Evaluasi Optimasi Hyperparameter Gaussian Naïve Bayes Menggunakan Algoritma Genetika untuk Klasifikasi Penyakit Hati

Nurjanna[1], Taghfirul Azhima Yoga Siswa*[2], Muhammad Ridwan[3]
Teknik Informarika, Universitas Muhammadiyah Kalimantan Timur
email: [1]2211102441214@umkt.ac.id , [2]tay758@umkt.ac.id, [3]mr624@umkt.ac.id
*Penulis Korespondensi

Diterima: 1 Agustus 2026; Direvisi: 18 Oktober 2026; Disetujui: 22 Oktober 2026

Abstrak

Hati memerlukan deteksi dini untuk mendukung penanganan yang tepat. Penelitian ini bertujuan mengevaluasi pengaruh optimasi hyperparameter menggunakan Genetic Algorithm (GA) terhadap performa Gaussian Naïve Bayes (GNB) dalam klasifikasi penyakit hati. Dataset yang digunakan adalah Indian Liver Patient Dataset (ILPD) dari UCI Machine Learning Repository, yang terdiri atas 570 sampel setelah prapemrosesan. Optimasi dilakukan terhadap parameter var_smoothing dalam rentang 10^{-11} *hingga* 10^{-7} *dengan accuracy sebagai fungsi fitness. GA menggunakan populasi 10 individu dan lima generasi, sedangkan evaluasi model dilakukan menggunakan 10-fold cross-validation. Hasil evaluasi menunjukkan bahwa GNB tanpa optimasi memperoleh rata-rata accuracy sebesar 56,14%, precision sebesar 96,60%, recall sebesar 39,93%, dan F1-score sebesar 56,04%. Setelah optimasi GA, rata-rata accuracy meningkat menjadi 56,32%, precision tetap 96,60%, recall menjadi 40,18%, dan F1-score menjadi 56,31%. Peningkatan tersebut relatif kecil, sedangkan tingginya jumlah false negative menunjukkan keterbatasan model dalam mengidentifikasi sampel penyakit hati. Hasil penelitian menunjukkan bahwa GA tidak meningkatkan performa GNB secara berarti pada konfigurasi dan rentang pencarian yang dievaluasi.*

Kata kunci: *gaussian naïve bayes, genetic algorithm, klasifikasi, penyakit hati, ilpd.*

Abstract

Liver disease requires early detection to support appropriate treatment. This study evaluates the effect of Genetic Algorithm (GA)-based hyperparameter optimization on Gaussian Naïve Bayes (GNB) performance in liver disease classification. The dataset was the Indian Liver Patient Dataset (ILPD) from the UCI Machine Learning Repository, comprising 570 samples after preprocessing. Optimization was performed on the var_smoothing parameter within the range of 10^{-11} *to* 10^{-7}*, using accuracy as the fitness function. The GA used a population of 10 individuals and five generations, while model evaluation employed 10-fold cross-validation. The baseline GNB achieved mean accuracy of 56.14%, precision of 96.60%, recall of 39.93%, and F1-score of 56.04%. After GA optimization, mean accuracy increased to 56.32%, precision remained at 96.60%, recall reached 40.18%, and F1-score reached 56.31%. These improvements were small, while the high number of false negatives indicated limitations in identifying liver disease samples. The results show that GA did not improve GNB performance substantially under the evaluated configuration and search range*.

Keywords*: gaussian naïve bayes, genetic algorithm, classification, liver disease, ilpd*.

I. PENDAHULUAN

Hati merupakan organ yang menjalankan berbagai fungsi penting dalam tubuh, termasuk metabolisme zat gizi, produksi senyawa yang dibutuhkan tubuh, dan proses detoksifikasi [1]. Gangguan pada organ ini dapat memengaruhi fungsi fisiologis dan menimbulkan berbagai masalah kesehatan [2]. Identifikasi penyakit hati sejak dini menjadi penting karena keterlambatan diagnosis dapat meningkatkan risiko komplikasi dan mempersulit penanganan [3]. Oleh sebab itu, diperlukan pendekatan yang dapat membantu proses identifikasi kondisi pasien secara tepat berdasarkan informasi kesehatan yang tersedia.

This is an open access article under the CC–BY-SA license.

Pemanfaatan *machine learning* bidang kesehatan memungkinkan analisis data pasien untuk mengidentifikasi pola yang berkaitan dengan kondisi penyakit [4]. Pendekatan *machine learning* juga telah digunakan dalam analisis data medis untuk mendukung proses klasifikasi penyakit [5]. Penerapan metode tersebut dalam klasifikasi penyakit hati telah dikaji dalam penelitian sebelumnya [6]. Penelitian [7] melaporkan bahwa algoritma Random Forest memperoleh akurasi sebesar 70,60%, sedangkan Naïve Bayes memperoleh akurasi sebesar 55,80%. Penelitian lain juga membahas penerapan *machine learning* untuk klasifikasi penyakit hati [9]. Perbedaan hasil penelitian menunjukkan perlunya mengevaluasi algoritma dan konfigurasi model berdasarkan karakteristik dataset yang digunakan.

Gaussian Naïve Bayes (GNB) merupakan salah satu algoritma klasifikasi yang memiliki proses pelatihan relatif sederhana dan kebutuhan komputasi yang relatif rendah [8]. Algoritma ini juga telah digunakan dalam penelitian untuk mendukung klasifikasi berdasarkan probabilitas kondisi yang diamati [9]. Meskipun demikian, GNB memiliki asumsi bahwa setiap fitur bersifat independen secara kondisional terhadap kelas. Pada data medis, hubungan antarfitur dapat memengaruhi kesesuaian asumsi tersebut sehingga berpotensi membatasi performa klasifikasi [10]. Oleh karena itu, performa GNB perlu dievaluasi berdasarkan hasil pengujian empiris pada dataset yang digunakan.

Salah satu pendekatan untuk mengevaluasi konfigurasi model adalah optimasi *hyperparameter*, yaitu proses pencarian nilai parameter yang memberikan hasil evaluasi terbaik dalam ruang pencarian yang ditentukan [11]. Pada Gaussian Naïve Bayes, parameter var_smoothing digunakan untuk menjaga kestabilan perhitungan dengan menambahkan nilai kecil pada variansi fitur. Pengaruh perubahan nilai parameter ini terhadap performa klasifikasi perlu diuji secara empiris karena pencarian nilai yang berbeda tidak selalu menghasilkan peningkatan performa [12]. Dengan demikian, penelitian ini berfokus pada evaluasi pengaruh perubahan var_smoothing terhadap performa GNB, bukan pada pemilihan atau pengurangan jumlah fitur.

Genetic Algorithm (GA) merupakan metode optimasi berbasis evolusi yang menggunakan mekanisme seleksi, *crossover*, dan mutasi untuk mencari kandidat solusi berdasarkan fungsi *fitness* [13]. Penelitian terdahulu juga telah menerapkan GA untuk seleksi fitur dalam prediksi gagal jantung [14]. Namun, seleksi fitur berbeda dari optimasi *hyperparameter*: seleksi fitur menentukan atribut yang digunakan oleh model, sedangkan optimasi *hyperparameter* mencari konfigurasi parameter model. Oleh karena itu, hasil penelitian yang menggunakan GA untuk seleksi fitur tidak dapat langsung dijadikan bukti bahwa GA akan meningkatkan performa GNB melalui optimasi *var_smoothing*. Pengaruh optimasi tersebut perlu dievaluasi secara khusus melalui pengujian yang sesuai.

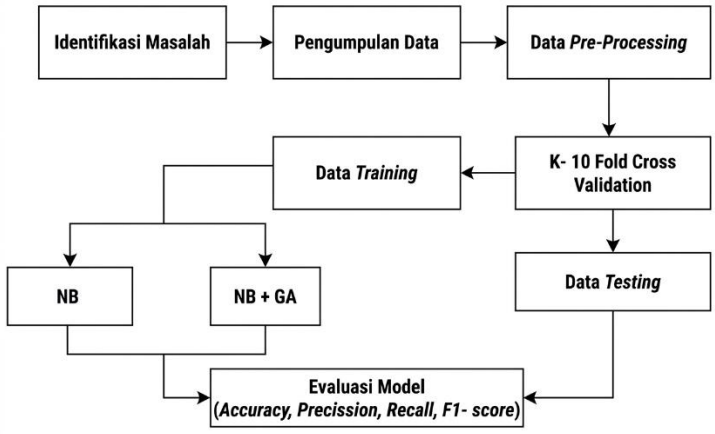
Berdasarkan uraian penelitian terdahulu, penelitian ini bertujuan mengevaluasi pengaruh optimasi *hyperparameter* var_smoothing menggunakan *Genetic Algorithm* (GA) terhadap performa *Gaussian Naïve Bayes* (GNB) dalam klasifikasi penyakit hati menggunakan *Indian Liver Patient Dataset* (ILPD) yang diperoleh dari *UCI Machine Learning Repository* [15]. Berbeda dari penelitian yang berfokus pada seleksi fitur, penelitian ini mengkaji pencarian nilai *hyperparameter* GNB melalui metode *10-fold stratified cross-validation*. Performa GNB tanpa optimasi dibandingkan dengan GNB menggunakan GA, *Grid Search*, dan *Random Search* [16], berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, penelitian ini menganalisis nilai var_smoothing yang dipilih, variasi hasil evaluasi, serta waktu komputasi untuk mengidentifikasi keterbatasan optimasi GA pada konfigurasi dan rentang pencarian yang diuji.

Dataset setelah prapemrosesan terdiri atas 570 sampel, yaitu 406 sampel kelas Liver dan 164 sampel kelas Non-Liver. Distribusi tersebut menunjukkan bahwa dataset memiliki ketidakseimbangan kelas. Sebagai pembanding sederhana, model yang selalu memprediksi seluruh sampel sebagai kelas *Liver* menghasilkan *accuracy* sebesar 71,23%, tetapi tidak mampu mengenali kelas *Non-Liver*. Kondisi ini menunjukkan bahwa *accuracy* saja belum cukup untuk menggambarkan kemampuan klasifikasi secara menyeluruh. Oleh karena itu, penelitian ini mengevaluasi performa model menggunakan *accuracy*, *precision*, *recall*, dan *F1-score* untuk memperoleh gambaran yang lebih lengkap mengenai hasil klasifikas.

II. METODE PENELITIAN

Penelitian ini menggunakan *Indian Liver Patient Dataset* (ILPD) yang diperoleh dari UCI *Machine Learning Repository*. Tahapan penelitian meliputi identifikasi masalah, pengumpulan data, prapemrosesan data, pembagian data menggunakan *10-fold stratified cross-validation*, pembentukan model *Gaussian Naïve Bayes* (GNB), optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA), serta evaluasi performa

model. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, *specificity*, dan *balanced accuracy*. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur penelitian

A. Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk menentukan fokus penelitian, yaitu mengevaluasi performa algoritma *Gaussian Naïve Bayes* (GNB) dalam klasifikasi penyakit hati serta menganalisis pengaruh optimasi *hyperparameter*, khususnya *var_smoothing*, terhadap hasil klasifikasi. Penelitian ini menggunakan *Genetic Algorithm* (GA) untuk mencari kandidat nilai *var_smoothing* berdasarkan fungsi *fitness* yang telah ditentukan. Hasil optimasi kemudian dibandingkan dengan model GNB tanpa optimasi dan metode pencarian *hyperparameter* lainnya untuk menganalisis perbedaan performa klasifikasi dan kebutuhan komputasinya.

B. Pengumpulan Data

Penelitian ini menggunakan *Indian Liver Patient Dataset* (ILPD) yang diperoleh *dari UCI Machine Learning Repository*. Dataset awal terdiri atas 583 sampel dengan 11 kolom, yang mencakup sepuluh atribut fitur dan satu atribut target. Atribut fitur meliputi *Age*, *Gender*, *Total Bilirubin*, *Direct Bilirubin*, *Alkaline Phosphatase*, *Alamine Aminotransferase*, *Aspartate Aminotransferase*, *Total Proteins*, *Albumin*, dan *Albumin and Globulin Ratio*. Sementara itu, atribut *Class* digunakan sebagai target klasifikasi untuk membedakan pasien kelas *Liver* dan *Non-Liver*. Dataset ILPD dipilih karena telah digunakan dalam penelitian klasifikasi penyakit hati sehingga hasil penelitian ini dapat dibahas dalam konteks penelitian terdahulu [6].

C. Data Preprocessing

Tahap *preprocessing* meliputi seleksi data (*data selection*), pembersihan data (*data cleaning*), dan transformasi data (*data transformation*). Pada tahap seleksi, sepuluh atribut digunakan sebagai fitur masukan, sedangkan atribut *Class* digunakan sebagai target klasifikasi. Pada tahap pembersihan, data duplikat dihapus dan nilai yang hilang pada atribut *Albumin and Globulin Ratio* ditangani menggunakan imputasi nilai rata-rata (*mean imputation*). Selanjutnya, pada tahap transformasi, atribut *Gender* dikonversi menjadi nilai numerik, yaitu *Male* = 1 dan *Female* = 0. Atribut target dipetakan menjadi kelas 1 untuk *Liver* dan kelas 0 untuk *Non-Liver*.

Setelah 13 data duplikat dihapus dari 583 data awal, diperoleh 570 sampel yang terdiri atas 406 sampel kelas *Liver* dan 164 sampel kelas *Non-Liver*. Untuk menghindari kebocoran data (*data leakage*), nilai rata-rata untuk imputasi dihitung hanya berdasarkan data pelatihan pada setiap *fold*, kemudian diterapkan pada data pengujian yang bersangkutan. Prosedur yang sama diterapkan pada validasi silang internal, yaitu dengan menghitung nilai imputasi berdasarkan data pelatihan pada setiap pembagian internal. Dengan demikian, informasi dari data pengujian tidak digunakan untuk menentukan nilai imputasi selama proses pelatihan dan optimasi *hyperparameter*.

D. K-Fold Cross Validation

Evaluasi model dilakukan menggunakan metode *10-fold stratified cross-validation* dengan pendekatan StratifiedKFold untuk mempertahankan proporsi kelas pada setiap pembagian data. Metode *cross-validation* digunakan untuk memperkirakan kemampuan generalisasi model, sedangkan stratifikasi bertujuan menjaga distribusi kelas pada setiap *fold* [17]. Dataset dibagi menjadi sepuluh bagian (*fold*). Pada setiap iterasi,

sembilan *fold* digunakan sebagai data pelatihan (*training set*) dan satu *fold* digunakan sebagai data pengujian (*testing set*). Proses ini diulang sebanyak sepuluh kali sehingga setiap *fold* memperoleh giliran sebagai data pengujian.

Pada setiap iterasi, optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA) dilakukan hanya pada data pelatihan melalui validasi silang internal sebanyak lima bagian (*5-fold cross-validation*). Prosedur ini merupakan validasi silang bertingkat (*nested cross-validation*), yaitu validasi silang internal untuk memilih *hyperparameter* dan validasi silang eksternal untuk mengevaluasi performa model pada data pengujian yang tidak digunakan selama proses optimasi. Setelah optimasi selesai, kandidat nilai var_smoothing dengan nilai *fitness* terbaik digunakan untuk membentuk model *Gaussian Naïve Bayes* (GNB). Model tersebut kemudian dievaluasi menggunakan data pengujian pada *fold* eksternal terkait.

E. Gaussian Naïve Bayes

Gaussian Naïve Bayes (GNB) merupakan algoritma klasifikasi berbasis Teorema Bayes yang mengasumsikan bahwa setiap fitur bersifat independen secara kondisional terhadap kelas. Algoritma ini menghitung probabilitas suatu sampel termasuk ke dalam kelas tertentu berdasarkan nilai fitur yang tersedia [18]. Pada implementasi GNB, parameter *var_smoothing* digunakan untuk menambahkan sebagian dari variansi terbesar seluruh fitur ke variansi fitur guna menjaga kestabilan perhitungan. Dalam penelitian ini, var_smoothing diperlakukan sebagai *hyperparameter* yang dioptimasi menggunakan *Genetic Algorithm* (GA). Kandidat nilai parameter dievaluasi berdasarkan nilai *fitness* yang diperoleh melalui validasi silang internal. Kandidat dengan nilai *fitness* terbaik dalam ruang pencarian yang diuji kemudian digunakan untuk membentuk model GNB yang dievaluasi pada data pengujian. Persamaan Teorema Bayes ditunjukkan pada Persamaan (1) [12].

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{1}$$

Keterangan:
P(A │B) : Probabilitas posterior, yaitu probabilitas data termasuk ke dalam kelas A setelah diketahui data B.
P(B │A) : Probabilitas kemunculan data B jika diketahui berada pada kelas A.
P(A) : Probabilitas prior kelas A.
P(B) : Probabilitas kemunculan data B (*evidence*).

F. Optimasi Hyperparameter Menggunakan Genetic Algorithm

Optimasi *hyperparameter* menggunakan GA dilakukan untuk mencari kandidat nilai *var_smoothing* pada algoritma GNB. Pencarian dilakukan dalam rentang 10^{-11} hingga 10^{-7}. Kandidat nilai dievaluasi menggunakan *accuracy* sebagai fungsi *fitness* yang dihitung melalui validasi silang internal sebanyak lima bagian (*5-fold cross-validation*).

GA menggunakan ukuran populasi sebanyak 10 individu dan lima generasi. Proses evolusi menerapkan metode *tournament selection* dengan ukuran turnamen lima individu, probabilitas *crossover* sebesar 0,8, probabilitas mutasi sebesar 0,2, serta mekanisme elitisme untuk mempertahankan individu terbaik pada generasi berikutnya. Nilai *random seed* ditetapkan sebesar 42 untuk mendukung keterulangan eksperimen. Rentang pencarian parameter menggunakan skala logaritmik agar kandidat dengan beberapa orde besaran dapat dievaluasi. Proses evolusi dihentikan setelah mencapai lima generasi sesuai dengan konfigurasi yang ditetapkan. Kandidat dengan nilai *fitness* terbaik yang ditemukan selama proses pencarian dipilih untuk digunakan pada evaluasi model.

Penerapan *Genetic Algorithm* bersama *Gaussian Naïve Baye*s dalam pengembangan model prediksi diabetes telah dibahas pada penelitian sebelumnya [19]. Penelitian tersebut menggabungkan GA dengan seleksi fitur, sedangkan penelitian ini berfokus pada optimasi nilai *var_smoothing*. Dengan demikian, fokus penelitian ini berbeda dari seleksi fitur karena optimasi dilakukan terhadap nilai *hyperparameter* pada model GNB.

Pada setiap *fold* eksternal, proses optimasi dijalankan secara terpisah menggunakan data pelatihan pada *fold* tersebut. Setiap individu merepresentasikan kandidat nilai *var_smoothing* yang dievaluasi berdasarkan nilai *fitness* dari validasi silang internal. Kandidat dengan nilai *fitness* terbaik kemudian digunakan untuk membentuk model GNB yang dievaluasi menggunakan data pengujian pada *fold* eksternal terkait.

Berdasarkan catatan eksperimen, proses GA mencatat 110 evaluasi kandidat pada setiap *fold*. Jumlah tersebut digunakan untuk menggambarkan kebutuhan komputasi selama proses pencarian *hyperparameter*.

G. Evaluasi Model

Evaluasi performa model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, *specificity*, dan *balanced accuracy*. Metrik tersebut digunakan untuk mengukur ketepatan prediksi serta kemampuan model dalam mengklasifikasikan pasien ke dalam kelas *Liver* dan *Non-Liver*. *Accuracy* mengukur proporsi seluruh prediksi yang benar, *precision* mengukur proporsi prediksi positif yang benar, sedangkan *recall* mengukur kemampuan model dalam mengidentifikasi sampel kelas positif. *F1-score* merupakan rata-rata harmonis antara *precision* dan *recall*. Sementara itu, *specificity* mengukur kemampuan model dalam mengidentifikasi kelas negatif, sedangkan *balanced accuracy* merupakan rata-rata *recall* dari kedua kelas sehingga memberikan gambaran performa yang lebih seimbang pada dataset dengan distribusi kelas yang tidak seimbang.

Dalam penelitian ini, kelas *Liver* ditetapkan sebagai kelas positif (*positive class*), sedangkan kelas *Non-Liver* sebagai kelas negatif (*negative class*). Evaluasi dilakukan terhadap prediksi pada data pengujian dalam setiap iterasi *10-fold cross-validation*. Nilai setiap metrik dihitung pada masing-masing *fold*, kemudian dirangkum menggunakan rata-rata dan simpangan baku (*standard deviation*) dari sepuluh *fold*. Selain itu, *confusion matrix* gabungan dari seluruh prediksi data pengujian digunakan untuk menggambarkan jumlah prediksi benar dan salah pada kedua kelas. Perhitungan metrik didasarkan pada nilai *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN), dengan persamaan yang disajikan pada Persamaan (2)–(5).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (4)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (5)$$

Keterangan:

TP (*True Positive*) : jumlah data kelas *liver* yang diprediksi dengan benar sebagai *liver*.
TN (*True Negative*) : jumlah data kelas bukan *liver* yang diprediksi dengan benar sebagai bukan *liver*.
FP (*False Positive*) : jumlah data kelas bukan *liver* yang salah diprediksi sebagai *liver*.
FN (*False Negative*) : jumlah data kelas *liver* yang salah diprediksi sebagai bukan *liver*.

III. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil eksperimen dan evaluasi penerapan optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA) pada algoritma *Gaussian Naïve Bayes* (GNB) untuk klasifikasi penyakit hati. Pembahasan meliputi hasil prapemrosesan data, proses optimasi parameter *var_smoothing*, evaluasi performa model, analisis *confusion matrix*, serta perbandingan hasil optimasi dengan model *baseline* dan metode pencarian *hyperparameter* lainnya. Hasil yang diperoleh selanjutnya dianalisis untuk mengetahui pengaruh optimasi GA terhadap performa klasifikasi dan efisiensi komputasi model.

A. Hasil Data Preprocessing

Prapemrosesan data dilakukan untuk mempersiapkan *Indian Liver Patient Dataset* (ILPD) sebelum digunakan dalam proses klasifikasi penyakit hati. Tahapan prapemrosesan meliputi seleksi data (*data selection*), pembersihan data (*data cleaning*), dan transformasi data (*data transformation*). Pada tahap seleksi, sepuluh atribut digunakan sebagai fitur masukan, sedangkan atribut *Class* digunakan sebagai target klasifikasi. Pada tahap pembersihan, data duplikat dihapus dan nilai yang hilang (*missing values*) ditangani sesuai dengan prosedur prapemrosesan yang diterapkan. Dari 583 sampel awal, sebanyak 13 sampel duplikat dihapus sehingga diperoleh 570 sampel untuk digunakan dalam penelitian.

Pada tahap transformasi, atribut *Gender* dikonversi menjadi nilai numerik, yaitu *Male* = 1 dan *Female* = 0. Atribut target juga dikodekan menjadi dua kelas, yaitu kelas 1 untuk *Liver* dan kelas 0 untuk *Non-Liver*. Setelah proses prapemrosesan, dataset terdiri atas 406 sampel kelas *Liver* dan 164 sampel kelas *Non-Liver*.

Distribusi tersebut menunjukkan bahwa dataset memiliki ketidakseimbangan kelas, dengan kelas *Liver* sebagai kelas mayoritas.

Dataset hasil prapemrosesan selanjutnya digunakan dalam eksperimen klasifikasi menggunakan metode *10-fold cross-validation*. Untuk menghindari kebocoran data (*data leakage*), nilai imputasi dipelajari hanya dari data pelatihan pada setiap *fold*, kemudian diterapkan pada data pengujian yang bersangkutan. Dengan demikian, data pengujian tidak digunakan untuk menentukan nilai imputasi. Dataset hasil prapemrosesan yang ditampilkan dalam Tabel 1 dibawah ini kemudian digunakan untuk mengevaluasi model *Gaussian Naïve Bayes* tanpa optimasi maupun dengan optimasi *hyperparameter.*

Tabel 1. Data setelah *preprocessing*

Age	***Gender***	**TB**	**DB**	**Alkphos**	**Sgpt**	**Sgot**	**Tp**	**ALB**	**A/G Ratio**	***Class*/Target**
26	0	0.09	0.02	154	16	12	7	3.05	1.00	1
29	0	0.09	0.03	202	14	11	6.07	3.06	1.01	1
17	1	0.09	0.03	202	22	19	7.04	4.01	1.02	0
55	1	0.07	0.02	290	53	58	6.08	3.04	1.00	1
57	1	0.06	0.01	210	51	59	5.09	2.07	0.08	1

B. Hasil Optimasi Hyperparameter Menggunakan Genetic Algorithm

Hasil Optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA) dilakukan untuk mencari nilai *var_smoothing* dengan *fitness* terbaik pada algoritma *Gaussian Naïve Bayes* (GNB). Pencarian dilakukan dalam rentang 10^{-11} hingga 10^{-7}, dengan *accuracy* sebagai fungsi *fitness* yang dihitung melalui validasi silang internal sebanyak lima bagian (*5-fold cross-validation*). Proses optimasi dilakukan pada data pelatihan di setiap *fold*, sedangkan data pengujian pada *fold* tersebut tidak digunakan dalam pemilihan parameter.

Pada eksperimen ini, GA menggunakan ukuran populasi sebanyak 10 individu dan lima generasi. Proses pencarian menerapkan metode *tournament selection* dengan ukuran turnamen lima individu, mekanisme elitisme, probabilitas *crossover* sebesar 0,8, dan probabilitas mutasi sebesar 0,2. Nilai *random seed* ditetapkan sebesar 42 untuk mendukung keterulangan eksperimen. Evaluasi kandidat dilakukan menggunakan *Stratified 5-Fold Cross-Validation* pada data pelatihan. Konfigurasi tersebut digunakan untuk menentukan kandidat *var_smoothing* terbaik pada setiap iterasi validasi silang eksternal. Nilai parameter yang terpilih merupakan kandidat dengan nilai *fitness* terbaik dalam ruang pencarian yang ditentukan, bukan jaminan bahwa nilai tersebut merupakan optimum global.

Tabel 2. Hasil optimasi *hyperparameter* menggunakan *genetic algorithm*

Fold	**Best Fitness**	**Optimal *var_smoothing***
1	0,5575	$1{,}00 \times 10^{-9}$
2	0,5575	$1{,}00 \times 10^{-9}$
3	0,5652	$1{,}00 \times 10^{-9}$
4	0,5674	$1{,}00 \times 10^{-9}$
5	0,5809	$1{,}00 \times 10^{-9}$
6	0,5770	$1{,}00 \times 10^{-9}$
7	0,5653	$1{,}00 \times 10^{-9}$
8	0,5634	$1{,}00 \times 10^{-9}$
9	0,5694	$1{,}00 \times 10^{-9}$
10	0,5575	$7{,}18 \times 10^{-8}$

Nilai *Best Fitness* pada *Fold* 1 dan *Fold* 10 sama-sama tercatat sebesar 0,5575 setelah pembulatan empat angka desimal. Namun, nilai *accuracy* validasi silang internal sebelum pembulatan berbeda, yaitu 0,5574528841 pada *Fold* 1 dan 0,5575480678 pada Fold 10. Perbedaan tersebut tidak terlihat pada tabel karena pembulatan angka. Selain itu, kedua *fold* menghasilkan nilai *var_smoothing* yang berbeda, yaitu $1{,}00 \times 10^{-9}$ pada *Fold* 1 dan $7{,}18 \times 10^{-8}$ pada *Fold* 10.

Berdasarkan Tabel 2, nilai *var_smoothing* yang terpilih pada sembilan *fold* adalah $1{,}00 \times 10^{-9}$, sedangkan pada *fold* ke-10 diperoleh nilai $7{,}18 \times 10^{-8}$. Nilai *best fitness* tertinggi diperoleh pada *fold* ke-5, yaitu sebesar 0,5809. Hasil tersebut menunjukkan bahwa GA cenderung memilih nilai *var_smoothing* yang sama pada sebagian besar *fold* dalam rentang pencarian yang ditentukan. Meskipun demikian, nilai *fitness* terbaik pada validasi internal tidak selalu menghasilkan peningkatan performa pada data pengujian eksternal.

Oleh karena itu, efektivitas optimasi GA perlu dinilai berdasarkan metrik evaluasi pada data pengujian, bukan hanya berdasarkan nilai *fitness* yang diperoleh selama proses optimasi.

C. Hasil Evaluasi Model

Evaluasi model dilakukan untuk membandingkan performa algoritma *Gaussian Naïve Bayes* (GNB) tanpa optimasi (*baseline*) dengan GNB yang menggunakan dan *Genetic Algorithm* (GA). Perbandingan dilakukan berdasarkan nilai rata-rata *accuracy*, *precision*, *recall*, dan *F1-score* dari pengujian menggunakan metode 10-*Fold Cross-Validation*. Hasil evaluasi masing-masing model disajikan pada Tabel 3.

Tabel 3. Perbandingan rata-rata hasil evaluasi model

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes (*Baseline*)	56,14%	96,60%	39,93%	56,04%
Naïve Bayes + Grid Search	56,32%	96,60%	40,18%	56,31%
Naïve Bayes + Random Search	56,32%	96,60%	40,18%	56,31%
Naïve Bayes + Genetic Algorithm	56,32%	96,60%	40,18%	56,31%

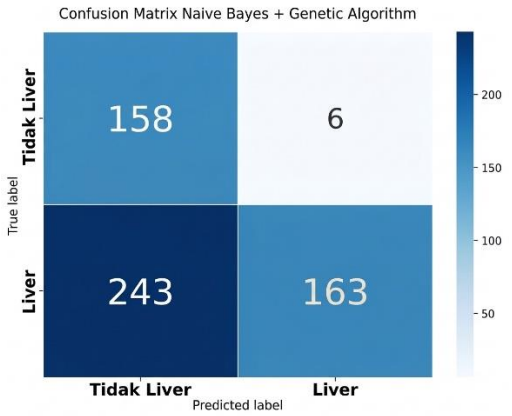
Berdasarkan Tabel 3, model *Gaussian Naïve Bayes* (*baseline*) memperoleh rata-rata *accuracy* sebesar 56,14%, *precision* sebesar 96,60%, *recall* sebesar 39,93%, dan *F1-score* sebesar 56,04%. Sementara itu, model dengan *Grid Search*, *Random Search*, dan *Genetic Algorithm* (GA) menghasilkan nilai rata-rata *accuracy* sebesar 56,32%, *precision* sebesar 96,60%, *recall* sebesar 40,18%, dan *F1-score* sebesar 56,31%. Hasil tersebut menunjukkan bahwa ketiga metode optimasi menghasilkan nilai rata-rata metrik yang sama pada tingkat pembulatan yang ditampilkan.

Dibandingkan dengan model *baseline*, ketiga metode optimasi meningkatkan *accuracy* sebesar 0,18 poin persentase, *recall* sebesar 0,25 poin persentase, dan *F1-score* sebesar 0,27 poin persentase, sedangkan *precision* tidak mengalami perubahan. Peningkatan tersebut relatif kecil sehingga belum menunjukkan perbaikan performa yang substansial. Kesamaan nilai rata-rata metrik juga menunjukkan bahwa GA belum memberikan keunggulan performa yang terukur dibandingkan *Grid Search* dan *Random Search* berdasarkan hasil evaluasi yang disajikan.

Nilai *precision* yang tinggi menunjukkan bahwa sebagian besar sampel yang diprediksi sebagai kelas positif, yaitu *Liver*, memang termasuk dalam kelas tersebut. Namun, nilai *recall* yang relatif rendah menunjukkan bahwa model masih belum mampu mengidentifikasi seluruh sampel *Liver* secara tepat. Dengan demikian, evaluasi model tidak cukup hanya mengandalkan *precision*, tetapi juga perlu mempertimbangkan *recall* dan *F1-score* untuk memperoleh gambaran yang lebih menyeluruh mengenai performa klasifikasi.

Secara keseluruhan, optimasi *hyperparameter* menggunakan GA belum menunjukkan peningkatan performa yang berarti dibandingkan metode optimasi lainnya pada konfigurasi eksperimen ini. Selain itu, GA membutuhkan waktu komputasi lebih lama dibandingkan Grid Search dan Random Search. Oleh karena itu, pemilihan metode optimasi perlu mempertimbangkan tidak hanya nilai metrik evaluasi, tetapi juga efisiensi komputasi yang diperlukan.

D. Confusion Matrix



Gambar 2. *Confusion matrix naïve bayes + genetic algorithm*

Berdasarkan Gambar 2, model Gaussian Naïve Bayes dengan optimasi Genetic Algorithm menghasilkan nilai *True Negative* (TN) sebesar 158, *False Positive* (FP) sebesar 6, *False Negative* (FN)

sebesar 243, dan *True Positive* (TP) sebesar 163. Model berhasil mengklasifikasikan 158 sampel *Non-Liver* dan 163 sampel *Liver* dengan benar. Sebanyak 6 sampel *Non-Liver* salah diklasifikasikan sebagai *Liver*, sedangkan 243 sampel Liver salah diklasifikasikan sebagai *Non-Liver*.

Berdasarkan hasil tersebut, model memperoleh *accuracy* sebesar 56,32%, *precision* sebesar 96,45%, *recall* sebesar 40,15%, *F1-score* sebesar 56,70%, dan specificity sebesar 96,34%. Nilai precision dan specificity yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam menghasilkan prediksi *positif* yang benar dan mengenali sampel *Non-Liver*. Namun, nilai *recall* yang relatif rendah serta tingginya jumlah *false negative* menunjukkan bahwa model masih memiliki keterbatasan dalam mengidentifikasi sampel Liver secara tepat. Oleh karena itu, optimasi model masih perlu dievaluasi lebih lanjut, terutama untuk meningkatkan kemampuan identifikasi sampel *Liver*.

E. Analisis Pengaruh Optimasi Genetic Algorithm

Berdasarkan hasil evaluasi, optimasi *hyperparameter* menggunakan *Genetic Algorithm* (GA) meningkatkan nilai *accuracy* dari 56,14% menjadi 56,32%, *recall* dari 39,93% menjadi 40,18%, dan *F1-score* dari 56,04% menjadi 56,31%. Sementara itu, nilai *precision* tetap sebesar 96,60%. Meskipun terdapat peningkatan pada beberapa metrik, perubahan tersebut relatif kecil. Hasil ini menunjukkan bahwa optimasi parameter *var_smoothing* belum memberikan peningkatan performa klasifikasi yang berarti pada konfigurasi dan rentang pencarian yang digunakan.

Selain performa klasifikasi, waktu komputasi juga menjadi pertimbangan dalam menilai efektivitas metode optimasi. Berdasarkan hasil eksperimen, GA membutuhkan waktu sekitar 5,86 detik, sedangkan *Grid Search* dan Random Search masing-masing membutuhkan sekitar 2,33 detik dan 2,80 detik. Dengan demikian, GA memerlukan waktu komputasi lebih lama tanpa menghasilkan peningkatan metrik evaluasi yang lebih tinggi dibandingkan kedua metode tersebut. Temuan ini menunjukkan bahwa efektivitas optimasi dipengaruhi oleh karakteristik dataset, rentang pencarian *hyperparameter*, dan konfigurasi algoritma. Oleh karena itu, pemilihan metode optimasi perlu mempertimbangkan keseimbangan antara performa klasifikasi dan biaya komputasi.

F. Perbandingan dengan Baseline dan Penelitian Terdahulu

Sebagai pembanding, penelitian ini menggunakan *majority-class baseline*, yaitu strategi yang memprediksi seluruh sampel sebagai kelas *Liver*. Dari 570 sampel, sebanyak 406 merupakan kelas Liver dan 164 merupakan kelas Non-Liver, sehingga diperoleh *accuracy* sebesar 71,23%. Namun, strategi tersebut tidak mampu mengenali kelas Non-Liver, sehingga *balanced accuracy* hanya sebesar 50%. Hasil ini menunjukkan bahwa nilai *accuracy* yang tinggi belum tentu mencerminkan kemampuan model dalam mengenali kedua kelas secara seimbang.

Sementara itu, model *Gaussian Naïve Bayes* dengan optimasi *Genetic Algorithm* (GA) memperoleh rata-rata *accuracy* sebesar 56,32%, *precision* sebesar 96,60%, *recall* sebesar 40,18%, dan *F1-score* sebesar 56,31% berdasarkan rata-rata hasil evaluasi 10-fold pada tabel 3. Berdasarkan *confusion matrix* gabungan, model memperoleh *accuracy* sebesar 56,32%, *recall* sebesar 40,15%, *F1-score* sebesar 56,70%, dan *balanced accuracy* sebesar 68,25%. Perbedaan nilai metrik tersebut terjadi karena satu kelompok dihitung dari rata-rata hasil setiap *fold*, sedangkan kelompok lainnya dihitung berdasarkan gabungan prediksi seluruh sampel.

Perbandingan metode optimasi menunjukkan bahwa GA, *Grid Search*, dan *Random Search* menghasilkan nilai rata-rata metrik yang sama pada tingkat pembulatan yang ditampilkan dalam tabel 3. Namun, GA membutuhkan waktu komputasi sekitar 5,86 detik, lebih lama dibandingkan *Grid Search* sebesar 2,33 detik dan Random Search sebesar 2,80 detik. Selain itu, sembilan *fold* memilih nilai *var_smoothing* sebesar 10^{-9}, yang merupakan nilai *default Gaussian Naïve Bayes* pada scikit-learn. Temuan ini menunjukkan bahwa GA belum memberikan keunggulan performa yang terukur dibandingkan metode optimasi lainnya pada konfigurasi eksperimen yang dievaluasi.

Penelitian ini berfokus pada optimasi *hyperparameter* menggunakan GA, berbeda dari penelitian yang menerapkan GA untuk seleksi fitur. Penelitian selanjutnya dapat mengevaluasi beberapa *random seed*, memperluas atau mengubah rentang pencarian *hyperparameter*, serta membandingkan metode optimasi dengan anggaran evaluasi yang setara untuk mengetahui konsistensi hasil dan peluang peningkatan performa klasifikas.

IV. KESIMPULAN

Penelitian ini mengevaluasi optimasi *hyperparameter* var_smoothing menggunakan *Genetic Algorithm* (GA) pada *Gaussian Naïve Bayes* (GNB) untuk klasifikasi penyakit hati menggunakan ILPD. Hasil evaluasi 10-fold *stratified cross-validation* menunjukkan bahwa *accuracy* meningkat dari 56,14% menjadi 56,32%, *recall* dari 39,93% menjadi 40,18%, dan *F1-score* dari 56,04% menjadi 56,31%, sedangkan *precision* tetap 96,60%. Peningkatan tersebut relatif kecil, dan GA tidak menunjukkan keunggulan performa yang terukur dibandingkan *Grid Search* dan *Random Search.*

Berdasarkan *confusion matrix*, model masih melewatkan 243 dari 406 sampel Liver (59,85%). Selain itu, GA membutuhkan waktu komputasi sekitar 5,86 detik, lebih lama dibandingkan Grid Search dan Random Search. Sembilan dari sepuluh fold juga memilih var_smoothing sebesar $1{,}00 \times 10^{-9}$, yaitu nilai default pada scikit-learn. Dengan demikian, GA tidak meningkatkan performa GNB secara berarti pada konfigurasi dan rentang pencarian yang dievaluasi. Penelitian selanjutnya dapat menguji beberapa *random seed*, rentang pencarian berbeda, dan fungsi *fitness* yang mempertimbangkan *recall* atau *balanced accuracy*.

DAFTAR PUSTAKA

[1] F. L. D. Cahyanti, F. Sarasati, W. Astuti, dan E. Firasari, "Klasifikasi Data Mining Dengan Algoritma Machine Larning Untuk Prediksi Penyakit Liver," *Technol. J. Ilm.*, vol. 14, no. 2, hal. 134, 2023, doi: 10.31602/tji.v14i2.10093.

[2] E. Dritsas dan M. Trigka, "Supervised Machine Learning Models for Liver Disease Risk Prediction," *Computers*, vol. 12, no. 1, hal. 19, 2023, doi: https://doi.org/10.3390/computers12010019.

[3] I. Irmawati, K. Widianto, F. Aziz, A. Rifai, dan A. Rahmawati, "Implementasi Artificial Neural Network Dalam Mendeteksi Penyakit Hati (Liver)," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 6, no. 1, hal. 193–198, 2022, doi: 10.52362/jisamar.v6i1.694.

[4] M. Putri dan Erlin, "Prediksi Penyakit Stroke Menggunakan Machine Learning Dengan Algoritma Random Forest," *J. Infomedia Tek. Inform.*, vol. 9, no. 2, 2024, doi: http://dx.doi.org/10.30811/jim.v9i1.5199.

[5] M. A. Prasetyo, A. K. Zyen, dan R. H. Kusumodestoni, "Optimasi Algoritma Naive Bayes Berbasis Kernel Untuk Klasifikasi Penyakit Hati," *J. Inform. Teknol. dan Sains*, vol. 6, no. 3, hal. 748–756, 2024, doi: https://doi.org/10.51401/jinteks.v6i3.4783.

[6] I. R. Hikmah dan R. N. Yasa, "Perbandingan Hasil Prediksi Diagnosis pada Indian Liver Patient Dataset (ILPD) dengan Teknik Supervised Learning Menggunakan Software Orange," *J. Telemat.*, vol. 16, no. 2, hal. 69–76, 2022, doi: 10.61769/telematika.v16i2.402.

[7] A. I. Lubis, U. Erdiansyah, dan R. Siregar, "Komparasi Akurasi Pada Naïve Bayes Dan Random Forest Dalam Klasifikasi Penyakit Liver," *J. Comput. Eng. Syst. Sci.*, vol. 7, no. January, hal. 81–89, 2022, doi: https://doi.org/10.24114/cess.v7i1.28888.

[8] A. Nurian dan B. N. Sari, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, hal. 829–835, 2023, doi: 10.23960/jitet.v11i3s1.3348.

[9] J. Surbakti, "Implementasi Metode Naïve Bayes Untuk Diagnosa Penyakit Hati," *J. Ilmu Komput. dan Sist. Inf.*, vol. 5, no. 1, hal. 34–40, 2022, doi: https://doi.org/10.55338/jikomsi.v5i1.243.

[10] S. R. Azizah, R. Herteno, A. Farmadi, D. Kartini, dan I. Budiman, "Kombinasi Seleksi Fitur Berbasis Filter dan Wrapper Menggunakan Naive Bayes pada Klasifikasi Penyakit Jantung," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 6, hal. 1361–1368, 2023, doi: 10.25126/jtiik.2023107467.

[11] J. A. Ilemobayo *et al.*, "Hyperparameter Tuning in Machine Learning: A Comprehensive Review," *J. Eng. Res. Reports*, vol. 26, no. 6, hal. 388–395, 2024, doi: 10.9734/jerr/2024/v26i61188.

[12] Lindawati, M. Fadhli, dan A. S. Wardana, "Optimasi Gaussian Naïve Bayes dengan Hyperparameter Tuning dan Univariate Feature Selection dalam Prediksi Cuaca," *J. Pendidik. Inform.*, vol. 7, no. 2, hal. 237–246, 2023, doi: 10.29408/edumatic.v7i2.21179.

[13] R. Salman, Suprapto, dan Irfandi, "Analisis Pengaruh Probabilitas Crossover Terhadap Kinerja Algoritma Genetika Dalam Optimasi Penjadwalan Matakuliah," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, vol. 11, no. 2, hal. 69–74, 2023, doi: 10.21063/jtif.2023.V11.2.69-74.

[14] D. C. P. Buani, "Penerapan Algoritma Naïve Bayes dengan Seleksi Fitur Algoritma Genetika Untuk Prediksi Gagal Jantung," *J. Sains dan Manaj.*, vol. 9, no. 2, hal. 43–48, 2021, doi: 10.31294/evolusi.v9i2.11141.

[15] B. Ramana dan N. Venkateswarlu, "ILPD (Indian Liver Patient Dataset), UCI Machine Learning Repository," 2022. doi: 10.24432/C5D02C.

[16] B. Bischl *et al.*, "Hyperparameter optimization : Foundations , algorithms , best practices , and open challenges,"

WIREs Data Min. Knowl. Discov., vol. 13, no. 2, hal. 1–43, 2023, doi: 10.1002/widm.1484.

[17] S. Widodo, H. Brawijaya, dan Samudi, "Stratified K-fold cross validation optimation on machine learning for prediction," *J. dan Penelit. Tek. Inform.*, vol. 6, no. 4, hal. 2407–2414, 2022, doi: 10.33395/sinkron.v7i4.11792.

[18] K. Anggriani, S. A. Zahra, dan A. Susanto, "Enhancing Malware Detection in IoT Networks using Ensemble Learning on IoT-23 Dataset," *J. Tek. Inform. (JUTIF)*, vol. 6, no. 4, hal. 1985–2000, 2025, doi: 10.52436/1.jutif.2025.6.4.4782.

[19] I. F. Ashari, M. C. Untoro, E. M. Sutowo, D. Salsabila, dan D. A. Zhabiyan, "Hyperparameter Tuning Feature Selection with Genetic Algorithm and Gaussian Naïve Bayes for Diabetes Disease Prediction," *J. Telemat.*, vol. 17, no. 1, hal. 17–23, 2022, doi: 10.61769/telematika.v17i1.488.