

# A Comparative Study of Structural Improvement and Hyperparameter Optimization for Enhancing Naïve Bayes Performance

Jejen Jaenudin <sup>1\*</sup>, Novita BR Ginting <sup>2\*</sup>, Yusup Maolani <sup>3\*</sup>

\* Sistem Informasi, Universitas Ibn Khaldun Bogor

[zen@uika-bogor.ac.id](mailto:zen@uika-bogor.ac.id) <sup>1</sup>, [novitawahab@uika-bogor.ac.id](mailto:novitawahab@uika-bogor.ac.id) <sup>2</sup>, [yusup17maolani@gmail.com](mailto:yusup17maolani@gmail.com) <sup>3</sup>

## Article Info

### Article history:

Received 2026-01-16

Revised 2026-03-28

Accepted 2026-04-26

### Keyword:

*Ant Colony Optimization,  
Genetic Algorithm,  
Grid Search,  
Naïve Bayes,  
Tree Augmented Naïve Bayes.*

## ABSTRACT

The Naïve Bayes (NB) algorithm is widely used due to its simplicity and computational efficiency; however, its performance often degrades on real-world data because the assumption of feature independence is frequently violated. This study evaluates two strategies for improving the performance of Naïve Bayes, namely hyperparameter optimization and feature dependency structure enhancement through Tree Augmented Naïve Bayes (TAN). The Breast Cancer Wisconsin (Diagnostic) dataset was selected because it is a representative and widely used public dataset for evaluating medical classification algorithms, thereby facilitating method validation and comparison with previous studies. Experiments were conducted on 30 numerical features with an imbalanced class distribution (62.74% Benign and 37.26% Malignant). The baseline Naïve Bayes model achieved an accuracy of 0.9386. Applying TAN as a standalone approach improved performance to an accuracy of 0.9474 and an F1-score of 0.9286. In contrast, hyperparameter optimization using Genetic Algorithm (GA), Grid Search (GS), and Ant Colony Optimization (ACO) without TAN did not yield meaningful performance improvements; in fact, GA resulted in a decrease in accuracy compared to the baseline model. The combinations of Naïve Bayes + TAN + GS and Naïve Bayes + TAN + ACO achieved the best performance, with identical metrics: accuracy of 0.9561, precision of 0.9744, recall of 0.9048, F1-score of 0.9383, and AUC of 0.9960. These results indicate that improving the feature dependency structure through TAN has a more fundamental impact on enhancing Naïve Bayes performance than hyperparameter optimization alone.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Perkembangan pesat teknologi informasi dan meningkatnya ketersediaan data dalam berbagai bidang mendorong pemanfaatan teknik *Machine Learning* untuk menyelesaikan permasalahan klasifikasi secara lebih efektif dan efisien. Salah satu algoritma klasifikasi yang banyak digunakan adalah *Naïve Bayes*, karena kesederhanaannya, efisiensi komputasi yang tinggi, serta kemampuannya memberikan hasil yang cukup baik pada berbagai jenis dataset. Ketergantungan antar atribut dapat menyebabkan penurunan *Accuracy* dan ketidakstabilan kinerja model, sehingga diperlukan pendekatan lanjutan untuk mengatasi kelemahan mendasar tersebut. Beberapa penelitian

menyimpulkan bahwa algoritma klasifikasi *Naïve Bayes* menunjukkan performa yang kurang maksimal bila dibandingkan dengan algoritma klasifikasi lainnya.

Masalah utama *Naïve Bayes* terletak pada asumsi independensi antar fitur yang jarang terpenuhi pada data nyata, sehingga membatasi kemampuannya dalam memodelkan hubungan antar atribut. Namun, hingga saat ini belum banyak penelitian yang secara eksplisit menguji apakah kelemahan tersebut lebih efektif diatasi melalui optimasi *hyperparameter* atau melalui perbaikan struktur dependensi fitur.

Hasil analisis penggunaan mobil listrik menunjukkan bahwa algoritma *Support Vector Machine* (SVM)

menghasilkan akurasi lebih tinggi dibandingkan *Naïve Bayes*, yaitu masing-masing sebesar 70.83% dan 63.02% [1].

Pada tahun yang sama hasil analisis ulasan aplikasi *LinkedIn* menunjukkan bahwa algoritma SVM memiliki kinerja lebih baik dibandingkan *Naïve Bayes*, dengan akurasi masing-masing sebesar 90% dan 88% [2].

Selanjutnya pada prediksi kanker paru, algoritma *Random Forest* dan *Decision Tree* mencapai performa sempurna dengan nilai *Accuracy*, *Precision*, dan *Recall* sebesar 100%, sedangkan *Naïve Bayes* menunjukkan kinerja yang lebih rendah dengan *Accuracy* 86%, *Precision* 84%, dan *Recall* 89% [3].

Selain itu pada klasifikasi curah hujan di Indonesia, algoritma *Random Forest* menunjukkan kinerja yang jauh lebih unggul dengan *Accuracy* 86.55% dan AUC 97% dibandingkan *Naïve Bayes* yang hanya mencapai *Accuracy* 36.61% dan AUC 61.75% [4].

Pada analisis sentimen *metaverse*, *Regresi Linear* menunjukkan kinerja yang lebih baik dengan peningkatan *Accuracy* dari 91% menjadi 95% setelah optimasi SMOTE, sedangkan *Naïve Bayes* hanya meningkat dari 90% menjadi 91% [5].

Pada analisis sentimen ulasan produk *marketplace*, algoritma *XGBoost* menunjukkan kinerja yang lebih unggul dengan nilai *F1-Score* tertinggi sebesar 0.941 pada kombinasi *Word2Vec*, sedangkan *Naïve Bayes* hanya mencapai *F1-Score* 0.915 dengan TF-IDF dan 0.900 dengan *Word2Vec* [6].

Penelitian ini membandingkan peningkatan kinerja *Naïve Bayes* melalui optimasi *hyperparameter* menggunakan *Genetic Algorithm*, *Ant Colony Optimization*, dan *Grid Search*, serta melalui perbaikan struktur dependensi fitur dengan *Tree Augmented Naïve Bayes* (TAN).

*Genetic Algorithm* (GA) adalah algoritma optimasi metaheuristik yang meniru proses evolusi biologis dengan menerapkan seleksi alam untuk menghasilkan solusi optimal [7].

*Grid Search* merupakan metode konvensional untuk optimasi *hyperparameter* yang dilakukan dengan mengevaluasi secara sistematis kombinasi parameter tertentu dalam ruang *hyperparameter* algoritma pelatihan [8].

*Ant Colony Optimization* dikembangkan sebagai suatu simulasi berbasis *multi-agen* yang menggunakan analogi perilaku alami semut untuk membantu menyelesaikan berbagai permasalahan dalam ruang fisik [9].

*Tree Augmented Naïve Bayes* (TAN) adalah pengembangan *Naïve Bayes* dengan struktur pohon, di mana simpul kelas terhubung ke semua atribut dan setiap atribut hanya memiliki satu induk atribut tambahan, sehingga setiap simpul memiliki maksimal dua induk [10].

*Genetic Algorithm* (GA) dipilih karena terbukti mampu meningkatkan kinerja *Naïve Bayes* dalam deteksi penipuan kartu kredit, dengan *Accuracy* 99.90% pada kombinasi *Naïve Bayes* + GA, lebih tinggi dibandingkan kombinasi *Naïve Bayes* + *Bagging* yang mencapai *Accuracy* 99.44% [11]. *Genetic Algorithm* (GA) terbukti efektif dalam mengoptimalkan *Convolutional Neural Network* (CNN)

untuk prediksi penyakit jantung dengan tingkat *Accuracy* sebesar 85.10% [12].

Salah satu pendekatan yang banyak digunakan untuk mengatasi keterbatasan tersebut adalah *Tree Augmented Naïve Bayes* (TAN). Metode ini dikenal lebih efisien dalam memodelkan dependensi fitur dibandingkan *Bayesian Network* yang lebih kompleks [13]. Dengan struktur yang relatif sederhana (*parsimonious*) namun tetap mampu menghasilkan performa klasifikasi yang kompetitif pada berbagai kasus aplikasi [14]. Hal ini didukung oleh beberapa penelitian sebelumnya, di mana TAN terbukti menjadi model terbaik dalam klasifikasi biotipe wajah dengan *Accuracy* sebesar  $74.09 \pm 3.62$  dan *F1-Score*  $75.14 \pm 3.24$  [15]. Serta menunjukkan efektivitas tinggi dalam memprediksi tingkat keparahan kecelakaan di Korea Selatan dengan *Accuracy* mencapai 91.5% [16].

*Grid Search* dipilih karena optimasi *hyperparameter* terbukti mampu meningkatkan akurasi model, sebagaimana ditunjukkan pada peningkatan *Accuracy* SVM dari 77.55% menjadi 82,00% melalui pendekatan *Randomized Search* [17]. Optimasi parameter menggunakan *Grid Search* dengan validasi *K-Fold* (*10-fold*) dan kernel RBF ( $C = 10$ ,  $\gamma = 0.2$ ) terbukti meningkatkan *Accuracy* model SVM dari 79.29% menjadi 85.71%, atau mengalami peningkatan sebesar 6.24% [18].

*Ant Colony Optimization* (ACO) dipilih karena kombinasi *Random Forest* dengan ACO terbukti meningkatkan *Accuracy* klasifikasi obesitas menjadi 96.17%, dibandingkan *Random Forest* tanpa optimasi yang hanya 94.49% [19]. Penelitian berikutnya menemukan bahwa ACO lebih unggul daripada *Genetic Algorithm* dalam optimasi *hyperparameter* CNN, mencapai *Accuracy* lebih tinggi pada data pelatihan (99.69% vs 97.59%) dan validasi (97.46% vs 94.60%), walau waktu pelatihannya sedikit lebih lama [20].

Penelitian ini bertujuan untuk meningkatkan dan mengevaluasi kinerja algoritma klasifikasi *Naïve Bayes* melalui penerapan optimasi *hyperparameter* serta perbaikan struktur dependensi fitur menggunakan *Tree Augmented Naïve Bayes* (TAN). Dalam tahap awal, performa *Naïve Bayes* dibandingkan dengan *Support Vector Machine* (SVM) sebagai model *baseline* untuk memperoleh gambaran komparatif. Selanjutnya, penelitian ini menganalisis pengaruh optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA), *Grid Search* (GS), dan *Ant Colony Optimization* (ACO), serta menilai efektivitas TAN dalam mengatasi keterbatasan asumsi independensi antar fitur. Evaluasi dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *Area Under the Curve* (AUC) guna memastikan penilaian kinerja yang komprehensif.

Kebaruan (*Novelty*) dari penelitian ini terletak pada evaluasi komprehensif dan sistematis terhadap dua pendekatan peningkatan kinerja algoritma *Naïve Bayes*, yaitu optimasi *hyperparameter* dan perbaikan struktur dependensi fitur menggunakan *Tree Augmented Naïve Bayes* (TAN), dalam satu kerangka eksperimen yang sama. Berbeda dengan penelitian-penelitian sebelumnya yang umumnya hanya

menekankan pada peningkatan performa melalui optimasi parameter atau sekedar membandingkan *Naïve Bayes* dengan algoritma lain, penelitian ini secara eksplisit menunjukkan bahwa optimasi *hyperparameter* saja tidak cukup efektif apabila kelemahan mendasar asumsi independensi fitur pada *Naïve Bayes* tidak diperbaiki. Selain itu, penelitian ini menghadirkan analisis empiris yang jelas mengenai peran dominan TAN dalam meningkatkan kinerja *Naïve Bayes*, serta membuktikan bahwa kombinasi *Naïve Bayes*+TAN+ACO diikuti dengan *NAïve Bayes*+TAN+Grid Search menghasilkan performa terbaik dibandingkan kombinasi lainnya. Dengan demikian, penelitian ini memberikan kontribusi baru dalam memahami prioritas strategi peningkatan kinerja *Naïve Bayes*, yaitu bahwa perbaikan struktur dependensi fitur memiliki pengaruh yang lebih signifikan dibandingkan optimasi *hyperparameter*.

Manfaat dari penelitian ini diharapkan dapat memberikan kontribusi baik secara teoretis maupun praktis. Secara teoretis, penelitian ini memberikan pemahaman yang lebih mendalam mengenai pengaruh perbaikan struktur dependensi fitur dibandingkan dengan optimasi *hyperparameter* dalam meningkatkan kinerja algoritma *Naïve Bayes*. Secara praktis, hasil penelitian ini dapat dijadikan acuan bagi peneliti dan praktisi dalam menerapkan *Naïve Bayes* pada *dataset* yang memiliki ketergantungan antar fitur, khususnya pada domain medis seperti klasifikasi kanker payudara. Selain itu, penelitian ini juga memberikan manfaat metodologis berupa kerangka evaluasi yang sistematis dalam membandingkan efektivitas optimasi dan pendekatan struktural pada algoritma klasifikasi *probabilistic*.

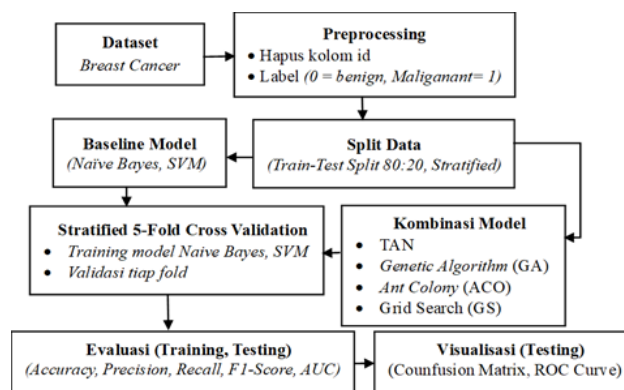
Fokus penelitian ini dibatasi pada pengembangan dan evaluasi algoritma *Naïve Bayes* sebagai model klasifikasi utama, dengan penerapan *Tree Augmented Naïve Bayes* (TAN) untuk memperbaiki asumsi independensi fitur serta optimasi menggunakan *Genetic Algorithm* (GA), *Grid Search* (GS), dan *Ant Colony Optimization* (ACO). *Dataset* yang digunakan adalah *Breast Cancer Dataset* dengan karakteristik fitur numerik dan ketidakseimbangan kelas moderat. Evaluasi performa model dilakukan menggunakan pendekatan *Stratified Cross-Validation* dan pengujian pada data uji, dengan metrik evaluasi utama berupa *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *AUC*.

## II. METODE

### A. ALUR PENELITIAN

Alur penelitian dimulai dengan tahap *preprocessing* data, kemudian *dataset* dibagi menjadi 80% data pelatihan dan 20% data pengujian. Model *baseline* dibangun menggunakan algoritma *Naïve Bayes*. Selanjutnya, dilakukan pengembangan model melalui kombinasi *Naïve Bayes* dengan metode optimasi *Genetic Algorithm* (GA), *Grid Search* (GS), dan *Ant Colony Optimization* (ACO), serta perbaikan struktur dependensi fitur menggunakan *Tree Augmented Naïve Bayes* (TAN). Evaluasi kinerja model dilakukan menggunakan *K-Fold Cross Validation* dan pengujian data uji, kemudian

hasilnya dianalisis dan divisualisasikan, alur penelitian digambarkan pada Gambar 1. Alur Penelitian.



Gambar 1. Alur Penelitian

### B. Profile Dataset

*Dataset* penelitian ini berasal dari Dr. William H. Wolberg (*University of Wisconsin Hospital*) yang diperoleh dari sampel cairan massa payudara. Analisis sitologi menggunakan perangkat lunak *Xcyt* untuk mengekstraksi 30 fitur numerik sel (rata-rata, kesalahan standar, dan nilai ekstrem), yang dinormalisasi pada skala 1–10, dari kondisi jinak hingga ganas [21]. Penelitian ini menggunakan *dataset* klasifikasi diagnosis kanker payudara yang disimpan dalam file *proses1.csv*, dengan variabel target berupa diagnosis. Pada tahap awal, data dibaca menggunakan pustaka *pandas*, kemudian label pada variabel target dikonversi ke bentuk numerik menggunakan *LabelEncoder*. Dalam pengkodean ini, kelas *Benign* direpresentasikan sebagai 0 dan kelas *Malignant* direpresentasikan sebagai 1

### C. Preprocessing

*Pre-processing* merupakan tahap persiapan data dalam *Machine Learning* yang bertujuan mengolah data mentah agar sesuai dengan kebutuhan algoritma pembelajaran. Tahap ini berperan penting karena kualitas data sangat menentukan kinerja model yang dihasilkan [22]. Pada penelitian ini, proses *preprocessing* terdiri atas dua langkah utama, yaitu menghilangkan atribut ID yang tidak memiliki kontribusi terhadap proses klasifikasi serta melakukan transformasi label kelas target *Benign* dan *Malignant* ke dalam bentuk numerik, masing-masing direpresentasikan sebagai nilai 0 dan 1.

### D. Split Data

*Splitting* data merupakan proses pembagian data yang digunakan dalam penelitian menjadi dua atau lebih bagian, biasanya digunakan untuk menguji model atau algoritma. *Dataset* umumnya dibagi menjadi data (*training*) dan data uji (*testing*) [23].

### E. Baseline Model Naïve Bayes

Pada penelitian ini yang menjadi *Baseline Model* adalah Algoritma klasifikasi *Naïve Bayes* dan *Support Vector Machine* tujuannya adalah untuk mengukur performa awal dari algoritma klasifikasi *Naïve Bayes* dan *SVM*. *Naïve Bayes* adalah algoritma klasifikasi yang bekerja dengan pendekatan probabilistik sederhana, di mana proses prediksinya dibangun dengan mengasumsikan bahwa setiap variabel penjelas tidak saling bergantung satu sama lain [24]. Bentuk persamaan *Naïve Bayes* ditunjukkan pada persamaan (1)

$$p(H|E) = \frac{p(H) \times p(H|E)}{p(E)} \quad (1)$$

Pada persamaan ini (1),  $H$  menyatakan hipotesis,  $X$  merupakan data atau label,  $P(H)$  adalah peluang awal hipotesis,  $P(X|H)$  menunjukkan peluang data pada hipotesis tertentu,  $P(X)$  adalah peluang data, dan  $P(H|X)$  merupakan probabilitas akhir hipotesis setelah mempertimbangkan data.

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# Inisialisasi dan training Gaussian Naïve Bayes
model = GaussianNB()
model.fit(X_train, y_train)

# Prediksi dan probabilitas kelas
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[: , 1]
```

Gambar 2. *Baseline Model Naïve Bayes*

Gambar 2. *Baseline Model Naïve Bayes Menampilkan*, implementasi inti algoritma *Gaussian Naïve Bayes* yang digunakan sebagai model *baseline* dalam penelitian ini, mencakup proses pelatihan dan inferensi *probabilistic*.

#### F. Peningkatan Algoritma *Naïve Bayes* Melalui Pendekatan Tunggal

Pada pendekatan tunggal, model *Naïve Bayes* dikombinasikan dengan *Tree Augmented Naïve Bayes* (TAN). Struktur TAN dipelajari menggunakan algoritma *TreeSearch* dengan variabel kelas sebagai pusat struktur, sehingga memungkinkan adanya satu dependensi tambahan antar atribut dalam bentuk struktur pohon bersyarat, sebagaimana dirumuskan pada Persamaan (2).

$$P(C, X_1, \dots, X_n) = P(C) \prod_{i=1}^n P(X_i|C, Pa(x_i)) \quad (2)$$

Dimana setiap fitur  $X_i$  bergantung pada kelas dan maksimal satu fitur lain sebagai *parent*. Struktur pohon dipelajari menggunakan pendekatan *Tree Search*, sedangkan parameter

probabilitas diestimasi menggunakan pendekatan frekuensi relatif (*Maximum Likelihood Estimation*) berdasarkan distribusi data pelatihan. Ilustrasi struktur TAN ditunjukkan pada Gambar 3.

```
# 4.b) Learn TAN structure from X_tr_disc
ts = TreeSearch(X_tr_disc)
try:
    tan_structure = ts.estimate(estimator_type="tan", class_node=target)
except Exception as e:
    # fallback if method signature differs in other pgmpy versions
    tan_structure = ts.tan(class_node=target)
edges = tan_structure.edges()
print("TAN edges count:", len(edges))
```

Gambar 3. Perbaikan Fitur *Naïve Bayes* dengan TAN

Pada tahap ini dilakukan optimasi *algoritma Naïve Bayes* menggunakan *Ant Colony Optimization* (ACO) dengan menjadikan kinerja model sebagai fungsi objektif. Setiap solusi yang dibentuk oleh semut dievaluasi berdasarkan nilai akurasi hasil validasi silang, yang kemudian digunakan untuk memperbarui intensitas *feromon*. Probabilitas pemilihan solusi pada iterasi ke- $t$  ditentukan oleh distribusi *feromon* dan dirumuskan pada Persamaan (3)

$$P_i(t) = \frac{[t_{ij}(t)]}{\sum_{j=1}^n T_j(t)} \quad (3)$$

Dimana  $T_i(t)$  menyatakan intensitas *feromon* pada kandidat ke- $i$  dan  $n$  adalah jumlah total kandidat solusi. Pembaruan *feromon* dilakukan melalui mekanisme *evaporasi* dan *deposisi*, sebagaimana ditunjukkan pada Persamaan (4)

$$T_i(t+1) = (1 - \rho) T_i(t) + f(x) \cdot x_i \quad (4)$$

Dimana  $\rho$  merupakan laju *evaporasi feromon*,  $f(x)$  adalah nilai fungsi objektif berupa akurasi rata-rata hasil validasi silang, dan  $x_i$  menunjukkan keterpilihan kandidat solusi ke- $i$ .

Dalam penelitian ini, ACO digunakan untuk mengoptimasi *hyperparameter var\_smoothing* pada *Gaussian Naïve Bayes*. Ruang pencarian parameter didefinisikan dalam rentang  $10^{-12}$  hingga  $10^{-6}$  pada skala logaritmik. Proses optimasi dilakukan menggunakan 20 semut dan 20 iterasi, dengan evaluasi berbasis *5-Fold Stratified Cross Validation*. Parameter terbaik dipilih berdasarkan nilai akurasi tertinggi dan selanjutnya digunakan dalam evaluasi model pada data uji.

Optimasi *Naïve Bayes* dilakukan menggunakan *Grid Search* untuk menentukan nilai optimal *var\_smoothing* dalam rentang  $10^{-12}$  hingga  $10^{-6}$ . Evaluasi dilakukan menggunakan *5-Fold Stratified Cross Validation* dengan metrik akurasi sebagai fungsi objektif. Parameter terbaik dipilih berdasarkan nilai akurasi rata-rata tertinggi, kemudian digunakan dalam pelatihan model akhir dan pengujian pada data uji.

Secara formal, proses pemilihan parameter optimal dirumuskan pada Persamaan (5).

$$\theta^* = \arg \max_{\theta \in \Theta} \frac{1}{K} \sum_{k=1}^n \mathcal{M}^{(k)}(\theta) \quad (5)$$

Dimana,  $\theta$  merupakan parameter yang dioptimasi  $\Theta$  adalah ruang pencarian parameter,  $K$  adalah jumlah *Fold* dalam validasi silang, dan  $\mathcal{M}^{(k)}(\theta)$  adalah nilai metrik evaluasi pada *Fold* ke- $k$ . Nilai rata-rata dari seluruh *Fold* digunakan untuk menentukan parameter dengan kinerja terbaik

Optimasi *Naïve Bayes* dilakukan menggunakan *Genetic Algorithm* (GA) untuk menentukan nilai optimal *var\_smoothing* dalam rentang  $10^{-12}$  hingga  $10^{-6}$ . Proses optimasi menggunakan populasi awal sebanyak 10 individu dan berlangsung selama 10 generasi dengan mekanisme seleksi, *Crossover*, dan mutasi. Evaluasi dilakukan menggunakan AUC berbasis *5-Fold Stratified Cross Validation*. Parameter terbaik dipilih berdasarkan nilai AUC tertinggi dan digunakan untuk evaluasi model pada data uji.

#### G. Peningkatan Algoritma Naïve Bayes Melalui Pendekatan Terintegrasi

Pada pendekatan terintegrasi ini, *Tree Augmented Naïve Bayes* (TAN) digunakan untuk memodelkan dependensi antarfitur dengan variabel kelas sebagai *parent* utama, sementara *Ant Colony Optimization* (ACO) mengoptimasi parameter *preprocessing* dan estimasi probabilitas tanpa mengubah struktur dasar model. Parameter yang dioptimasi meliputi jumlah *bin* diskretisasi (*n\_bins*) dan jenis *prior Bayesian* (*prior\_type*).

Seluruh fitur numerik didiskretisasi menggunakan *KBinsDiscretizer* dengan *encode = ordinal* dan *strategy = quantile*, di mana nilai *n\_bins* dipilih dari {3, 5, 7, 9} dan *prior\_type* dari {*BDeu*, *K2*}. Struktur TAN dibangun menggunakan *TreeSearch* (*estimator\_type = "tan"*, *class\_node = diagnosis*), dan parameter probabilitas diestimasi menggunakan *BayesianEstimator*. Prediksi dilakukan melalui inferensi *Variable Elimination* berdasarkan probabilitas *posterior* tertinggi.

Optimasi ACO menggunakan 10 semut dan 10 iterasi dengan laju evaporasi 0,3 dan konstanta deposisi 1,0. Fungsi *fitness* berupa rata-rata *F1-score* dari *5-Fold Stratified Cross-Validation*, dengan proses diskretisasi yang hanya dilatih pada data pelatihan untuk mencegah kebocoran data.

Pada penelitian ini, *Grid Search* digunakan untuk mengoptimasi parameter *Tree Augmented Naïve Bayes* (TAN), yaitu jumlah *bin diskretisasi* (*n\_bins*) dan jenis *prior Bayesian* (*prior\_type*), dengan rata-rata *F1-score* dari *5-fold Stratified Cross-Validation* sebagai dasar pemilihan parameter terbaik. Seluruh fitur numerik didiskretisasi menggunakan *KBinsDiscretizer* (*encode = ordinal*, *strategy = quantile*), kemudian struktur dependensi dipelajari menggunakan *TreeSearch* dengan variabel kelas sebagai pusat (*class\_node = diagnosis*), dan parameter probabilitas diestimasi menggunakan *BayesianEstimator*. Prediksi dilakukan melalui inferensi *Variable Elimination* berdasarkan probabilitas *posterior* tertinggi.

*Grid Search* berfungsi menentukan kombinasi parameter optimal dari *n\_bins* {3, 5, 7, 9} dan *prior\_type* {*BDeu*, *K2*} tanpa mengubah formulasi probabilistik TAN. Model terbaik selanjutnya dievaluasi menggunakan *5-fold Cross-Validation* dan data uji 20% dengan metrik *Accuracy*, *Precision*, *Recall*, *F1-score*, dan AUC, serta dilengkapi analisis struktur dependensi, *mutual information*, dan efisiensi komputasi.

Pada penelitian ini, *Genetic Algorithm* (GA) digunakan untuk mengoptimasi parameter pada *Tree Augmented Naïve Bayes* (TAN), khususnya jumlah *bin diskretisasi* (*n\_bins*), tanpa mengubah struktur probabilistik model. Dataset dibagi menjadi 80% data latih dan 20% data uji secara *stratified* (*random\_state = 42*). Seluruh fitur numerik didiskretisasi menggunakan *KBinsDiscretizer* dengan *encode = ordinal* dan *strategy = quantile*. Struktur dependensi antarfitur dipelajari menggunakan *TreeSearch* dengan *estimator\_type = "tan"* dan *class\_node = diagnosis*, sedangkan parameter probabilitas diestimasi menggunakan *BayesianEstimator* dengan *prior BDeu*. Proses optimasi dilakukan menggunakan GA sederhana dengan ukuran populasi 6 dan 4 generasi, menggunakan rata-rata *Macro F1-score* dari *5-fold Stratified Cross-Validation* sebagai fungsi *fitness*. Model terbaik kemudian dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-score*, dan AUC, serta dianalisis melalui struktur dependensi, *mutual information*, *confusion matrix*, ROC curve, dan kompleksitas waktu, sehingga memberikan evaluasi yang komprehensif dari sisi kinerja, interpretabilitas, dan efisiensi komputasi.

#### H. Stratified Cross Validation

*Stratified K-Fold Cross-Validation* membagi data ke dalam  $K$  subset dengan menjaga proporsi kelas pada setiap *fold*. Model dilatih pada  $K - 1$  fold dan diuji pada satu *fold* secara bergantian, kemudian kinerja model dihitung sebagai rata-rata hasil evaluasi dari seluruh fold [25].

#### I. Evaluasi

*Confusion Matrix* merupakan metode evaluasi yang digunakan untuk menilai kinerja algoritma klasifikasi dengan membandingkan hasil prediksi terhadap data aktual. Matriks ini terdiri dari empat komponen utama, yaitu *True Positive* (TP) dan *True Negative* (TN) sebagai prediksi yang sesuai dengan nilai aktual, serta *False Positive* (FP) dan *False Negative* (FN) sebagai prediksi yang tidak sesuai dengan nilai aktual [26].

*Accuracy* menunjukkan tingkat ketepatan prediksi model secara keseluruhan (6), *Precision* mengukur ketepatan prediksi kelas positif (7), *Recall* menilai kemampuan model dalam mengenali seluruh data positif (8), sedangkan *F1-Score* merupakan rata-rata harmonik antara *Precision* dan *Recall* untuk menyeimbangkan kedua metrik tersebut (9)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$F1 - \text{Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (9)$$

#### J. Visualisasi

Visualisasi digunakan untuk menyajikan hasil evaluasi model secara grafis agar lebih mudah dipahami dan dianalisis. Salah satu bentuk visualisasi yang digunakan adalah *Confusion Matrix* dan *ROC Curve*. *ROC Curve* merupakan alat evaluasi yang menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Sementara itu, nilai *Area Under the Curve* (AUC) digunakan untuk menilai kinerja model secara keseluruhan, di mana performa model dinilai semakin baik apabila nilai AUC mendekati satu [27].

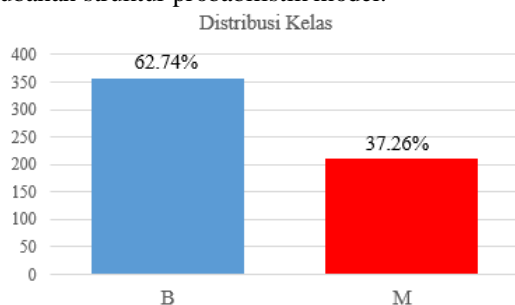
### III. HASIL DAN PEMBAHASAN

#### A. Dataset

Dataset *Breast Cancer* yang digunakan dalam penelitian ini diperoleh dari *Kaggle* dan terdiri atas 569 sampel dengan 30 fitur numerik serta satu label target, yaitu benign dan malignant. Dataset tersebut menunjukkan ketidakseimbangan kelas yang bersifat moderat, dengan proporsi 62,74% *benign* dan 37,26% *malignant*. Eksperimen dilakukan dengan pembagian data 80:20 menggunakan *Stratified Sampling*, kemudian dievaluasi kembali pada data latih menggunakan *Stratified 5-Fold Cross-Validation* guna memperoleh estimasi kinerja yang lebih *robust*.

Pada model *Naïve Bayes* dasar, fitur numerik digunakan tanpa melalui proses normalisasi maupun standarisasi. Sebaliknya, pada model *Tree Augmented Naïve Bayes* (TAN), data numerik terlebih dahulu *didiskretisasi* menggunakan *KBinsDiscretizer* karena metode ini memerlukan data *diskrit*. Penelitian ini juga tidak menerapkan teknik *resampling* agar distribusi kelas asli pada dataset tetap dipertahankan.

Sebagai bagian dari protokol evaluasi, penyesuaian *threshold* klasifikasi tidak dilakukan dalam penelitian ini karena fokus utama penelitian adalah membandingkan pengaruh perbaikan struktur model dan optimasi parameter terhadap kinerja *Naïve Bayes*, bukan optimasi batas keputusan (*decision boundary*). Seluruh model dievaluasi menggunakan *threshold default* yang sama untuk memastikan perbandingan yang adil serta untuk mengisolasi dampak perubahan struktur probabilitik model.



Gambar 2. Distribusi Kelas *Breast Cancer*

#### B. Split Dataset

Pada penelitian ini sub set pertama 80% digunakan untuk (*Training Set*), dan sub set ke dua 20 % digunakan untuk (*Testing Set*) data pengujian. Data dipisahkan menjadi variabel fitur (X) dan variabel target (y). Selanjutnya, dataset dibagi menjadi data latih dan data uji menggunakan metode *stratified hold-out* dengan proporsi 80% untuk data latih dan 20% untuk data uji. Teknik stratifikasi digunakan untuk mempertahankan proporsi distribusi kelas pada kedua subset data, sehingga data latih dan data uji tetap representatif terhadap distribusi data asli. Pembagian data dilakukan dengan *random\_state = 42* agar eksperimen dapat direplikasi secara konsisten

#### C. Stratified Cross Validation

Hasil *Stratified Cross Validation* menunjukkan bahwa model SVM mencapai kinerja terbaik dengan *Accuracy* sebesar 0,9714 dan standar deviasi yang rendah, yang mengindikasikan stabilitas model yang tinggi. Model *Naïve Bayes* dasar memiliki performa yang lebih rendah dengan variabilitas yang lebih besar pada beberapa metrik evaluasi.

Penerapan *Tree Augmented Naïve Bayes* (TAN) memberikan peningkatan kinerja dibandingkan *Naïve Bayes*, meskipun tidak signifikan. Integrasi metode optimasi, khususnya pada kombinasi *Naïve Bayes + TAN + ACO* dan *Naïve Bayes + TAN + Grid Search*, menunjukkan peningkatan performa yang kompetitif dengan *Accuracy* mencapai 0,9670 serta tingkat stabilitas yang lebih baik.

Secara keseluruhan, hasil ini menunjukkan bahwa pemodelan dependensi fitur dan optimasi parameter berkontribusi positif dalam meningkatkan kinerja dan konsistensi model klasifikasi, hasil *Stratified Cross Validation* dapat di lihat pada Tabel 1.

TABEL 1.  
STARTIFIED CROSS VALIDATION

| Model                           | Accuracy | Precision | Recall | F1-Score | AUC    |
|---------------------------------|----------|-----------|--------|----------|--------|
| Naïve Bayes                     | 0.9407   | 0.9402    | 0.9000 | 0.9185   | 0.9878 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| SVM                             | 0.9714   | 0.9769    | 0.9471 | 0.9610   | 0.9949 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| Naïve Bayes + TAN               | 0.9429   | 0.9456    | 0.9000 | 0.9218   | 0.9801 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| Naïve Bayes + ACO               | 0.9407   | 0.9450    | 0.8941 | 0.9178   | 0.9875 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| Naïve Bayes + TAN + ACO         | 0.9670   | 0.9608    | 0.9529 | 0.9559   | 0.9928 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| Naïve Bayes + TAN + Grid Search | 0.9670   | 0.9667    | 0.9642 | 0.9648   | 0.9928 |
|                                 | Std      | Std       | Std    | Std      | Std    |
| Naïve Bayes + ACO + Grid Search | 0.9670   | 0.9667    | 0.9642 | 0.9648   | 0.9928 |
|                                 | Std      | Std       | Std    | Std      | Std    |

#### D. Baseline Model Naïve Bayes

Meskipun *Naïve Bayes* pada Tabel 2. Menunjukkan AUC yang tinggi, ketidakseimbangan antara *Precision* dan *Recall* mengindikasikan bahwa model masih bias terhadap kelas mayoritas, sehingga membuka peluang untuk peningkatan melalui perbaikan struktur model

TABEL 2.  
BASELINE MODEL NAÏVE BAYES

| Model       | Accuracy | Precision | Recall | F1-Score | AUC    |
|-------------|----------|-----------|--------|----------|--------|
| Naïve Bayes | 0.9386   | 1.0000    | 0.8333 | 0.9091   | 0.9934 |
| SVM         | 0.9737   | 1.0000    | 0.9286 | 0.9630   | 0.9947 |

Hasil pengujian pada Tabel 3. Menunjukkan bahwa *Naïve Bayes* dan SVM memiliki karakteristik komputasi yang berbeda. *Naïve Bayes* lebih efisien pada tahap pelatihan karena proses pembentukan modelnya lebih sederhana, sedangkan SVM memerlukan komputasi lebih tinggi pada tahap training karena harus menentukan *hyperplane* optimal. Namun, pada tahap inferensi, SVM cenderung lebih efisien sehingga lebih cepat dalam melakukan prediksi pada data baru. Temuan ini menunjukkan bahwa evaluasi model perlu mempertimbangkan tidak hanya akurasi, tetapi juga efisiensi komputasi sesuai kebutuhan implementasi sistem.

TABEL 3  
WAKTU PELATIHAN DAN INFERENSI BASELINE MODEL

| Model       | Training Time | Inference Time | Rata - rata |
|-------------|---------------|----------------|-------------|
| Naïve Bayes | 0.012985      | 0.009367       | 0.00008217  |
| SVM         | 0.025214      | 0.005558       | 0.00004875  |

Berdasarkan Tabel 4. matriks evaluasi per kelas, SVM menunjukkan kinerja yang lebih baik dibandingkan *Naïve Bayes* pada kelas *benign* dan *malignant*. Kedua model sama-sama mampu mengenali kelas *benign* dengan sangat baik, tetapi SVM memiliki nilai *Precision* dan *F1-score* yang lebih tinggi. Pada kelas *malignant*, SVM juga lebih unggul karena memberikan keseimbangan metrik yang lebih baik, sehingga menunjukkan kemampuan klasifikasi yang lebih konsisten pada kedua kelas.

TABEL 4  
KINERJA NAÏVE BAYES DAN SVM BERDASARKAN METRIK EVALUASI SETIAP KELAS

| Model       | Kelas     | Precision | Recall | F1-Score | Support |
|-------------|-----------|-----------|--------|----------|---------|
| Naïve Bayes | Benignt   | 0.9114    | 1.0000 | 0.9536   | 72      |
|             | Malignant | 1.0000    | 0.8333 | 0.9091   | 42      |
| SVM         | Benignt   | 0.9600    | 1.0000 | 0.9796   | 72      |
|             | Malignant | 1.0000    | 0.9286 | 0.9630   | 42      |

#### E. Peningkatan Kinerja Algoritma Naïve Bayes Secara Tunggal

Pada tahap peningkatan kinerja algoritma secara tunggal dilakukan beberapa kombinasi pada Algoritma klasifikasi *Naïve Bayes* melalui optimasi parameter menggunakan *Genetic Algorithm* (GA), *Ant Colony Optimization* (ACO),

dan *Grid Search*, serta pengembangan struktur model dengan *Tree Augmented Naïve Bayes* (TAN). Hasil kombinasi tersebut disajikan pada Tabel 5.

TABEL 5.  
PENINGKATAN KINERJA ALGORITMA NAÏVE BAYES SECARA TUNGGAL

| Model            | Accuracy | Precision | Recall | F1-Score | AUC    |
|------------------|----------|-----------|--------|----------|--------|
| NB + TAN         | 0.9474   | 0.9500    | 0.9048 | 0.9268   | 0.9894 |
| NB + ACO         | 0.9386   | 1.0000    | 0.8333 | 0.9091   | 0.9940 |
| NB + Grid Search | 0.9298   | 0.9722    | 0.8333 | 0.8974   | 0.9924 |
| NB + GA          | 0.9211   | 0.9231    | 0.8571 | 0.8889   | 0.9891 |

Tabel 5. Menunjukkan bahwa penerapan *Tree Augmented Naïve Bayes* (TAN) memberikan peningkatan kinerja paling signifikan dibandingkan teknik optimasi lainnya. Kombinasi NB+TAN menghasilkan performa yang lebih seimbang antara *Precision* dan *Recall*. Hal ini terjadi karena TAN memodelkan ketergantungan antar fitur secara eksplisit, sehingga mengurangi kesalahan akibat asumsi independensi pada *Naïve Bayes* standar.

TABEL 6.  
KOMBINASI FITUR BERDASARKAN NILAI CONDITIONAL MUTUAL INFORMATION (CMI)

| No | Featur 1       | Featur 2        | CMI Score |
|----|----------------|-----------------|-----------|
| 1  | radius_mean    | area_mean       | 1.164790  |
| 2  | radius_mean    | perimeter_mean  | 1.018162  |
| 3  | perimeter_mean | area_mean       | 1.002856  |
| 4  | radius_worst   | area_worst      | 1.002602  |
| 5  | radius_worst   | perimeter_worst | 0.854795  |

Tabel 6. Menunjukkan bahwa struktur dependensi pada model TAN dibangun dari hubungan antar fitur yang memiliki nilai *Conditional Mutual Information* tertinggi terhadap variabel kelas. Hasil tersebut memperlihatkan bahwa hubungan terkuat didominasi oleh fitur-fitur yang merepresentasikan ukuran tumor, terutama radius, perimeter, dan area. Temuan ini menunjukkan bahwa fitur-fitur morfologis memiliki keterkaitan statistik yang kuat dan berperan penting dalam pembentukan struktur TAN.

Berdasarkan Tabel 6. dan Tabel 5. Dapat dijelaskan bahwa keunggulan model NB + TAN tidak hanya ditunjukkan oleh hasil klasifikasi yang lebih baik, tetapi juga oleh kemampuannya dalam memodelkan hubungan dependensi antar fitur. Dalam penelitian ini, struktur TAN dibangun setelah data *didiskretisasi*, kemudian hubungan antar fitur dipelajari untuk memilih pasangan atribut yang paling informatif terhadap variabel kelas. Hasil analisis menunjukkan bahwa dependensi terkuat terletak pada kelompok fitur ukuran tumor, khususnya radius, perimeter, dan area, baik pada ukuran rata-rata maupun kondisi terburuk. Dengan demikian, peningkatan performa model NB + TAN dapat dijelaskan oleh kemampuannya dalam mengurangi keterbatasan asumsi independensi pada *Naïve Bayes* standar, sehingga proses klasifikasi menjadi lebih representatif dan akurat.

TABEL 7.  
WAKTU PELATIHAN DAN INFERENSI MELALUI PENDEKATAN TUNGGAL

| Model                     | Training Time | Inference Time | Rata - rata |
|---------------------------|---------------|----------------|-------------|
| Naïve Bayes + TAN         | 0.270480      | 0.052592       | 0.00046133  |
| Naïve Bayes + ACO         | 0.003625      | 0.004427       | 0.00003883  |
| Naïve Bayes + Grid Search | 0.003411      | 0.003951       | 0.00003466  |
| Naïve Bayes + GA          | 0.003727      | 0.003643       | 0.00003196  |

Hasil pengujian pada Tabel 7. Menunjukkan bahwa model *Naïve Bayes* + TAN memiliki kompleksitas komputasi yang paling tinggi pada tahap pelatihan maupun inferensi. Kondisi ini mengindikasikan bahwa pembentukan struktur ketergantungan antaratribut pada TAN memerlukan beban komputasi yang lebih besar dibandingkan pendekatan lainnya. Sebaliknya, model *Naïve Bayes* + *Grid Search*, *Naïve Bayes* + ACO, dan *Naïve Bayes* + GA menunjukkan efisiensi waktu yang lebih baik. Secara keseluruhan, *Naïve Bayes* + GA cenderung memberikan kinerja komputasi paling efisien, diikuti oleh *Grid Search* dan ACO. Temuan ini menegaskan bahwa perbedaan pendekatan optimasi dan pengembangan struktur model berpengaruh terhadap efisiensi komputasi, sehingga aspek waktu pelatihan dan inferensi perlu dipertimbangkan dalam pemilihan model

#### F. Peningkatan Kinerja Algoritma Naïve Bayes Melalui Pendekatan Terintegerasi

Selanjutnya, dilakukan peningkatan kinerja algoritma *Naïve Bayes* secara Terintegerasi, peningkatan kinerja Algoritma *Naïve Bayes* dengan cara mengintegrasikan berbagai metode optimasi dan perbaikan fitur dengan *Tree Augmented Naïve Bayes* (TAN). Dari hasil pengujian tersebut di peroleh hasil yang dijelaskan pada Tabel 8.

TABEL 8.  
PENINGKATAN KINERJA ALGORITMA NAÏVE BAYES MELALUI PENDEKATAN TERINTEGERASI

| Model          | Accuracy | Precision | Recall | F1-Score | AUC    |
|----------------|----------|-----------|--------|----------|--------|
| NB + TAN+ GS   | 0.9561   | 0.9744    | 0.9048 | 0.9383   | 0.9960 |
| NB + TAN + GA  | 0.9123   | 0.9211    | 0.8333 | 0.8750   | 0.9841 |
| NB + TAN + ACO | 0.9561   | 0.9744    | 0.9048 | 0.9383   | 0.9960 |

Pada Tabel 8. Tabel tersebut menunjukkan bahwa peran TAN memberikan pengaruh terhadap kinerja model *Naïve Bayes*. Secara umum, perbaikan independensi oleh TAN yang

dikombinasikan dengan *Grid Search* dan *Ant Colony Optimization* menunjukkan performa yang lebih baik dan lebih konsisten dibandingkan *Genetic Algorithm* pada seluruh metrik evaluasi. Hasil ini mengindikasikan bahwa kedua pendekatan tersebut lebih efektif dalam meningkatkan kemampuan klasifikasi model, baik dari aspek *Accuracy*, keseimbangan prediksi, maupun kemampuan pemisahan kelas.

TABEL 9.  
WAKTU PELATIHAN DAN INFERENSI MELALUI PENDEKATAN TERINTEGERASI

| Model                   | Training Time | Inference Time | Rata - rata |
|-------------------------|---------------|----------------|-------------|
| Naïve Bayes + TAN + GS  | 0.603024      | 0.131080       | 0.00114982  |
| Naïve Bayes + TAN + GA  | 0.623435      | 0.161162       | 0.00141370  |
| Naïve Bayes + TAN + ACO | 0.576025      | 0.127067       | 0.00111462  |

Dari sisi efisiensi waktu pada Tabel 9. Hasil pengujian menunjukkan bahwa ketiga model memiliki beban komputasi yang relatif tinggi karena mengombinasikan struktur TAN dengan metode optimasi. Di antara ketiganya, *Naïve Bayes* + TAN + ACO menunjukkan efisiensi waktu terbaik, baik pada tahap pelatihan maupun inferensi, sedangkan *Naïve Bayes* + TAN + GA memiliki beban komputasi paling tinggi. Sementara itu, *Naïve Bayes* + TAN + GS berada pada tingkat efisiensi menengah. Temuan ini menunjukkan bahwa penggunaan metode optimasi yang berbeda pada model berbasis TAN memberikan pengaruh terhadap kompleksitas komputasi, sehingga efisiensi waktu perlu menjadi pertimbangan dalam pemilihan model.

TABEL 10.  
HASIL UJI WILCOXON PADA METRIK PRECISION, RECALL, DAN F1-SCORE

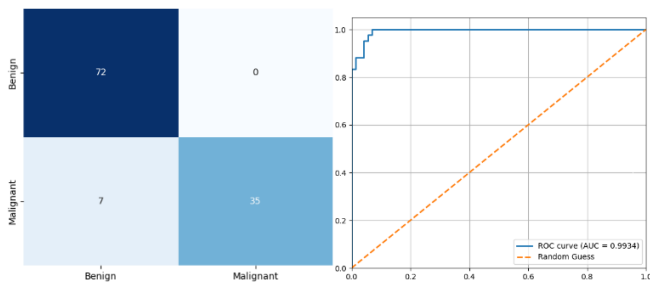
| Metrik    | Statistik Wilcoxon | p-Value | Keputusan        |
|-----------|--------------------|---------|------------------|
| Precision | 7.0000             | 1.0000  | Tidak Signifikan |
| Recall    | 4.0000             | 0.4375  | Tidak Signifikan |
| F1-Score  | 0.0000             | 0.0625  | Tidak Signifikan |

Dari hasil pengujian pada Tabel 10. Hasil uji *Wilcoxon* menunjukkan bahwa nilai *p-value* pada metrik *Precision*, *Recall*, dan *F1-Score* seluruhnya lebih besar dari 0.05. Hal ini menunjukkan bahwa tidak terdapat perbedaan yang signifikan secara statistik antara model NB + TAN + ACO dan NB + TAN + *Grid Search*. Dengan demikian, kedua model dapat dinyatakan memiliki performa yang setara secara statistik

#### G. Visualisasi Model

##### 1. Baseline Model

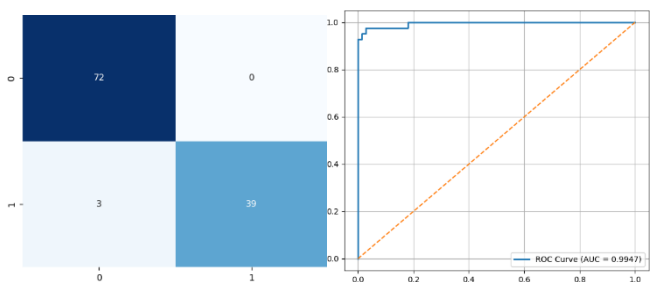
*Baseline Naïve Bayes* dievaluasi menggunakan *Confusion Matrix* dan *Curve ROC* untuk menggambarkan distribusi prediksi serta kemampuan diskriminasi model. Nilai AUC digunakan sebagai indikator performa agregat dalam membedakan kelas, sebagaimana ditunjukkan pada Gambar 3.



Gambar 3. *Confusion Matrix Curve ROC Baseline Model Naïve Bayes*

Hasil *Confusion Matrix* menunjukkan 72 *True Negative*, 35 *True Positive*, tanpa *False Positive*, serta 7 *False Negative*. Nilai AUC sebesar 0,9934 mengindikasikan kemampuan diskriminasi yang sangat baik. Namun, keberadaan *False Negative* pada kelas *Malignant* menunjukkan bahwa sensitivitas model belum optimal.

Selanjutnya hal yang sama dilakukan pada *baseline Support Vector Machine (SVM)* visualisasi Baseline model SVM dapat di lihat pada Gambar 4.

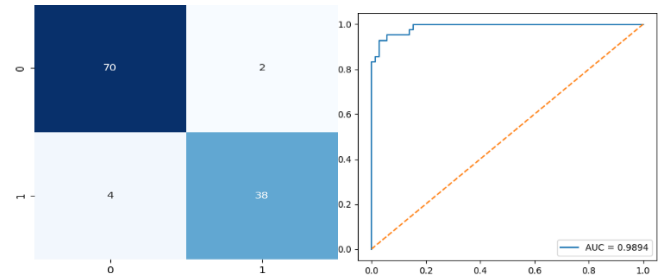


Gambar 4. *Confusion Matrix Curve ROC Baseline Model SVM*

Model menunjukkan performa yang sangat baik dengan tidak adanya *False Positive* (spesifisitas maksimum) dan hanya 3 *False Negative*, sehingga kemampuan deteksi terhadap kasus *malignant* tergolong tinggi. Nilai AUC sebesar 0,9947 mengindikasikan kemampuan diskriminasi yang sangat kuat. *Curve ROC* yang mendekati titik ideal (0,1) menunjukkan bahwa model memiliki pemisahan kelas yang efektif. Secara komparatif, SVM menunjukkan performa yang sedikit lebih unggul dibandingkan *Naïve Bayes*

2. Peningkatan Kinerja Algoritma *Naïve Bayes* Melalui Pendekatan Tunggal

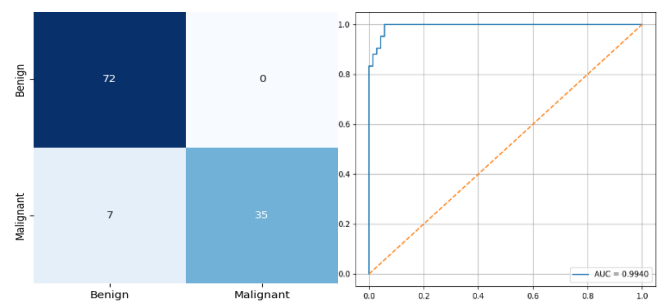
Pendekatan tunggal *Naïve Bayes* berbasis TAN divisualisasikan melalui *Confusion Matrix* untuk menunjukkan hasil klasifikasi, serta *Curve ROC* untuk menilai pemisahan kelas. Nilai AUC merepresentasikan performa model secara keseluruhan, sebagaimana terlihat pada Gambar 5.



Gambar 5. *Confusion Matrix Curve ROC Model Naïve Bayes dengan TAN*

Hasil *Confusion Matrix* menunjukkan 70 *True Negative*, 38 *True Positive*, 2 *False Positive*, dan 4 *False Negative*. Nilai AUC sebesar 0,9894 mengindikasikan kemampuan diskriminasi yang tinggi. Namun, keberadaan *False Positive* dan *False Negative* menunjukkan bahwa model masih mengalami misklasifikasi, yang mengindikasikan adanya tumpang tindih distribusi probabilitas antar kelas.

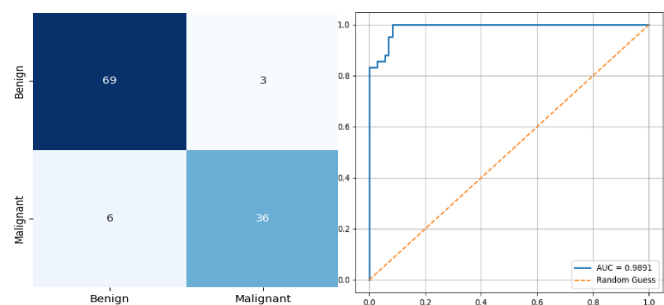
Hal yang sama di lakukan pada visualisasi pendekatan Tunggal *Naive Bayes* dengan *Ant Colony* yang dapat di lihat pada gambar 6



Gambar 6. *Confusion Matrix Curve ROC Model Naïve Bayes dengan ACO*

Hasil *Confusion Matrix* menunjukkan 72 *True Negative*, 35 *True Positive*, tanpa *False Positive*, serta 7 *False Negative*. Nilai AUC sebesar 0,9940 menunjukkan kemampuan diskriminasi yang sangat tinggi. Namun, keberadaan *False Negative* mengindikasikan bahwa sensitivitas model masih belum optimal dalam mendeteksi seluruh kasus positif.

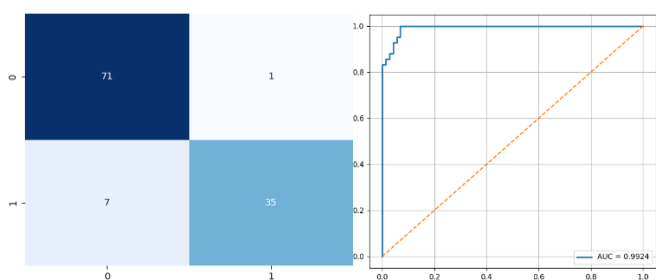
Selanjutnya visualisasi pendekatan *Naïve Bayes* dengan *Genetic Algorithm* dapat di lohat pada Gambar 7.



Gambar 7. *Confusion Matrix Curve ROC Model Naïve Bayes dengan GA*

Hasil *Confusion Matrix* menunjukkan 69 *True Negative*, 36 *True Positive*, 3 *False Positive*, dan 6 *False Negative*. Nilai AUC sebesar 0,9891 mengindikasikan kemampuan diskriminasi yang tinggi. Meskipun demikian, keberadaan misklasifikasi menunjukkan adanya overlap distribusi probabilitas antara kelas *benign* dan *malignant*, yang berpotensi memengaruhi *Accuracy* prediksi pada kondisi tertentu.

Kemudian visualisasi *Naïve Bayes* dengan *Grid Search* dapat di lihat juga pada Gambar 8.



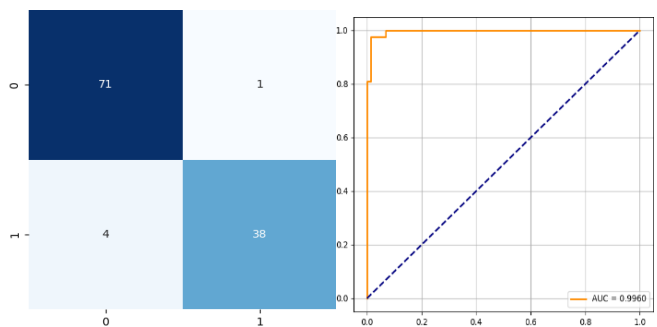
Gambar 8. *Confusion Matrix Curve ROC Model Naïve Bayes dengan Grid Search*

Hasil *Confusion Matrix* menunjukkan 71 *True Negative*, 35 *True Positive*, 1 *False Positive*, dan 7 *False Negative*. Nilai AUC sebesar 0,9924 menunjukkan kemampuan diskriminasi yang sangat kuat. Namun, keberadaan *False Negative* pada kelas *malignant* mengindikasikan bahwa sensitivitas model masih belum optimal, yang mencerminkan adanya tumpang tindih distribusi probabilitas antar kelas.

### 3. Peningkatan Algoritma *Naïve Bayes* Melalui Pendekatan Terintegrasi

Sama halnya dengan *baseline Model Naïve Bayes* dan pendekatan Tunggal *Naïve Bayes*, pendekatan Terintegrasi di visualisasikan dengan melalui *Confusion Matrix* dan *Curve ROC*.

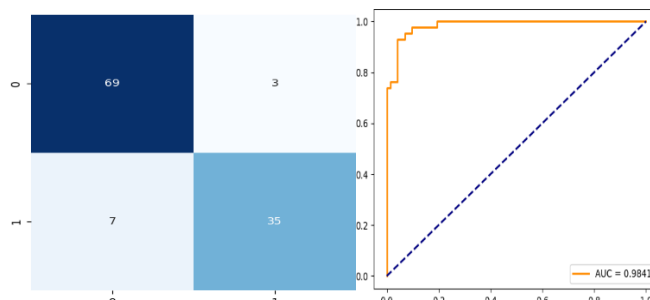
Pada Gambar 9, di jelaskan visualisasi pendekatan terintegrasi Model *Naïve Bayes* dengan TAN dan *Grid Search*.



Gambar 9. *Confusion Matrix Curve ROC Model Naïve Bayes dengan TAN dan Grid Search*

Hasil *Confusion Matrix* menunjukkan 71 *True Negative*, 38 *True Positive*, 1 *False Positive*, dan 4 *False Negative*. Nilai AUC sebesar 0,9960 menunjukkan kemampuan diskriminasi yang sangat tinggi. Meskipun demikian, keberadaan *False Negative* mengindikasikan adanya tumpang tindih distribusi probabilitas antar kelas.

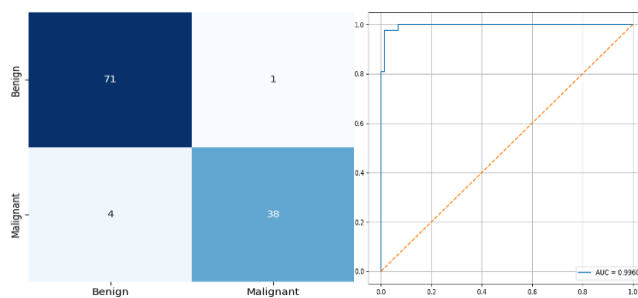
Selanjutnya visualisasi *Naïve Bayes* dengan TAN dan *Genetic Algorithm* dapat di lihat pada Gambar 10.



Gambar 10. *Confusion Matrix Curve ROC Model Naïve Bayes dengan TAN dan GA*

Hasil *Confusion Matrix* menunjukkan 69 *True Negative* dan 35 *True Positive*, dengan 3 *False Positive* serta 7 *False Negative*. Nilai AUC sebesar 0,9841 mengindikasikan kemampuan diskriminasi model yang sangat kuat dalam membedakan kelas positif dan negatif secara agregat. Namun, bentuk *Curve ROC* yang belum sepenuhnya mendekati titik ideal (0,1) menunjukkan masih adanya misklasifikasi pada *threshold* operasional. Hal ini konsisten dengan keberadaan *False Positive* dan *False Negative* pada *Confusion Matrix*. Secara khusus, keberadaan *False Negative* mengindikasikan adanya *overlap* distribusi probabilitas antar kelas.

Kemudian visualisasi *Naive Bayes* dengan TAN dan *Ant Colony* dapat di lihat pada Gambar 11.



Gambar 11. *Confusion Matrix Curve ROC Model Naïve Bayes dengan TAN dan ACO*

Hasil *Confusion Matrix* menunjukkan 71 *True Negative* dan 38 *True Positive*, dengan 1 *False Positive* serta 4 *False Negative*. Nilai AUC sebesar 0,9960 mengindikasikan kemampuan diskriminasi model yang sangat tinggi dalam membedakan kelas secara agregat. Nilai *False Positive* yang

rendah mencerminkan spesifisitas yang kuat, sementara keberadaan *False Negative* menunjukkan masih adanya kesalahan deteksi pada kelas positif. Pola kurva ROC yang belum sepenuhnya mencapai titik ideal mengindikasikan adanya *overlap* distribusi probabilitas antar kelas, meskipun dalam tingkat yang relatif kecil. Integrasi *Tree Augmented Naïve Bayes* (TAN) dan *Ant Colony Optimization* (ACO) berkontribusi dalam meningkatkan performa model.

#### H. Diskusi

Pada pendekatan tunggal, seluruh model menunjukkan kemampuan diskriminasi yang sangat baik dengan AUC 0.9891 – 0.9940. Pendekatan ACO menghasilkan spesifisitas maksimum (*tanpa False Positive*), sedangkan TAN memberikan sensitivitas terbaik dengan jumlah *False Negative* terendah (4 kasus). Namun, masih adanya *False Negative* pada semua model mengindikasikan adanya tumpang tindih distribusi probabilitas antar kelas.

Pada pendekatan terintegrasi, kombinasi TAN + *Grid Search* dan TAN + ACO menghasilkan performa terbaik dengan AUC 0.9960, lebih tinggi dibandingkan TAN + GA (0.9841). Kedua model tersebut juga menunjukkan kesalahan klasifikasi yang lebih rendah, khususnya pada *False Negative*, dibandingkan pendekatan GA. Hal ini menunjukkan bahwa strategi optimasi parameter berpengaruh signifikan terhadap peningkatan kinerja model. Meskipun nilai AUC yang diperoleh sangat tinggi, potensi *overfitting* diminimalkan melalui penggunaan *5-Fold Stratified Cross-Validation* serta pemisahan data latih dan data uji, sehingga performa yang dihasilkan tetap mencerminkan kemampuan generalisasi model.

Secara keseluruhan, hasil penelitian ini menegaskan bahwa penerapan *Tree Augmented Naïve Bayes* (TAN) mampu meningkatkan performa *Naïve Bayes* dengan memodelkan dependensi antar fitur. Lebih lanjut, peningkatan tersebut menjadi lebih optimal ketika dikombinasikan dengan metode optimasi, sehingga memberikan kinerja yang lebih baik dibandingkan pendekatan tunggal.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan, di antaranya penggunaan satu dataset serta proses *diskretisasi* yang berpotensi mengurangi informasi kontinu. Oleh karena itu, penelitian selanjutnya disarankan untuk menguji model pada dataset yang lebih beragam serta mengeksplorasi metode yang lebih adaptif guna meningkatkan kemampuan generalisasi model, khususnya dalam konteks sistem pendukung keputusan medis. Hal ini penting mengingat dalam konteks medis, kesalahan klasifikasi seperti *False Negative* dapat berdampak signifikan terhadap keterlambatan diagnosis

TABEL 11  
PENINGKATAN PERFORMA NAÏVE BAYES DENGAN TAN

| No | Tahun           | Temuan   |
|----|-----------------|--|
| 1  | (2022)<br>[15]. | Menjadi model terbaik dari semua model dengan nilai <i>Accuracy</i> 74.09±3.62, <i>Precision</i> 73.89±2.54, <i>Recall</i> 76.88±2.34, <i>F1-Score</i> 75.14±3.24, AUC 0.59±0.08 |

|   |                 |   |
|---|-----------------|---|
| 2 | (2022)<br>[16]. | Menghasilkan nilai <i>Accuracy</i> 91.5%, itu artinya model <i>Tree Augmented Naïve Bayes</i> (TAN) sangat efektif dalam memprediksi tingkat keparahan dalam kecelakaan keluar jalur di jalan korea selatan |
|---|-----------------|---|

#### I. Pembahasan

Berdasarkan hasil pengujian, kombinasi optimasi *hyperparameter* dan perbaikan struktur independensi menghasilkan beberapa temuan utama. Temuan pertama menunjukkan bahwa *integrasi Naïve Bayes dengan Tree Augmented Naïve Bayes* (TAN) mampu meningkatkan kinerja model. Hal ini ditunjukkan oleh peningkatan *Accuracy* dari 0.9386 menjadi 0.9474, disertai kenaikan *Recall* dan *F1-Score*, meskipun peningkatannya relatif moderat. Peningkatan ini mengindikasikan bahwa pemodelan dependensi antar fitur pada TAN berhasil mengurangi keterbatasan asumsi independensi pada *Naïve Bayes*.

Temuan kedua menunjukkan bahwa optimasi *hyperparameter* menggunakan *Ant Colony Optimization* (ACO), *Grid Search*, dan *Genetic Algorithm* (GA) hanya memberikan peningkatan kinerja yang terbatas, bahkan pada GA terjadi penurunan *Accuracy* dibandingkan model *baseline*. Hal ini disebabkan optimasi hanya berfokus pada parameter *var\_smoothing*, sehingga perbaikannya bersifat numerik dan tidak menyentuh permasalahan utama, yaitu asumsi independensi antar fitur.

Temuan ketiga menunjukkan bahwa kombinasi *Naïve Bayes*, TAN, dan optimasi lanjutan menghasilkan kinerja terbaik, khususnya pada skema TAN + ACO dan TAN + *Grid Search*, dengan *Accuracy* mencapai 0.9561 dan AUC sebesar 0.9960. Hasil ini menunjukkan bahwa peningkatan performa yang signifikan lebih dipengaruhi oleh perbaikan struktur model dibandingkan optimasi parameter semata, dan kombinasi keduanya memberikan hasil yang paling optimal.

Temuan keempat menunjukkan adanya perbedaan performa antara *K-Fold Cross Validation* dan pengujian pada data uji, yang merupakan fenomena umum dalam evaluasi model *machine learning*. *Cross Validation* memberikan estimasi yang lebih stabil karena proses pelatihan dan validasi dilakukan secara berulang, sedangkan data uji merepresentasikan kemampuan generalisasi model terhadap data yang benar-benar baru.

Dari sisi komputasi, model *Naïve Bayes* dengan optimasi *hyperparameter* memiliki waktu pelatihan dan inferensi yang sangat efisien. Sebaliknya, penggunaan TAN meningkatkan kompleksitas komputasi akibat proses pembentukan struktur dependensi antar fitur. Kombinasi TAN dengan metode optimasi lebih lanjut meningkatkan waktu komputasi, baik pada tahap pelatihan maupun *inferensi*. Dengan demikian, terdapat *trade-off* antara kinerja dan efisiensi, di mana model dengan struktur yang lebih kompleks dan optimasi lanjutan memberikan performa yang lebih tinggi, namun dengan biaya komputasi yang lebih besar.

#### IV. KESIMPULAN

Studi ini menyimpulkan bahwa dari berbagai pengujian yang dilakukan, *Tree Augmented Naive Bayes* (TAN) memiliki peran yang signifikan dalam meningkatkan kinerja algoritma klasifikasi *Naive Bayes*. Hal ini disebabkan karena TAN mampu memperbaiki kelemahan utama *Naive Bayes* yang mengasumsikan independensi antar fitur. Sebaliknya, optimasi *hyperparameter* tidak mampu memperbaiki keterbatasan struktural tersebut, sehingga peningkatan kinerja yang dihasilkan relatif terbatas.

Dari sisi komputasi, penerapan TAN dan kombinasi dengan metode optimasi menyebabkan peningkatan waktu pelatihan dan inferensi dibandingkan dengan *Naive Bayes* tanpa struktur dependensi. Meskipun demikian, peningkatan kompleksitas tersebut sebanding dengan peningkatan performa yang diperoleh.

Dengan demikian, dapat disimpulkan bahwa perbaikan dependensi fitur melalui TAN lebih efektif dalam meningkatkan performa algoritma *Naive Bayes* dibandingkan dengan optimasi *hyperparameter*, dengan konsekuensi peningkatan biaya komputasi.

#### DAFTAR PUSTAKA

- [1] W. Ningsih, B. Alfiana, and D. Wulandari, "Comparison of Naive Bayes and SVM algorithms in Twitter sentiment analysis on electric car use in Indonesia," *Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 556–562, 2024, doi: 10.57152/malcom.v4i2.1253.
- [2] G. S. Al-Husna, D. Asmarajati, I. A. Ihsanuddin, and R. Mahmudati, "Perbandingan metode Naive Bayes dan Support Vector Machine untuk analisis sentimen pada ulasan pengguna Aplikasi LinkedIn," *Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 3, no. 2, pp. 139–144, 2024, doi: 10.55123/storage.v3i2.3602.
- [3] B. Shafa, H. H. Handayani, S. Arum, and P. Lestari, "Prediksi kanker paru dengan normalisasi menggunakan perbandingan algoritma Random Forest, Decision Tree, dan Naive Bayes," *Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 3, pp. 1057–1070, 2024, doi: 10.51454/decode.v4i3.779.
- [4] N. A. Prakoso Indaryono, "Analisa perbandingan algoritma Random Forest dan Naive Bayes untuk klasifikasi curah hujan berdasarkan iklim di Indonesia," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 9, no. 1, pp. 158–167, 2024, doi: 10.29100/jipi.v9i1.4421.
- [5] B. Ramadhani and R. R. Suryono, "Komparasi algoritma Naive Bayes dan Logistic Regression untuk analisis sentimen metaverse," *Jurnal Media Informatika Budidarma*, vol. 8, no. 2, pp. 714–720, 2024, doi: 10.30865/mib.v8i2.7458.
- [6] I. R. Hendrawan, E. Utami, and A. D. Hartanto, "Comparison of Naive Bayes algorithm and XGBoost on local product review text classification," *Edumatic: Jurnal Pendidikan Informatika*, vol. 6, no. 1, pp. 143–149, 2022, doi: 10.29408/edumatic.v6i1.5613.
- [7] B. Pramuditya and A. Prabowo, "Optimasi fuzzy logic menggunakan genetic algorithm dalam menentukan program diet dan bulking," *Jurnal Pengembangan IT*, vol. 10, no. 4, pp. 1044–1058, 2025, doi: 10.30591/jpit.v10i4.9459.
- [8] M. Fawzan and D. Udjulawa, "Optimasi hyperparameter CNN dengan arsitektur VGG16 menggunakan Grid Search untuk klasifikasi penyakit buah delima," *Journal of Computer Science and Artificial Intelligence*, vol. 5, no. 2, pp. 306–331, 2025, doi: 10.29240/arcitech.v5i2.15175.
- [9] F. Suryana, N. Nurdin, and D. Hamdhana, "Implementation of Ant Colony Optimization algorithm for route optimization of tourist paths in Takengon," *Journal of Applied Informatics and Computing*, vol. 9, no. 4, pp. 1886–1896, Aug. 2025, doi: 10.30871/jaic.v9i4.9706.
- [10] A. Alwi, I. Iskandar, and D. Setyanto, "The philosophy of Naive Bayes and its comparison with Tree Augmented Naive Bayes," *Saudi Journal of Engineering and Technology*, vol. 7, no. 7, pp. 377–385, 2022, doi: 10.36348/sjet.2022.v07i07.005.
- [11] M. Agustriya, M. Ula, and K. Agustini, "Analisis kinerja algoritma klasifikasi Naive Bayes menggunakan genetic algorithm dan bagging untuk Data Publik Risiko Transaksi Kartu Kredit," *Jurnal Sistem dan Teknologi Informasi*, vol. 12, no. 3, pp. 584–591, 2024, doi: 10.26418/justin.v12i3.80136.
- [12] E. Y. Hidayat et al., "Genetic algorithm-based convolutional neural network feature engineering for optimizing coronary heart disease prediction," *Healthcare Informatics Research*, vol. 30, no. 3, pp. 234–243, 2024, doi: 10.42558/hir.2024.30.3.234.
- [13] S. L. Pamungkas, R. O. S. Gurning, D. W. Handani, and A. Hafizh, "Tree Augmented Naive Bayesian network application on multiplicative premium equation," *International Journal of Safety and Security Engineering*, vol. 15, no. 12, pp. 2625–2638, 2025, doi: 10.18280/ijss.151218.
- [14] F. R. Dastjerdi and L. Cai, "Augmenting Naive Bayes classifiers with k-tree topology," *Mathematics*, vol. 13, no. 13, pp. 1–16, 2025, doi: 10.3390/math13132185.
- [15] G. A. Ruz, P. Araya-Díaz, and P. A. Henríquez, "Facial biotype classification using tree augmented Naive Bayes," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, pp. 1–10, 2022, doi: 10.1186/s12911-022-02062-7.
- [16] H. Kim et al., "Prediction of run-off road crash severity through Tree Augmented Naive Bayes learning," *Applied Sciences*, vol. 12, no. 3, 2022, doi: 10.3390/app12031120.
- [17] A. D. Rachmatsyah, T. Sugihartono, and K. Irfan, "Perbandingan teknik optimasi Grid Search dan Randomized Search pada klasifikasi SVM," *Sistem Komputer dan Teknik Informatika*, vol. 8, no. 1, pp. 13–22, 2024, doi: 10.36080/skanika.v8i1.3328.
- [18] A. Nadroh et al., "Klasifikasi status gizi balita menggunakan SVM dengan optimasi Grid Search Cross-Validation," *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 8, no. 2, pp. 250–257, 2024, doi: 10.46880/jmika.vol8no2.pp250-257.
- [19] M. D. Wardana et al., "Implementation of Ant Colony Optimization in obesity level classification using Random Forest," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 5, pp. 3543–3557, 2025, doi: 10.52436/1.jutif.2025.6.5.4696.
- [20] F. Y. Santoso, E. Sedyono, and H. D. Purnomo, "Optimalisasi hyperparameter convolutional neural networks menggunakan Ant Colony Optimization," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, pp. 243–248, 2024, doi: 10.25126/jtiik.20241127105.
- [21] M. M. Hossin et al., "Breast cancer detection using machine learning algorithms on the Wisconsin dataset," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2446–2456, 2023, doi: 10.11591/eei.v12i4.4448.
- [22] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan metode Naive Bayes dan SVM pada analisis sentimen Twitter," *Smatika Journal*, vol. 10, no. 2, pp. 71–76, 2020, doi: 10.32664/smatika.v10i02.455.
- [23] A. Putri et al., "Comparison of K-NN, Naive Bayes and SVM algorithms for graduation prediction," *Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 1, pp. 20–26, 2023, doi: 10.57152/malcom.v3i1.610.
- [24] M. R. Zuhri, K. Kusri, and D. Ariatanto, "Analisis perbandingan algoritma klasifikasi untuk identifikasi diabetes," *Jurnal Informatika Teknologi dan Sains*, vol. 7, no. 1, pp. 11–20, 2025, doi: 10.51401/jinteks.v7i1.5146.
- [25] I. H. Kusuma and N. Cahyono, "Analisis sentimen masyarakat terhadap penggunaan e-commerce menggunakan KNN," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 3, pp. 302–307, 2023, doi: 10.30591/jpit.v8i3.5734.
- [26] S. Sathyanarayanan, "Confusion matrix-based performance evaluation metrics," *African Journal of Biomedical Research*, vol. 27, no. 4, pp. 4023–4031, 2024, doi: 10.53555/ajbr.v27i4s.4345.
- [27] L. Lavazza, S. Morasca, and G. Rotoloni, "Software defect prediction evaluation: New metrics based on the ROC curve," *Information and Software Technology*, vol. 187, p. 107865, 2025, doi: 10.1016/j.infsof.2025.107865.