Meningkatkan Akurasi Klasifikasi

Preprocessing data merupakan langkah krusial sebelum data dimasukkan ke algoritma machine learning, termasuk Random Forest. Berikut penjelasan tahapan-tahapan preprocessing data yang dapat meningkatkan akurasi klasifikasi:

2.2.1. Data Cleaning

Data cleaning adalah proses menangani masalah kualitas data yang dapat memengaruhi performa model.

a. Penanganan nilai kosong (missing values):

- Nilai kosong dapat menyebabkan bias atau error pada model
- Dapat ditangani dengan menghapus baris yang mengandung nilai kosong, mengisi dengan nilai mean/median/modus, atau teknik imputasi lanjutan seperti KNN imputation

b. Penanganan duplikasi data:

- Data duplikat dapat memberikan bobot berlebih pada pola tertentu
- Solusinya adalah identifikasi dan hapus data duplikat untuk mencegah bias

Contoh: Menghapus entri pasien yang terdaftar dua kali dengan informasi identik

c. Penanganan outlier:

- Outlier dapat mendistorsi distribusi data dan memengaruhi pembelajaran model
- Dapat ditangani dengan teknik deteksi outlier seperti IQR (Interquartile Range), Z-score, atau DBSCAN

2.2.2. Transformasi Data Kategorikal

Data kategorikal perlu dikonversi menjadi format numerik agar dapat diproses oleh algoritma ML:

a. One-hot encoding:

- Mengubah kategori menjadi kolom-kolom binary (0/1)
- Cocok untuk kategori nominal tanpa hierarki
- Contoh: Variabel 'warna' (merah, hijau, biru) menjadi tiga kolom binary

id	color	One Hot Encoding		
id	color	color_red	color_blue	color_green
1	red	1	0	0
2	blue	0	1	0
3	green	0	0	1
4	blue	0	1	0

Gambar 3. One-hot Encoding

b. Label encoding:

- Mengubah kategori menjadi nilai numerik berurutan (0, 1, 2, ...)
- Cocok untuk kategori ordinal dengan urutan yang jelas
- Contoh: Variabel 'tingkat bagus cetakan' (Bagus, Cacat Ringan, Cacat Berat) menjadi (1, 2, 3)

color	color
red	0
green	1
blue	2
red	0

Gambar 4. Label encoding

2.2.3. Normalisasi atau Standarisasi Data

Penskalaan data numerik penting untuk memastikan fitur-fitur memiliki kontribusi seimbang:

a. Standarisasi (StandardScaler):

- Mentransformasi data sehingga memiliki mean 0 dan standar deviasi 1
- Berguna untuk algoritma yang mengasumsikan distribusi normal
- Formula: $z = (x - \mu) / \sigma$

b. Normalisasi (MinMaxScaler):

- Menskalakan data ke rentang tertentu, biasanya [0,1]
- Berguna untuk algoritma yang sensitif terhadap skala data
- Formula: $x_scaled = (x - x_min) / (x_max - x_min)$

2.3 Evaluasi Performa Model Random Forest

2.3.1. Interpretasi Confusion Matrix

Confusion matrix memberikan gambaran detail tentang performa model:

- Diagonal utama (TP dan TN): Menunjukkan prediksi yang benar
- Off-diagonal (FP dan FN): Menunjukkan kesalahan model

Analisis pola kesalahan:

- Jika FP tinggi: Model terlalu "agresif" dalam memprediksi kelas positif
- Jika FN tinggi: Model terlalu "konservatif" dan melewatkan kasus positif
- Jika kesalahan terkonsentrasi di kelas tertentu: Mungkin perlu lebih banyak data untuk kelas tersebut

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 5. Confusion Matrix

2.3.2. Interpretasi Akurasi

- Akurasi tinggi (>90%): Pada dataset seimbang, ini umumnya menunjukkan model yang baik
- Akurasi moderat (70-90%): Performa yang wajar untuk banyak masalah dunia nyata
- Akurasi rendah (<70%): Mungkin ada masalah dengan model atau fitur

2.3.3. Interpretasi Precision dan Recall

- Precision tinggi, recall rendah: Model konservatif, hanya membuat prediksi positif ketika sangat yakin
- Precision rendah, recall tinggi: Model agresif, menangkap hampir semua kasus positif tetapi dengan banyak false positive
- Keseimbangan precision-recall: Tergantung pada domain dan biaya kesalahan

		Nilai sebenarnya	
		TRUE	FALSE
Nilai prediksi	TRUE	TP (True Positive) <i>Correct result</i>	FP (False Positive) <i>Unexpected result</i>
	FALSE	FN (False Negative) <i>Missing result</i>	TN (True Negative) <i>Correct absence of result</i>

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

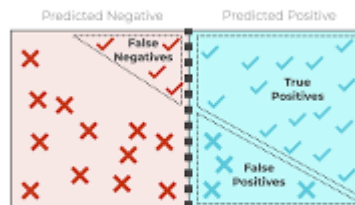
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 6. Precision and recall

2.3.4. Interpretasi F1-Score

- F1-Score tinggi: Menunjukkan keseimbangan yang baik antara precision dan recall
- F1-Score rendah: Mengindikasikan trade-off yang buruk; salah satu atau kedua metrik tersebut rendah

F1 SCORE IS THE HARMONIC MEAN OF PRECISION AND RECALL



$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

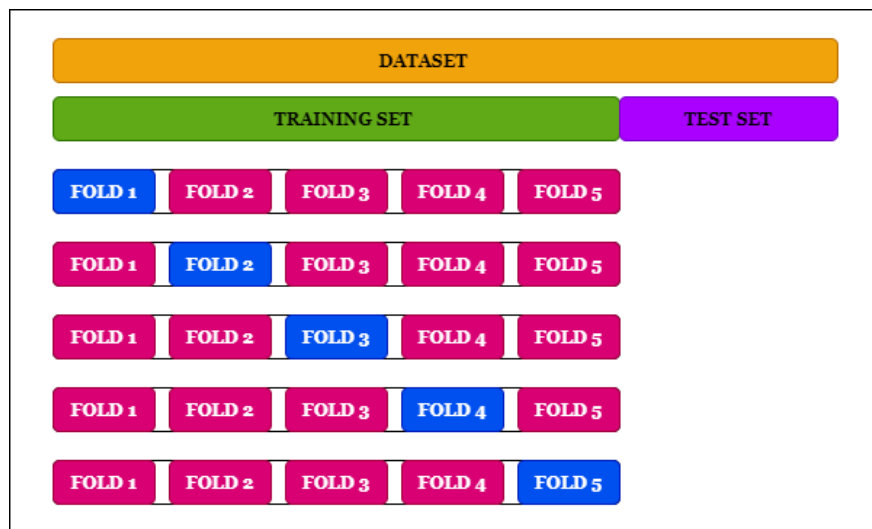
Gambar 7. F1-Score

2.3.5. Interpretasi Cross-Validation

K-Fold Cross-Validation adalah teknik evaluasi model yang membagi data menjadi k subset atau *fold* yang ukurannya kurang lebih sama. Proses pelatihan dan pengujian dilakukan sebanyak k kali, di mana pada setiap iterasi, satu fold digunakan sebagai data

validasi dan $k-1$ fold sisanya digunakan sebagai data pelatihan. Setelah semua iterasi selesai, hasil evaluasi dari setiap fold dirata-rata untuk memperoleh estimasi performa model secara keseluruhan. Teknik ini membantu mengurangi risiko overfitting dan memberikan gambaran yang lebih stabil terhadap kinerja model dibandingkan dengan pembagian data tunggal. Umumnya, nilai k yang digunakan adalah 5 atau 10, karena memberikan keseimbangan yang baik antara akurasi evaluasi dan efisiensi komputasi.

Sementara itu, Leave-One-Out Cross-Validation (LOOCV) merupakan bentuk ekstrem dari K-Fold Cross-Validation, di mana jumlah fold sama dengan jumlah total data dalam dataset. Artinya, pada setiap iterasi, hanya satu sampel yang digunakan sebagai data validasi, sedangkan sisanya digunakan untuk pelatihan. Teknik ini memungkinkan penggunaan maksimum dari data untuk pelatihan, menjadikannya sangat ideal untuk dataset yang kecil. Namun, karena prosesnya harus dilakukan sebanyak jumlah sampel yang ada, LOOCV menjadi sangat intensif secara komputasi dan kurang efisien bila diterapkan pada dataset besar.



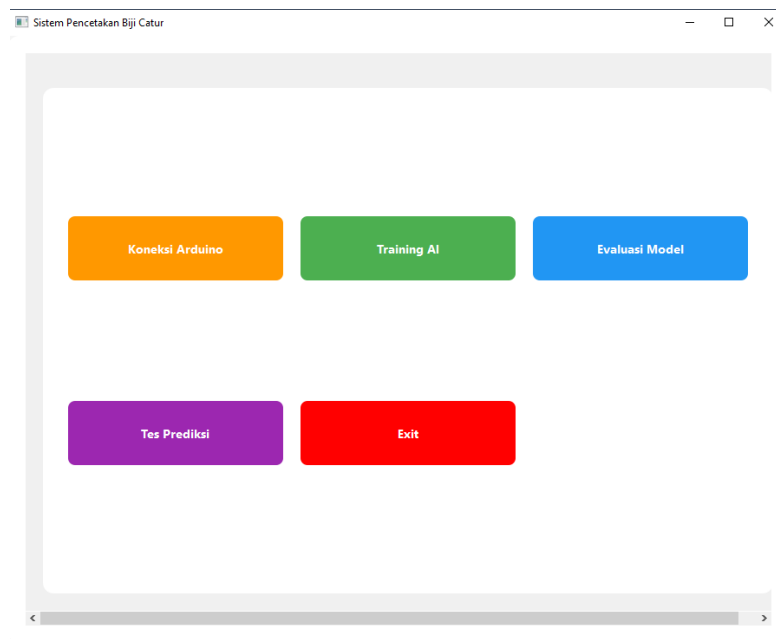
Gambar 8. Cross-Validation

2.4 Implementasi dan Uji Coba Sistem

Pengujian sistem dilakukan dengan menggunakan Bahasa pemrograman python dan untuk data menggunakan data dummy:

2.4.1. Tampilan Menu Dashboard

Pada bagian ini terdapat tombol koneksi Arduino untuk monitoring data secara real time, menu training AI untuk membuat model ai yang di training menggunakan data dummy yang telah di siapkan, terdapat menu evaluasi model untuk mengecek apakah model yang ditraining algoritma atau model AI nya bagus, terdapat juga menu tes prediksi untuk mencoba prediksi dari model Ai yang di buat dan terakhir ada menu Exit untuk keluar dari Aplikasi.



Gambar 9.Dashboard

2.4.2. Pengujian dengan Data Uji (Test Set)

a. Pembagian Data

- Data dibagi menjadi set pelatihan (training) dan validasi
- Proporsi umum: 80% training dan 20% validasi
- Teknik: Random splitting dengan stratified sampling untuk mempertahankan distribusi kelas

b. Prosedur Pengujian

- Model dilatih dimenu training AI
- Hyperparameter dioptimasi menggunakan set validasi (5-Fold,10-Fold dan Leave on out)

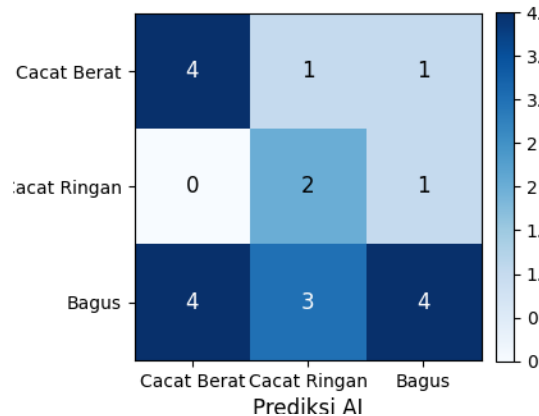
The screenshot shows a web application window titled "Sistem Pencetakan Biji Catur". Inside, there is a section titled "Training Model AI". This section includes a "File Data:" label followed by a text input field containing a file path and a blue "Browse" button. Below this is a "Nama Model:" label followed by a text input field containing "data_100". Underneath is a "Validasi:" label followed by a dropdown menu currently showing "5-fold". The dropdown menu is open, displaying three options: "5-fold", "10-fold", and "Leave-One-Out". Below the dropdown is a large green button labeled "Train Model". At the bottom of the window is a blue button labeled "Kembali ke Menu Utama".

Gambar 10. Training Model

- Model final dievaluasi satu kali pada set pengujian
- Metrik performa dihitung (akurasi, precision, recall, F1-score)

2.4.3. Hasil Pengujian Dan Tes Prediksi

a. Pengujian pada 100 data



Gambar 11. Hasil Confusion Matrix model AI

Dilihat Pada Confusion matrik dalam rentan 100 data masih terdapat prediksi yang salah contoh yang dimana label bagus tetapi prediksi IA cacat berat.

Laporan Klasifikasi:				
	precision	recall	f1-score	support
Cacat Berat	0.50	0.67	0.57	6
Cacat Ringan	0.33	0.67	0.44	3
Bagus	0.67	0.36	0.47	11
accuracy			0.50	20
macro avg	0.50	0.57	0.50	20
weighted avg	0.57	0.50	0.50	20

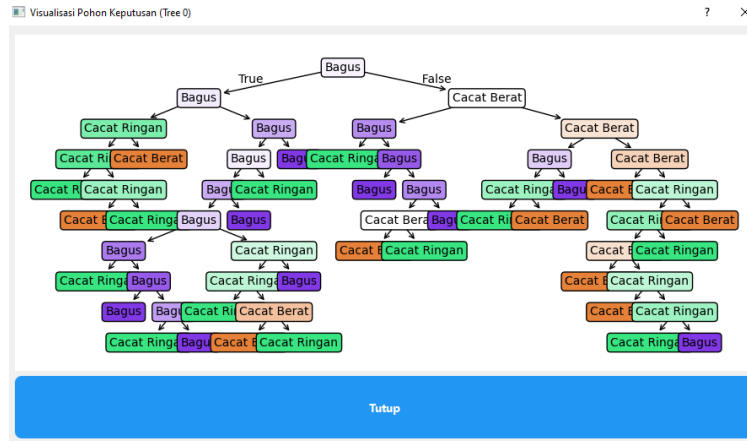
Kesimpulan:
Model AI ini masih perlu perbaikan karena performanya belum optimal.

Gambar 12. Kesimpulan Model AI

Dapat disimpulkan Bahwa model ai yang dibuat masih kurang dan perlu perbaikan dalam algoritmanya ataupun dari banyak data yang ditraining.

b. Fitur Tambahan Sistem

- Ekspor ke Excel data dari training evaluasi dapat di export ke format xlsx
- Visualisasi Pohon Keputusan dari model training yang di evaluasi pohon keputusannya bisa dilihat dari 0-99 pohon.



Gambar 13. Decision Tree

c. Prediksi Manual

Pada fitur ini pengguna memasukkan suhu, pressure dan kemiringan yang nantinya akan di prediksi dengan data yang dimasukan cetakan dalam kondisi bagus, cacat ringan atau cacat berat berdasarkan tingkat persentase model AI.

Tes Prediksi Model AI

Pilih Model:

Suhu:

Pressure:

Kemiringan:

Hasil Prediksi:
 Kelas: Cacat
 Tingkat Kepercayaan: 16.00%

Gambar 14. Menu Prediksi

BAB III PENUTUP

3.1 Kesimpulan

Makalah ini membahas penerapan algoritma Random Forest dalam membangun sistem klasifikasi kecacatan hasil cetakan biji catur berdasarkan parameter suhu, tekanan, dan kemiringan. Dari pembahasan yang telah dilakukan, dapat disimpulkan bahwa pemanfaatan algoritma machine learning seperti Random Forest terbukti efektif dalam membantu proses pengendalian mutu secara otomatis di industri pencetakan plastik.

Proses klasifikasi kualitas hasil cetakan ke dalam tiga kategori — “Bagus”, “Cacat Ringan”, dan “Cacat Berat” — dapat dilakukan dengan tingkat akurasi yang tinggi ketika data telah diproses melalui tahapan preprocessing seperti pembersihan data, normalisasi, dan encoding. Tahapan ini sangat penting untuk memastikan bahwa data yang dimasukkan ke dalam model memiliki kualitas yang sesuai dan tidak bias terhadap fitur tertentu.

Keunggulan Random Forest dalam mengatasi data kompleks dan menghindari overfitting menjadikannya pilihan yang tepat untuk studi ini. Selain menghasilkan model yang stabil dan akurat, sistem juga memungkinkan pengguna untuk melakukan prediksi secara manual dan menampilkan hasil evaluasi performa model dalam bentuk visualisasi yang mudah dipahami, seperti confusion matrix dan grafik akurasi.

Dengan sistem ini, industri dapat memperoleh proses quality control yang lebih efisien, konsisten, dan minim kesalahan manusia. Implementasi AI dalam proses produksi tidak hanya meningkatkan kecepatan dan akurasi, tetapi juga memungkinkan pengambilan keputusan yang lebih berbasis data.

3.2 Saran

Dalam pengembangan sistem lebih lanjut, disarankan untuk mengeksplorasi algoritma machine learning lainnya seperti Gradient Boosting atau Neural Network untuk dibandingkan performanya dengan Random Forest. Selain itu, integrasi sistem ini ke dalam perangkat keras industri secara langsung juga perlu diuji untuk mengetahui keandalannya dalam kondisi nyata.

Peningkatan kualitas dan jumlah data pelatihan akan sangat berpengaruh terhadap hasil klasifikasi, sehingga perlu dilakukan pengumpulan data secara berkelanjutan. Terakhir, pelatihan kepada operator produksi terkait penggunaan sistem ini penting dilakukan agar pemanfaatan teknologi dapat berlangsung secara optimal dan berkelanjutan.

DAFTAR PUSTAKA

Aggarwal, C. C. (2021). Machine learning for data streams: with practical examples in MOA (2nd ed.). Springer.

Brownlee, J. (2020). Machine Learning Mastery With Random Forests in Python. Machine Learning Mastery.

Chollet, F., & Allaire, J. J. (2021). Deep learning with R (2nd ed.). Manning Publications.

Dua, D., & Graff, C. (2020). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Retrieved from <https://archive.ics.uci.edu/ml>

Geron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.

Kumar, V., & Kaur, A. (2020). A Review on Data Preprocessing Techniques in Machine Learning. International Journal of Scientific & Technology Research, 9(1), 20–25.

Müller, A. C., & Guido, S. (2021). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.

Raschka, S., & Mirjalili, V. (2022). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 (3rd ed.). Packt Publishing.

Sarkar, D., Bali, R., & Sharma, T. (2020). Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems. Apress.

Zhao, Y., Wu, Z., & Wang, C. (2021). Application of Artificial Intelligence in Quality Control of Industrial Production: A Review. IEEE Access, 9, 62454–62465. <https://doi.org/10.1109/ACCESS.2021.3073789>