 ISSN: 2715-9426

STUDI KOMPARATIF ALGORITMA *MACHINE LEARNING* DALAM KLASIFIKASI DAN PREDIKSI KANKER PARU-PARU

Atma Agilia Triwardani[1)], Erna Daniati[2)], Anita Sari Wardani[3)]
Program Studi Sistem Informasi, Universitas Nusantara PGRI Kediri [1),2),3)]
ernadaniati@unpkediri.ac.id[2]

Abstrak

Penelitian ini bertujuan untuk membandingkan performa lima algoritma klasifikasi machine learning, yaitu Naive Bayes dan Random Forest, dalam mendeteksi kanker paru-paru berdasarkan data gejala klinis pasien. Data penelitian diperoleh dari dataset terbuka Kaggle yang telah melalui tahapan preprocessing, transformasi, dan pembagian data uji dan latih. Evaluasi performa dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score, baik secara otomatis maupun manual. Hasil penelitian menunjukkan bahwa semua algoritma memiliki performa klasifikasi yang baik, namun Naive Bayes unggul dengan akurasi sebesar 93,53% dan F1-score 0,9272. Temuan ini menunjukkan bahwa pendekatan probabilistik sederhana masih sangat relevan dan efektif dalam mengklasifikasikan penyakit berbasis data klinis. Penelitian ini juga memberikan dasar untuk eksplorasi lebih lanjut terkait pengembangan sistem deteksi dini penyakit berbasis machine learning yang lebih akurat dan efisien.

Kata kunci: *machine learning, kanker paru-paru, Naive Bayes, Random Forest, klasifikasi*

1. Pendahuluan

Indonesia saat ini sedang menghadapi tantangan besar di sektor kesehatan akibat melonjaknya prevalensi penyakit tidak menular (PTM). PTM, yang juga dikenal sebagai penyakit degeneratif, menjadi isu utama kesehatan masyarakat pada abad ke-21 karena tingginya angka morbiditas dan mortalitas di tingkat global. Berdasarkan data dari World Health Organization, PTM mencakup penyakit seperti jantung, stroke, tekanan darah tinggi, diabetes, kanker, serta penyakit ginjal kronis. Diduga menyebabkan sekitar 36 juta kematian setiap tahun. dikarenakan kehidupan sehari-hari dan gaya hidup tidak sehat yang meningkatkan risiko munculnya penyakit tidak menular tersebut [1].

Perkembangan teknologi membawa dampak yang signifikan dalam berbagai bidang termasuk bidang kesehatan. Konsep kecerdasan buatan yang meniru kecerdasan manusia untuk memecahkan berbagai masalah adalah *machine learning*. Cara kerja *machine learning* adalah dengan mengembangkan kemampuan pembelajaran mesin untuk menangani kumpulan data tertentu dan kemudian memberikan wawasan yang bermanfaat bagi orang yang menggunakannya. Kecerdasan buatan dan kumpulan data ini dimanfaatkan bidang kesehatan untuk mendeteksi kanker paru-paru pada pasien [2]

Machine learning bagian dari kecerdasan buatan dimana algoritma dan metode tertentu digunakan untuk memprediksi, mengenala pola, dan mengklasifikasi. Analisis data yang dilakukan dikenal sebagai klasifikasi untuk membantu memprediksi label kelas mana yang harus diberikan kepada sampel. Berbagai bidang, seperti statistik, sistem pakar, dan pembelajaran mesin, telah mengusulkan berbagai metode klasifikasi. Algoritma *machine learning* termasuk *Naive Bayes, Random Forest,* dan masih banyak lagi [3].

Berbagai studi sebelumnya yang relevan dijadikan rujukan dalam penelitian ini karena membahas penerapan algoritma klasifikasi *Naive Bayes* dan *Random Forest.* Penelitian pertama membahas pengaruh data tidak seimbang terhadap performa model klasifikasi penyakit diabetes menggunakan algoritma *Random Forest* menggunakan teknik *SMOTE*. Hasil menunjukkan bahwa meskipun kedua skenario mencapai akurasi tinggi (~97-97,5%) [4]. Penelitian kedua membahas pengembangan model prediksi stroke menggunakan *Random Forest* dengan optimasi preprocessing (*KNNImputer* dan *SMOTE-Tomek*). Hasil menunjukkan akurasi hingga 96%, dengan precision, recall, dan F1-score tinggi [5]. Penelitian ketiga membahas perbandingan kinerja model Decision Tree dan Random Forest dalam klasifikasi evaluasi kendaraan menggunakan kriteria pemisahan yang berbeda seperti Information Gain, Gain Ratio, dan Gini Index. Hasil menunjukkan bahwa Random Forest dengan Gain Ratio mencapai akurasi tertinggi sebesar 96,51%, sedangkan Decision Tree dengan kriteria tertentu mencapai sekitar 94,19%, menyoroti dampak signifikan dari pilihan kriteria pemisahan terhadap kinerja model [6]. Penelitian ke empat tersebut menjelaskan sistem pendukung keputusan yang memanfaatkan klasifikasi *Naive Bayes, SVM, C4.5,* dan DBSCAN untuk merekomendasikan program akademik kepada siswa berdasarkan data masukan mereka. Akurasi dari metode Naïve Bayes ini memiliki presisi 100% yang sama dengan *Linear Kernel SVM* sedangkan C4.5 hanya menghasilkan 60% [7]. Penelitian ke lima membahas tentang klasifikasi pengguna video game menggunakan algoritma Naive Bayes, yang bertujuan untuk memahami perilaku pemain serta membantu pengembang dan pemasar. Studi ini mencapai akurasi 97,51% dalam mengkategorikan pemain berdasarkan tingkat antusiasme mereka, yang menunjukkan efektivitas Naive Bayes dalam menganalisis kumpulan data besar dalam industri game [8]

Berdasarkan studi penelitian sebelumnya dapat disimpulkan bahwa dengan teknologi yang semakin berkembang, *machine learning* diharapkan dapat memberikan hasil yang optimal serta dapat diandalkan sebagai solusi dari permasalahan penelitian ini. Hal tersebut memberikan harapan baru dalam mengurangi tingkat kematian akibat penyakit kanker paru-paru. Oleh karena itu, perlu dilakukan studi komparatif secara sistematis dan empiris untuk menganalisis performa *algoritma machine learning* dalam mengklasifikasikan jenis kanker paru-paru berdasarkan metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score*.

2. Kajian Pustaka dan pengembangan hipotesis

2.1.*Naive Bayes*

Naive Bayes adalah salah satu teknik klasifikasi yang didasarkan dari Teorema Bayes yang dikombinasikan dengan *Naive* yang berarti bahwa setiap atribut atau variabel bersifat independen [9]. Untuk menyelesaikan kasus pembelajaran terbimbing, dimana terdapat suatu label, kelas, atau target pada himpunan data yang dijadikan acuan atau pengajar, maka metode *Naive Bayes Classifier* merupakan bagian dari pengklasifikasi statistika yang mampu memprediksi probabilitas partisipan data dalam kelas tertentu yang ditentukan oleh perhitungan probabilitas. [10]. *Naive Bayes* juga mudah digunakan, lebih cepat dalam perhitungan, dan memerlukan sedikit data untuk klasifikasi seperti pada rumus 2.1 [11].

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)} \quad \ldots\ldots\ldots\ldots \quad (2.1)$$

Rumus teorema *Naive Bayes* :

X : Data dengan kelas yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

P(H|X) : Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)

P(H) : Probabilitas hipotesis H (prior prob.)

P(X|H) : Probabilitas X berdasarkan kondisi tersebut

P(X) : Probabilitas dari X [12].

2.2.*Random Forest*

Random Forest Regression / RF merupakan metode klasifikasi berbasis *ensemble method* yang menggunakan beberapa *Decision Tree* dengan karakteristik yang berbeda-beda. Metode *ensemble* sendiri terdiri dari beberapa model berbeda yang dilatih untuk menyelesaikan masalah yang sama dan dikombinasikan untuk mendapatkan hasil terbaik. Pada metode *Random Forest*, setiap pohon keputusan akan menggunakan observasi dan prediktor yang berbeda dari hasil sampling, dan implementasi RF sangat sederhana dan tidak memerlukan banyak perhitungan. Selain itu, implementasi RF mudah dan sangat cepat karena tidak membutuhkan banyak memori, menjadikannya model yang sering digunakan oleh para peneliti untuk mengembangkan *machine learning* [13]. Berikut adalah rumus dari *Random Forest* yang ditunjukkan pada rumus 2.2 [14] :

$$F(x) = \frac{1}{J}\sum_{j}^{J} = 1\ h^{j}(x) \ldots\ldots\ldots \quad (2.2)$$

Keterangan :
F(x) : output dari *Random Forest*
J : jumlah pohon dalam ensemble
$h_j(x)$: output dari pohon ke – (j).

2.3.*Classificication report* dan *confusion matrix*

Confusion Matrix dimanfaatkan untuk menilai performa atau tingkat ketepatan dari proses klasifikasi. Parameter *recall, precision*, dan *accuracy* digunakan untuk mengukur kualitas hasil klasifikasi. *Recall* menggambarkan rasio keberhasilan identifikasi positif yang benar terhadap seluruh data positif yang sebenarnya, sedangkan precision menunjukkan rasio keberhasilan identifikasi positif yang benar terhadap total identifikasi positif yang dihasilkan[15]. Rumus *Confusion Matrix* untuk menghitung *accuracy, precision*, dan *recall* dapat ditunjukkan pada rumus 2.3, 2.4 dan 2.5. [16] :

$$accuracy : \frac{TP+TN}{Total} \ldots\ldots (2.3)$$

$$precision : \frac{TP}{TP+FP} \ldots\ldots (2.4)$$

$$recall : \frac{TP}{TP+FN} \ldots\ldots\ldots (2.5)$$

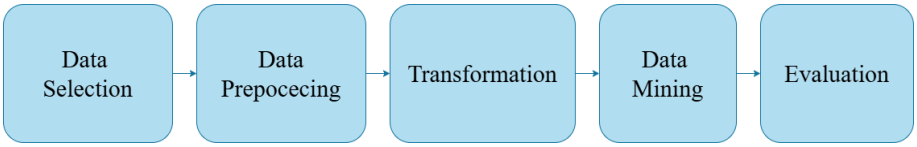
Tabel 2.1 *Confusion Matrix*

		Kelas Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Keterangan :
TP (*True Positive*) : data positif yang diprediksi data positif
FP (*False Positif*) : data negatif tetapi diprediksi sebagai data positif
FN (*False Negatif*) : data positif tetapi diprediksi sebagai data negatif
TN *(True Negative*) : data negatif yang diprediksi data negative

3. Metode Penelitian

Tahap awal dalam penelitian ini adalah mengidentifikasi permasalahan berdasarkan Gambar 3.1 dibawah ini :



Gambar 3.1. Metode penelitian

Proses dimulai dengan *Data Selection* atau mengumpulkan data yang diperoleh berupa data medis pasien kanker paru-paru dari *platform* Kaggle. Setelah itu melakukan *Data Prepocecing* atau pra-pemrosesan data meliputi cek *missing value* atau cek data yang hilang, selanjutnya cek duplikat data menghapus data duplikatnya tetapi mempertahankan data yang pertama. Tahap selanjutnya adalah melabeli data yaitu membagi data target dan data fitur. Selanjutnya akan dilakukan *Transformation* yaitu tahapan mengonversi data kategorikal (kolom GENDER dan LUNG_CANCER) ke format numerik. Tahap berikutnya adalah *Data Mining* menggunakan *machine learning* yang sebelumnya data akan di*split* atau dibagi menjadi 80% *data training* dan 20% *data testing* lalu dilatih dengan algoritma *Naive Bayes* dan *Random Forest*. Tahap terakhir *Evaluation* dimana performa model akan dievaluasi dari *classification report* akurasi, presisi, *recall*, F1-Score, serta *confusion matrix*.

4. Hasil dan Pembahasan

4.1.Data Selection

Pada tahap ini, peneliti memilih dan mengumpulkan data yang akan digunakan untuk proses pelatihan dan pengujian model pembelajaran mesin. Dataset yang digunakan dalam penelitian ini berasal dari situs berbasis komunitas *Kaggle*, yang menyediakan berbagai kumpulan data terbuka yang dapat digunakan untuk keperluan penelitian dan pengembangan model berbasis *data science*. Dataset yang dipilih berjudul *Lung Cancer Dataset* dan berisi 1.157 data pasien dengan 16 kolom fitur yang mencakup informasi demografis seperti umur dan jenis kelamin, serta beberapa gejala atau kebiasaan yang berhubungan dengan risiko kanker paru-paru, seperti batuk, nyeri dada, merokok, dan kelelahan. Salah satu kolom yang paling penting adalah kolom target atau label, yaitu *LUNG_CANCER*, yang berisi nilai "*YES*" atau "*NO*" yang menunjukkan apakah pasien tersebut mengidap kanker paru-paru atau tidak.

4.2.Data Prepocecing

Langkah awal pada tahap pra-pemrosesan adalah melakukan pemeriksaan terhadap nilai yang hilang (*missing value*). Berdasarkan hasil analisis awal, tidak ditemukan adanya nilai kosong dalam dataset yang digunakan, sehingga proses imputasi atau pengisian nilai tidak diperlukan. Selain memeriksa *missing value*, juga dilakukan pemeriksaan terhadap data duplikat yang dapat ditemukan 2 baris data yang memiliki isi yang identik. Untuk menghapus duplikat, fungsi *data.drop_duplicates(keep='first')* hanya mempertahankan kemunculan pertama dari setiap baris yang sama. Untuk menjaga kualitas dan keakuratan model yang akan dibangun, data duplikat pada gambar 4.1 dihapus dari dataset, dengan tetap mempertahankan satu entri pertama dari setiap data yang terduplikasi. Data yang telah dibersihkan disimpan dalam variabel *data_cleaned*, dan *data_cleaned.duplicated* digunakan untuk memverifikasinya.

```
[31] # Define df as an alias for csv_data
     df

     # Periksa baris duplikat dalam kumpulan data
     duplicate_rows = df[df.duplicated()]

     # Hitung jumlah baris duplikat
     duplicate_count = duplicate_rows.shape[0]

     # Tampilkan baris duplikat jika ada
     duplicate_count, duplicate_rows.head()

     jumlah_duplikat = df.duplicated().sum()
     print("Jumlah data duplikat:", jumlah_duplikat)

     Jumlah data duplikat: 246
```

Gambar 4.1 Duplikat data

Tahap selanjutnya adalah *labeling* atau penanda khusus pada setiap data untuk menunjukkan kategori atau kelas dari data tersebut. Kolom *LUNG_CANCER* dipilih sebagai label karena proses *labeling* ini digunakan untuk menandai apakah seseorang dalam dataset terindikasi menderita kanker paru-paru atau tidak.

```
[48] from sklearn.model_selection import train_test_split

     # Daftar fitur terpilih berdasarkan korelasi
     selected_features = [
         'SHORTNESS OF BREATH',
         'CHEST PAIN',
         'SWALLOWING DIFFICULTY',
         'SMOKING',
         'FATIGUE ',
         'COUGHING',
         'WHEEZING',
         'YELLOW_FINGERS',
         'AGE',
         'CHRONIC DISEASE',
         'PEER_PRESSURE',
         'LUNG_CANCER'  # target
     ]
```

```
[48] # Seleksi DataFrame berdasarkan fitur tersebut
     df_selected = df[selected_features].copy() # Use .copy()

     # Define X (features) and y (target)
     X = df_selected.drop('LUNG_CANCER', axis=1) # Drop the ta
     y = df_selected['LUNG_CANCER'] # Select the target column

     # 4. Split data ke train dan test (80% latih, 20% uji)
     X_train, X_test, y_train, y_test = train_test_split(
         X, y, test_size=0.2, random_state=42, stratify=y
     )

     # 5. Cek hasil ukuran data
     print("Ukuran data latih:", X_train.shape)
     print("Ukuran data uji:", X_test.shape)

     Ukuran data latih: (925, 11)
     Ukuran data uji: (232, 11)
```

Gambar 4.2 *Split data*

Dataset kemudian difilter agar hanya memuat fitur-fitur pada gambar 4.2, termasuk variabel target *lung cancer*. Selanjutnya, data dibagi menjadi dua bagian menggunakan fungsi train_test_split dari pustaka Scikit-Learn, dengan proporsi 80% sebagai data latih dan 20% sebagai data uji. Parameter stratify digunakan untuk menjaga proporsi distribusi kelas target agar tetap seimbang antara data latih dan uji, sedangkan random_state=42 digunakan untuk menjamin reprodusibilitas hasil. Pemisahan ini bertujuan agar model machine learning dapat dilatih dengan data yang representatif dan diuji performanya secara objektif pada data yang belum pernah dilihat sebelumnya.

4.3.**Transformation**

Dalam konteks penelitian ini, transformasi difokuskan pada pengolahan variabel kategorikal yang masih berbentuk string atau teks. Dua fitur dalam dataset yang digunakan masih memiliki nilai berupa teks, yaitu kolom *GENDER* yang berisi nilai *"MALE"* dan *"FEMALE"*, serta kolom *LUNG_CANCER* sebagai label target yang memiliki nilai *"YES"* dan *"NO "*. *Algoritma machine learning* tidak dapat memproses data dalam format string, sehingga diperlukan konversi ke bentuk numerik. Konversi dilakukan menggunakan metode *Label Encoding*, yaitu proses mengubah setiap nilai kategorikal menjadi angka unik. Pada kasus ini pada kolom *GENDER* nilai "*MALE*" diubah menjadi 1, dan "*FEMALE*" menjadi 2. Hal yang sama juga diterapkan pada kolom *LUNG_CANCER*, di mana "*YES*" dikonversi menjadi 2 dan "*NO*" menjadi 1.

4.4.**Data Mining**

Pada tahap ini, data yang telah diproses melalui proses seleksi, pra-pemrosesan, dan transformasi kemudian dianalisis menggunakan algoritma pembelajaran mesin untuk melakukan klasifikasi. Dalam penelitian ini, lima algoritma klasifikasi yang biasa digunakan untuk mendeteksi penyakit digunakan *Naive Bayes* dan *Random Forest.*

```
[84] from sklearn.naive_bayes import GaussianNB
     from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
     import seaborn as sns
     import matplotlib.pyplot as plt

     # --- Buat dan latih model Naive Bayes ---
     nb_model = GaussianNB()
     nb_model.fit(X_train, y_train)
```

Gambar 4.1. Naive Bayes

Untuk membangun model probabilistik pada gambar 4.1, library yang digunakan mencakup *GaussianNB* dari modul *sklearn.naive_bayes*. Pada bagian utama kode, konstruktor *GaussianNB()* digunakan untuk membuat objek model *nb_model*, yang menganggap bahwa setiap fitur memiliki distribusi normal. Selanjutnya, metode digunakan untuk menguji *model. fit()* pada data kereta *X_train* dan *y_train*.

```
[40] from sklearn.ensemble import RandomForestClassifier
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
     import matplotlib.pyplot as plt

     # --- Buat model Random Forest ---
     rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
     rf_model.fit(X_train, y_train)

     # --- Prediksi ---
     y_pred = rf_model.predict(X_test)
```

Gambar 4.2. Random Forest

Pada Gambar 4.2 dilakukan impor berbagai modul yang dibutuhkan, termasuk *RandomForestClassifier* dari *sklearn.ensemble*, serta fungsi evaluasi dan visualisasi. Model *Random Forest* dibuat dengan parameter *n_estimators*=100, yang berarti model akan menggunakan 100 pohon keputusan untuk membentuk *ensemble*, serta *random_state*=42 untuk memastikan hasil yang konsisten. Model tersebut kemudian dilatih menggunakan data pelatihan (*X_train* dan *y_train*). Setelah pelatihan selesai, model digunakan untuk memprediksi label dari data pengujian (*X_test*) dan hasil prediksi disimpan dalam variabel *y_pred*.

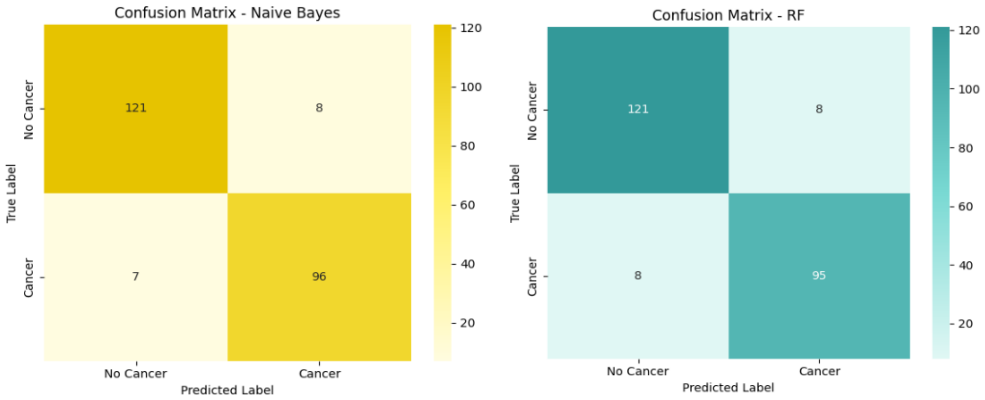
4.5.**Evaluation**

Setelah melakukan tahap *prepocecing* dan *data mining*, tahap evaluasi dilakukan untuk mengukur seberapa baik algoritma dalam mengklasifikasikan data. Proses evaluasi dilakukan dengan membandingkan hasil prediksi masing-masing model dengan data uji. Ini dilakukan dengan menggunakan berbagai metrik evaluasi, termasuk *accuracy* (akurasi), *precision* (presisi), *recall* (sensitivitas), skor F1, dan visualisasi matrix confusion.

Tabel 2.1 *Evaluation*

Algoritma	*Akurasi*	*Presisi*	*Recall*	*F1 - Score*
Naive Bayes	93,53%	0 = 0.95 1 = 0.92	0 = 0.94 1 = 0.93	0 = 0.94 1 = 0.93
Random Forest	93,10%	0 = 0.94 1 = 0.92	0 = 0.94 1 = 0.92	0 = 0.94 1 = 0.92

untuk memberikan informasi lebih detail tentang performa model dalam hal presisi, recall, dan F1-score untuk masing-masing kelas dan untuk *confusion matrix* divisualisasikan dalam bentuk *heatmap* yaitu matiks berwarna menggunakan fungsi *sns.heatmap* seperti pada Gambar 4.3 berikut ini :



Gambar 4.3 *Confusion matrix*

Berdasarkan *confusion matrix* model *Naive Bayes* pada gambar 4.3, model berhasil mengklasifikasikan 121 data "*No Cancer*" dan 96 data "*Cancer*" secara benar. Terdapat 8 kesalahan prediksi positif palsu (*False Positive*) dan 7 negatif palsu (*False Negative*). Dengan nilai akurasi sekitar 93,53%, *precision* 92,31%, *recall* 93,20%, dan *F1-score* 92,72%, model ini menunjukkan performa yang cukup baik dan seimbang dalam mendeteksi kasus kanker. Berdasarkan *confusion matrix* model *Random Forest*, model berhasil mengklasifikasikan 121 data "*No Cancer*" dan 95 data "*Cancer*" secara benar. Terdapat 8 kesalahan prediksi positif palsu (*False Positive*) dan 8 negatif palsu (*False Negative*). Dengan nilai akurasi sekitar 93,10%, *precision* 92,23%, *recall* 92,23%, dan *F1-score* 92,23%, model ini menunjukkan performa yang cukup baik dan seimbang dalam mendeteksi kasus kanker.

5. Kesimpulan dan Saran

Berdasarkan hasil penelitian yang telah dilakukan mengenai deteksi dini kanker paru-paru menggunakan metode klasifikasi berbasis *machine learning*, dapat disimpulkan bahwa kedua algoritma klasifikasi yang digunakan menunjukkan kinerja yang baik dalam mendeteksi kanker paru-paru, meskipun performanya bervariasi. Berdasarkan evaluasi model, algoritma *Naive Bayes* memberikan hasil terbaik dengan akurasi 93,53% dan F1-score 0,9272, menunjukkan bahwa pendekatan probabilistik sederhana tetap efektif untuk klasifikasi berbasis gejala klinis.

Berikut saran yang dapat menjadi bahan pertimbangan bagi penelitian lanjutan di bidang yang sama, disarankan untuk melakukan eksplorasi lanjutan terhadap algoritma klasifikasi lain atau teknik optimasi hyperparameter seperti Grid Search atau Randomized Search, serta penerapan metode ensemble hybrid (misalnya kombinasi Random Forest dan Gradient Boosting), guna meningkatkan performa dan generalisasi model dalam menghadapi data yang lebih kompleks.

Referensi

[1] N. Susanti, A. Nuraida, I. A. Amanda, and K. Khairunnisa, "Pencegahan Dan Penanggulangan Penyakit Tidak Menular Pada Remaja," *Jurnal Kesehatan Tambusai*, vol. 5, no. 2, pp. 4223–4233, Jun. 2024, doi: 10.31004/JKT.V5I2.28636.

[2] S. A. A. Kharis, A. H. A. Zili, F. I. Fajar, A. Putri, and M. Arisanty, “A Literature Review to Evaluate the Impact of Machine Learning and Artificial Intelligence for Lung Cancer Patient in COVID-19 Pandemic,” *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 1, pp. 576–583, Jan. 2024, doi: 10.33379/GTECH.V8I1.3891.

[3] J. F. Idris, R. Ramadhani, and M. M. Mutoffar, “Klasifikasi Penyakit Kanker Paru Menggunakan Perbandingan Algoritma Machine Learning,” *Jurnal Media Akademik (JMA)*, vol. 2, no. 2, Feb. 2024, doi: 10.62281/V2I2.145.

[4] A. Nugroho and D. Harini, “Teknik Random Forest untuk Meningkatan Akurasi Data Tidak Seimbang,” *JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer*, vol. 2, no. 2, pp. 128–140, Jun. 2024, doi: 10.53624/JSITIK.V2I2.379.

[5] A. Ristyawan, A. Nugroho, and T. K. Amarya, “Optimasi Preprocessing Model Random Forest untuk Prediksi Stroke,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 12, no. 1, pp. 29–44, Mar. 2025, doi: 10.35957/JATISI.V12I1.9587.

[6] A. Nugroho, “Analisa Splitting Criteria Pada Decision Tree dan Random Forest untuk Klasifikasi Evaluasi Kendaraan,” *JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer*, vol. 1, no. 1, pp. 41–49, Dec. 2022, doi: 10.53624/JSITIK.V1I1.154.

[7] E. Daniati, “Decision support systems to determining programme for students using DBSCAN and naive bayes: Case study: Engineering faculty of universitas nusantara PGRI kediri,” *Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology, ICAIIT 2019*, pp. 238–243, Mar. 2019, doi: 10.1109/ICAIIT.2019.8834474.

[8] M. Iqbal, I. Jauhar, A. A. Bayu, A. Ristyawan, and E. Daniati, “Klasifikasi Penggunaan Video Game Dengan Menggunakan Metode Algoritma Naive Bayes,” *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, vol. 8, no. 2, pp. 910–917, Jul. 2024, doi: 10.29407/INOTEK.V8I2.5020.

[9] M. N. Fadli, I. S. Damanik, and E. Irawan, “Penerapan Metode Naive Bayes Dalam Menentukan Tingkat Kenyamanan Pada Rumah Sakit Terhadap Pasien,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 2, no. 3, pp. 117–122, Dec. 2021, doi: 10.30865/KLIK.V2I3.297.

[10] A. Y. Simanjuntak, I. S. eptian S. Simatupang, and A. Anita, “Implementasi Data Mining Menggunakan Metode Naïve Bayes Classifier Untuk Data Kenaikan Pangkat Dinas Ketenagakerjaan Kota Medan,” *Journal Of Science And Social Research*, vol. 5, no. 1, pp. 85–91, Feb. 2022, doi: 10.54314/JSSR.V5I1.804.

[11] O. D. Kurnia, E. T. A’Fena, D. Y. A. Ningrum, E. Daniati, and A. Ristyawan, “Analisis Perbandingan Algoritma Naïve Bayes dengan K-Nearest Neighbor (KNN) Pada Dataset Mobile Price Classification,” *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, vol. 8, no. 2, pp. 1174–1183, Jul. 2024, doi: 10.29407/INOTEK.V8I2.5053.

[12] achmad Ridwan, “Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus,” *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, Oct. 2020, doi: 10.47970/SISKOM-KB.V4I1.169.

[13] B. Kriswantara and R. Sadikin, “Machine Learning Used Car Price Prediction with Random Forest Regressor Model,” *Journal of Information System, Informatics and Computing*, vol. 6, no. 1, pp. 40–49, Jun. 2022, doi: 10.52362/JISICOM.V6I1.752.

[14] P. K. Sari and R. R. Suryono, "Komparasi Algoritma Support Vector Machine Dan Random Forest Untuk Analisis Sentimen Metaverse," *Jurnal Mnemonic*, vol. 7, no. 1, pp. 31–39, Feb. 2024, doi: 10.36040/MNEMONIC.V7I1.8977.

[15] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 5, no. 2, pp. 697–711, Sep. 2021, doi: 10.30645/J-SAKTI.V5I2.369.

[16] G. Rininda, I. H. Santi, and S. Kirom, "Penerapan Svm Dalam Analisis Sentimen Pada Edlink Menggunakan Pengujian Confusion Matrix," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 5, pp. 3335–3342, Jan. 2023, doi: 10.36040/JATI.V7I5.7420.

[17] Chandan, M. S. R. (2021). *More Accurate Lung Cancer Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/chandanmsr/more-accurate-lung-cancer-dataset