PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR (KNN) UNTUK MENGKLASIFIKASIKAN STATUS KESEHATAN

SKRIPSI

Oleh :
NAZHIFATUL MUTHOHHAROH
2021503082

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO
2025

PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR (KNN) UNTUK MENGKLASIFIKASIKAN STATUS KESEHATAN

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Persyaratan dalam Menyelesaikan
Program Sarjana (S-1) Pada Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi Universitas Ibrahimy

Oleh :
NAZHIFATUL MUTHOHHAROH
2021503082

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO
2025

PERSETUJUAN PEMBIMBING

Nama : **Nazhifatul Muthohharoh**

NPM/NIM : 2021503082

Judul : **Perbandingan Algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) untuk Mengklasifikasikan Status Kesehatan**

Telah disetujui oleh :

Pembimbing 1	Pembimbing 2
Lukman Fakih Lidimillah, M. Kom	Ahmad Homaidi, M. Kom
NIDN: 0715099001	NIDN: 0705078901

PENGESAHAN

SKRIPSI

PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR (KNN) UNTUK MENGKLASIFIKASIKAN STATUS KESEHATAN

NAZHIFATUL MUTHOHHAROH
2021503082

telah dipertahankan di depan dewan penguji Sidang/Munaqasyah Skripsi pada hari Kamis, Tanggal 28 Agustus 2025 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom) pada Fakultas Sains dan Teknologi Universitas Ibrahimy

Tim Penguji,

Ketua Sidang,	Sekretaris Sidang,
Ahmad Homaidi, M.Kom	**Khairur Rosiqin, S.Kom**
NIDN.0705078901	**NIDN.-**
Penguji I,	Penguji II,
Adi Susanto, M.Kom	**A. Hamdani, M.Kom**
NIDN.0708079104	**NIDN.0730118806**

Mengetahui
Dekan,

Abd. Ghofur, M.Kom
NIDN.0711088303

MOTTO

"Segala Sesuatu Yang Diusahakan Maksimal, Maka Hasilnya Akan Maksimal, Bukan Sempurna"

-Zainuri Arifin Billah-

KATA PENGANTAR

Segala puji syukur peneliti sampaikan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya, perencanaan, pelaksanaan dan penyelesaian tugas akhir/skripsi dengan judul "Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbor (KNN) Untuk Mengklasifikasikan Status Kesehatan" sebagai salah satu syarat penyelesaian program diploma/sarjana dapat terselesaikan dengan baik dan lancar, oleh karena itu kami mengucapkan terima kasih kepada :

1. KHR. Ahmad Azaim Ibrahimy selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo.
2. KH. Ahmad Fadloil, S.H., M.H. selaku Rektor Universitas Ibrahimy Situbondo.
3. Abd. Ghofur, M. Kom., selaku Dekan Fakultas Sains dan Teknologi Universitas Ibrahimy
4. Firman Santoso, M. Kom., selaku Ketua Program Studi Teknologi Informasi.
5. Lukman Fakih Lidimilah, M. Kom., dan Ahmad Homaidi, M.Kom selaku Dosen Pembimbing I dan Pembimbing II yang telah membimbing saya untuk penyelesaian skripsi ini.
6. Seluruh Dosen Fakultas Sains dan Teknologi Universitas Ibrahimy yang telah memberikan ilmu sehingga saya dapat menyelesaikan Tugas Akhir pada tahun ini.

Situbondo, 21 Agustus 2025

Peneliti

Nazhiiatul Muthohharoh

PERSEMBAHAN

Saya persembahkan skripsi ini kepada kepada kedua orang tua, dan sulung yang memberikan segalanya untuk selesainya rangkaian tugas akhir skripsi ini, serta seluruh orang-orang yang tidak bisa saya sebutkan satu persatu. Baik dari pihak dosen dan teman-teman. Terimakasih atas seluruh do'a, dukungan dan apapun itu.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR LAMPIRAN

DAFTAR SEGMEN PROGRAM

DAFTAR TABEL

ABSTRAK

Nazhifatul Muthohharoh. 2025. **Perbandingan Algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) Untuk Mengklasifikasikan Status Kesehatan**. Skripsi, Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Ibrahimy. Pembimbing: (I) Lukman Fakih Lidimilah, M. Kom., (II) Ahmad Homaidi, M.Kom.

Kesehatan merupakan aspek penting dalam kehidupan manusia yang dipengaruhi oleh berbagai faktor, termasuk gaya hidup, aktivitas sehari-hari, pola makan, serta kebiasaan merokok dan olahraga. Gaya hidup yang tidak sehat dapat meningkatkan risiko penyakit kronis sehingga diperlukan sistem prediksi yang mampu mengklasifikasikan status kesehatan secara akurat. Penelitian ini bertujuan untuk membandingkan performa algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) dalam mengklasifikasikan status kesehatan menggunakan dataset Lifestyle Habits yang diperoleh dari Kaggle dengan jumlah 389 *record*. Metode penelitian menggunakan pendekatan CRISP-DM yang meliputi tahap *business understanding, data understanding, data preparation, modeling, evaluation,* hingga *deployment*. Proses pemodelan dilakukan menggunakan *Python* di *Google Colab* dengan *library scikit-learn*, sedangkan implementasi sistem berbasis web menggunakan Streamlit. Evaluasi model dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *f1-score*. Hasil penelitian menunjukkan bahwa algoritma Naïve Bayes menghasilkan akurasi sebesar 92%, sedangkan algoritma KNN (dengan parameter K terbaik) juga menunjukkan performa yang kompetitif. Dari hasil perbandingan, dapat disimpulkan bahwa kedua algoritma sama-sama mampu melakukan klasifikasi status kesehatan dengan baik, namun terdapat perbedaan karakteristik kinerja pada masing-masing algoritma. Penelitian ini diharapkan dapat memberikan kontribusi dalam mendukung upaya deteksi dini faktor risiko kesehatan serta menjadi alternatif solusi dalam pengambilan keputusan di bidang kesehatan berbasis *data mining*.

Kata kunci : Data Mining, K-Nearest Neighbor, Klasifikasi, Naïve Bayes, Status Kesehatan

ABSTRACT

Nazhifatul Muthohharoh. 2025. ***Comparison of Naïve Bayes and K-Nearest Neighbor (KNN) Algorithms for Classifying Health Status***. *Thesis, Information Technology Study Program, Faculty Science and Technology, Ibrahimy University. Supervisor*: (I) Lukman Fakih Lidimilah, M. Kom., (II) Ahmad Homaidi, M. Kom.

Health is an important aspect of human life that is influenced by various factors, including lifestyle, daily activities, diet, smoking habits, and exercise. An unhealthy lifestyle can increase the risk of chronic diseases, so a prediction system that can accurately classify health status is needed. This study aims to compare the performance of the Naïve Bayes and K-Nearest Neighbor (KNN) algorithms in classifying health status using the Lifestyle Habits dataset obtained from Kaggle, which contains 389 records. The research method employs the CRISP-DM approach, which includes the stages of business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The modeling process was conducted using Python on Google Colab with the scikit-learn library, while the web-based system implementation used Streamlit. Model evaluation was conducted using accuracy, precision, recall, and F1-score metrics. The results showed that the Naïve Bayes algorithm achieved an accuracy of 92%, while the KNN algorithm (with the optimal K parameter) also demonstrated competitive performance. From the comparison results, it can be concluded that both algorithms are equally capable of classifying health status effectively, but there are differences in performance characteristics between the two algorithms. This study is expected to contribute to efforts in supporting early detection of factors.

Keywords: *Data Mining, K-Nearest Neighbor, Classification, Naïve Bayes, Health Status*

BAB I

PENDAHULUAN

1.1 Latar belakang

Salah Satu aspek penting dalam kehidupan manusia adalah kesehatan. Status kesehatan seseorang dipengaruhi oleh berbagai faktor, termasuk gaya hidup, aktivitas sehari-hari, kebiasaan bekerja, merokok, dan olahraga. Gaya hidup yang tidak sehat dapat meningkatkan risiko berbagai penyakit, sehingga penting untuk mengidentifikasi faktor-faktor yang berkontribusi terhadap status kesehatan guna mengambil langkah preventif.

Gaya hidup yang tidak sehat, seperti kurangnya aktivitas fisik, pola makan yang tidak seimbang, dan kebiasaan merokok, merupakan penyebab utama munculnya berbagai penyakit kronis yang berdampak signifikan terhadap kualitas hidup individu serta meningkatkan beban ekonomi pada sistem kesehatan masyarakat, sehingga upaya untuk mengidentifikasi secara dini faktor risiko kesehatan yang berkaitan dengan gaya hidup menjadi tantangan penting dalam dunia kesehatan modern [1].

Di tengah kondisi ini, ada kebutuhan yang mendesak untuk mengembangkan sistem prediksi kesehatan yang dapat membantu mengidentifikasi individu-individu yang berisiko mengalami masalah kesehatan akibat gaya hidup mereka. Namun, metode manual yang digunakan oleh tenaga medis saat ini sering kali terbatas oleh waktu dan tenaga. Dengan memanfaatkan teknik data mining, sebuah data akan dapat dianalisis untuk menghasilkan informasi yang dapat membantu dalam pengambilan keputusan terkait kesehatan [2]. Dua algoritma yang

sering digunakan dalam klasifikasi data adalah Naive Bayes dan K-Nearest Neighbors (K-NN). Kedua algoritma ini memiliki karakteristik yang berbeda dalam mengolah dan mengklasifikasikan data.

Klasifikasi adalah salah satu cabang dalam ilmu data yang berfokus pada pengelompokan atau pengkategorian data ke dalam kelas-kelas tertentu berdasarkan karakteristik atau atribut yang dimiliki oleh data tersebut. Proses ini bertujuan untuk membuat prediksi atau keputusan berdasarkan pola yang ditemukan dalam data historis. Secara teknis, klasifikasi merupakan bagian dari pembelajaran mesin (*machine learning*) yang termasuk dalam kategori pembelajaran terawasi (*supervised learning*)[3] . Dalam proses ini, algoritma dilatih menggunakan dataset yang telah diberi label atau kategori tertentu sebelumnya. Data pelatihan ini mengandung pasangan input (atribut) dan output (label) yang memungkinkan algoritma belajar mengenali hubungan atau pola antara keduanya. Setelah pelatihan selesai, model klasifikasi diharapkan mampu memprediksi kelas dari data baru yang belum pernah dilihat sebelumnya [4].

Naive Bayes dan K-Nearest Neighbor (K-NN) merupakan dua algoritma klasifikasi dalam pembelajaran mesin yang memiliki karakteristik dan pendekatan berbeda namun tetap menunjukkan beberapa kesamaan. Naive Bayes didasarkan pada Teorema Bayes dengan asumsi independensi antar fitur, dan bekerja dengan menghitung probabilitas setiap kelas berdasarkan distribusi data, menjadikannya efisien dalam proses training serta cocok untuk dataset besar dan berdimensi tinggi. Sebaliknya, K-NN merupakan algoritma berbasis instance-based learning yang melakukan klasifikasi dengan mengukur kedekatan atau jarak antar data, tanpa

membangun model eksplisit, sehingga proses training-nya cepat namun prediksi bisa lambat, terutama pada dataset besar. Dalam hal akurasi, Naive Bayes unggul pada data dengan fitur independen dan distribusi yang jelas, sedangkan K-NN lebih akurat pada dataset kecil-menengah dengan pola non-linear, walaupun sensitif terhadap jumlah tetangga dan keberadaan outlier. Dari segi interpretabilitas, Naive Bayes mudah dipahami karena menghasilkan probabilitas untuk setiap prediksi, sedangkan K-NN sulit dianalisis karena tidak menghasilkan model yang jelas. Meski begitu, keduanya digunakan untuk klasifikasi dalam supervised learning, mendukung klasifikasi multi-kelas, dapat menangani data diskrit maupun kontinu, dan bergantung pada kualitas data latih dalam menentukan akurasi hasil klasifikasi. K-NN bersifat non-parametrik sepenuhnya, sementara Naive Bayes dianggap semi-parametrik karena tetap membuat asumsi tertentu namun mampu menyesuaikan diri terhadap data.

Metode klasifikasi yang sering dijumpai salah satunya adalah metode *K-Nearest Neighbor* (KNN) dan *Naïve Bayes*. Dalam penelitian terdahulu, telah dilakukan pengujian kedua metode tersebut yang menunjukkan bagaimana teknologi *Data Mining*, khususnya metode *K-Nearest Neighbor* (K-NN) dan *Naïve Bayes*, dapat digunakan untuk klasifikasi status gizi pada balita, penelitian ini relevan karena mengklasifikasikan sebuah status pada suatu data kesehatan. Hasil penelitian menunjukkan bahwa algoritma K-NN menghasilkan tingkat akurasi yang tinggi [5]. Selain itu, dalam penelitian lain menunjukkan bahwa algoritma naive bayes lebih unggul dari pada algoritima K-NN dalam klasifikasi status pertumbuhan anak stunting [6].

Penelitian ini bertujuan untuk mengetahui algoritma mana yang lebih efektif dan efisien diantara naive bayes dan K-NN. Adanya penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam upaya pencegahan penyakit dalam mengidentifikasi pola kebiasaan yang beresiko serta memberi rekomendasi yang lebih akurat dalam pengambilan keputusan di bidang kesehatan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, maka diperoleh identifikasi masalah sebagai berikut:

a. Banyak individu mengalami penurunan status kesehatan akibat kebiasaan gaya hidup yang tidak sehat, seperti kurangnya aktivitas fisik, kebiasaan merokok, serta pola makan yang tidak seimbang. Kondisi ini berkontribusi terhadap peningkatan risiko penyakit kronis seperti obesitas, diabetes, dan penyakit jantung, sehingga diperlukan metode yang dapat membantu dalam mengklasifikasikan status kesehatan berdasarkan faktor gaya hidup.
b. Metode K-Nearest Neighbor (K-NN) dan Naïve Bayes memiliki pendekatan berbeda dalam klasifikasi data. K-NN mengandalkan kedekatan data untuk menentukan kelas, sedangkan Naïve Bayes menggunakan pendekatan probabilistik. Perbedaan ini menimbulkan tantangan dalam memilih metode yang lebih efektif, karena K-NN unggul dalam akurasi, sementara Naïve Bayes lebih cepat namun kurang akurat pada data dengan fitur saling berkorelasi.

1.3 Rumusan Masalah

Rumusan masalah yang dapat diambil dari latar belakang diatas yaitu Bagaimana mengevaluasi performa algoritma seperti Naive Bayes dan K-NN dalam memproses data kesehatan untuk menentukan algoritma yang paling efektif, sehingga informasi yang didapatkan dapat dijadikan sebagai alternatif untuk menangani permasalahan yang ada.

1.4 Batasan Masalah

Untuk memudahkan penulisan dalam penyusunan penelitian ini, dan agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka penulis membatasi permasalahan sebagai berikut:

a. Penggunaan data yang diteliti didapat dari website kaggle.
b. Data yang akan diuji dan diolah dalam data mining berupa data status kesehatan menggunakan perbandingan metode naive bayes dan K-NN.

1.5 Tujuan Penelitian

Tujuan pengamatan ini adalah untuk membandingkan akurasi dan efisiensi algoritma *K-Nearest Neighbor* (K-NN) dan *Naïve Bayes* dalam mengklasifikasikan status kesehatan. Evaluasi dilakukan dengan membandingkan atribut kinerja, yaitu akurasi, presisi, *recall*, *f1-score*, dan waktu komputasi, guna menentukan metode yang lebih optimal untuk mengklasifikasikan status kesehatan secara efektif dan efisien.

1.6 Manfaat Penelitian

Diharapkan, analisis dan penelitian ini dapat memberikan kontribusi yang dapat diketahui oleh berbagai pihak :

a. Meningkatkan akurasi diagnosis: Dengan mengevaluasi dan memilih algoritma terbaik berdasarkan karakteristik data, penelitian ini dapat membantu meningkatkan akurasi diagnosis yang mendukung perawatan pasien yang lebih tepat sasaran.

b. Deteksi dini faktor-faktor risiko kesehatan: memberikan kontribusi dalam upaya peningkatan deteksi dini faktor-faktor risiko kesehatan melalui pendekatan teknologi yang lebih praktis dan cepat.

1.7 Metode Penelitian

1.7.1 Jenis Penelitian

Penelitian ini tergolong ke dalam jenis penelitian kuantitatif, yang bertujuan menganalisis data numerik secara objektif dengan dukungan metode statistik. Dalam studi ini, data diolah menggunakan algoritma *machine learning,* yakni algoritma Naive Bayes dan K-NN, guna membangun model klasifikasi yang mampu mendeteksi sattus kesehatan pada individu berdasarkan sejumlah atribut tertentu. Hasil analisis tersebut kemudian dimanfaatkan untuk merancang sistem deteksi dini yang dapat diakses melalui *platform web* [7].

1.7.2 Teknik Pengumpulan Data

Untuk memperoleh hasil penelitian, tahapan pengumpulan data dilakukan melalui langkah-langkah berikut :

a. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan oleh penulis adalah data yang disediakan oleh penyedia dataset online melalui website https://www.kaggle.com/

b. Studi Pustaka

Teknik pengumpulan data yang kedua adalah studi pustaka. Studi pustaka bertujuan mencari informasi yang mendukung penelitian ini, baik melalui media seperti buku, internet, maupun media informasi lainnya

1.7.3 Metode Pengembangan Sistem

Metode yang digunakan dalam penelitian ini adalah CRISP-DM (Cross Industry Standard Process for Data Mining). Metode ini dipilih karena memiliki tahapan yang sistematis, fleksibel, dan sesuai untuk proses analisis serta penerapan model data mining[8]. Tahapan CRISP-DM yang digunakan dalam penelitian ini adalah sebagai berikut:

1. **Business Understanding**

   Tahap ini bertujuan untuk memahami tujuan penelitian dan permasalahan yang akan diselesaikan. Fokus penelitian ini adalah membandingkan kinerja algoritma Naive Bayes dan K-Nearest Neighbor (K-NN) dalam mengklasifikasikan status kesehatan berdasarkan data kebiasaan gaya hidup *(Lifestyle Habits Dataset)*. Tujuan akhirnya adalah mengetahui algoritma yang memberikan hasil klasifikasi terbaik.

2. **Data Understanding**

   Pada tahap ini dilakukan pengumpulan dan pemahaman terhadap dataset yang digunakan. Dataset diperoleh dari platform Kaggle, kemudian dianalisis untuk mengetahui jumlah data, jenis atribut, distribusi kelas, serta mengidentifikasi adanya missing values atau data yang tidak konsisten.

3. Data Preparation

Data yang telah dikumpulkan diproses agar siap digunakan pada tahap pemodelan. Langkah-langkah pada tahap ini meliputi:

a. Menghapus atau memperbaiki nilai yang hilang (missing values).

b. Melakukan label encoding pada data kategorikal.

c. Melakukan normalisasi pada atribut numerik.

d. Membagi dataset menjadi data latih (training set) dan data uji (testing set) dengan proporsi tertentu.

4. Modeling

Pada tahap ini, model dibangun menggunakan algoritma Naive Bayes dan K-NN. Proses pemodelan dilakukan di Google Colab dengan bahasa pemrograman Python. Untuk algoritma K-NN, dilakukan pengujian dengan beberapa nilai parameter k untuk mendapatkan performa terbaik.

5. Evaluation

Kinerja model dievaluasi menggunakan metrik akurasi, precision, recall, dan f1-score. Perbandingan dilakukan untuk menentukan algoritma dengan performa terbaik.

6. Deployment

Model terbaik hasil penelitian diimplementasikan ke dalam aplikasi berbasis web menggunakan Streamlit. Aplikasi ini memungkinkan pengguna memasukkan data kebiasaan gaya hidup melalui antarmuka sederhana, kemudian sistem akan memproses data tersebut menggunakan model yang telah dilatih dan menampilkan hasil prediksi status kesehatan. Tahap ini bertujuan untuk

mempermudah penggunaan model oleh pihak yang tidak memiliki latar belakang teknis [9].

Gambar 1. 1 Metode Pengembangan Sistem

1.8 Sistematika Pembahasan

Sistematika pembahasan yang akan dicantukman dalam karya tulis ilmiah ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penelitian, identifikasi, rumusan, dan Batasan masalah, tujuan dan manfaat penelitian, metode dan jenis penelitian, Teknik pengumpulan data, metode pengembangan data, metode pengembangan sistem, dan sistematika pembahasan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang beberapa literatur yang menjadi dasar pemikiran dalam penelitian tersebut, diantaranya adalah penelitian terdahulu, landasan teori, pemodelan, dan perangkat lunak yang digunakan.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang bagaimana peneliti mendapatkan data yang digunakannya dalam penelitian ini dan bagaimana cara mendapatkannya berikut

memprosesnya sebelum melakukan Analisa lebih lanjut menggunakan algoritma yang terdapat pada Data Mining.

BAB IV HASIL DAN PEMBAHASAN

Bab ini adalah hasil dari Analisa yang telah diputuskan sebelumnya. Berisi tentang beberapa penjelasan maupun perhitungan dari data yang telah diolah sehingga menghadilkan sebuah prediksi berdasarkan kemungkinan yang telah didapatkan sebelumnya dan juga tingkat akurasi dari prediksi yang telah dihasilkan.

BAB V PENUTUP

Bab terakhir ini berisi tentang kesimpulan maupun saran yang diberikan dari sistem yang telah diteliti, dirancang, dan diimplementasikan.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian terdahulu

Penelitian ini meninjau beberapa penelitian sebelumnya meliputi:

Perbandingan Implementasi Machine Learning Menggunakan Metode KNN, Naive Bayes dan *Logistic Regression* Untuk Mengklasifikasi Penyakit Diabetes.

Penelitian ini dilakukan oleh Dewi Nasien, Ricalvin Darwin, Alexander Cia, Andrean Leo Winata, Jerry Go, Richard M.C, Ryan Charles Wijaya, dan Kevin Charles Lo, yang dipublikasikan di JEKIN (Jurnal Teknik Informatika), Volume 4, Nomor 1, pada tahun 2024, halaman 11–17.

Permasalahan yang dibahas dalam penelitian ini adalah tingginya angka kasus diabetes dan kebutuhan untuk mengidentifikasi metode klasifikasi yang optimal dalam menangani data diabetes. Selain itu, penelitian ini ingin mengetahui algoritma terbaik dalam membagi data untuk menghasilkan prediksi yang akurat.

Batasan masalah penelitian ini adalah membandingkan kinerja tiga algoritma klasifikasi K-Nearest Neighbor (KNN), Naive Bayes, dan Logistic Regression engan menggunakan data dari Kaggle yang dibagi menjadi dua skenario, yaitu 70:30 dan 80:20. Data diproses melalui Principal Component Analysis (PCA) dengan threshold 80% untuk ekstraksi fitur.

Tujuannya adalah untuk menganalisis kinerja masing-masing algoritma dalam klasifikasi data diabetes, mengevaluasi hasil akurasi, presisi, recall, dan F1-score dari setiap metode, serta memberikan rekomendasi algoritma terbaik

berdasarkan hasil perbandingan. Tujuan dari penelitian ini adalah mengidentifikasi algoritma klasifikasi yang paling optimal dalam menangani dataset diabetes, yang dapat mendukung diagnosis dini serta pengelolaan penyakit ini dengan lebih efisien.

Berdasarkan hasil penelitian, algoritma Naive Bayes menghasilkan akurasi terbaik sebesar 79% pada skenario pembagian data 80:20, sehingga dianggap sebagai algoritma paling sesuai untuk data diabetes dalam konteks penelitian ini [10].

Perbandingan Algoritma K-Nearest Neighbor dan Naïve Bayes Classifier Untuk Klasifikasi Status Gizi Pada Balita

Penelitian ini dilakukan oleh Septi Kenia Pita Loka dan Arif Marsal, yang dipublikasikan di MALCOM: Indonesian Journal of Machine Learning and Computer Science, Volume 3, Issue 1, pada bulan April 2023, halaman 8–14.

Permasalahan yang dibahas dalam penelitian ini adalah bagaimana mengklasifikasikan status gizi balita dengan lebih akurat, khususnya dalam kasus peningkatan balita yang mengalami wasting di Kota Solok. Data dari beberapa puskesmas menunjukkan adanya tren peningkatan balita dengan masalah gizi, sehingga diperlukan metode klasifikasi yang efektif untuk membantu proses pemantauan status gizi.

Batasan masalah penelitian ini adalah penggunaan dua algoritma klasifikasi, yaitu K-Nearest Neighbors (KNN) dan Naïve Bayes Classifier (NBC), dengan dataset yang diperoleh dari Dinas Kesehatan Kota Solok. Data yang digunakan terdiri dari atribut jenis kelamin, umur, berat badan (BB), tinggi badan (TB), dan

status BB/TB sebagai label. Tujuannya adalah untuk membandingkan performa algoritma KNN dan NBC dalam mengklasifikasikan status gizi balita berdasarkan data penimbangan massal, serta menentukan algoritma yang memberikan akurasi terbaik.

Tujuan dari penelitian ini adalah menghasilkan model klasifikasi yang dapat membantu Dinas Kesehatan Kota Solok dalam menentukan status gizi balita secara lebih efektif dan efisien. Hasil penelitian menunjukkan bahwa algoritma KNN dengan parameter k = 3 k=3 memiliki akurasi tertinggi sebesar 96,10%, sementara algoritma NBC memperoleh akurasi sebesar 90,94%. Dengan demikian, algoritma KNN lebih direkomendasikan untuk klasifikasi status gizi balita dibandingkan NBC [5].

Perbandingan Algoritma Naive Bayes dan K-Nearest Neighbors Untuk Klasifikasi Metabolik Sindrom.

Penelitian ini dilakukan oleh Fitriana Sholekhah, Adinda Dwi Putri, Rahmaddeni, dan Lusiana Efrizoni, yang dipublikasikan di MALCOM: Indonesian Journal of Machine Learning and Computer Science, Volume 4, Issue 2, pada bulan April 2024, halaman 507–514.

Permasalahan yang dibahas dalam penelitian ini adalah perlunya model untuk mendiagnosis sindrom metabolik yang dapat meningkatkan risiko penyakit kardiovaskular, diabetes, stroke, dan masalah kesehatan lainnya. Penelitian ini juga mengevaluasi efektivitas algoritma klasifikasi dalam menangani data kesehatan untuk diagnosis yang lebih akurat.

Batasan masalah penelitian ini adalah penggunaan dua algoritma klasifikasi utama, yaitu Naïve Bayes (NB) dan K-Nearest Neighbors (KNN), untuk menganalisis dataset sindrom metabolik yang diambil dari Kaggle, dengan pembagian data dalam tiga skenario: 50:50, 60:40, dan 70:30. Tujuannya adalah untuk membandingkan kinerja algoritma NB dan KNN berdasarkan akurasi, precision, dan recall dalam mengklasifikasikan data sindrom metabolik, serta menentukan skenario pembagian data terbaik untuk mencapai hasil optimal.

Tujuan dari penelitian ini adalah menghasilkan model klasifikasi yang optimal untuk diagnosis sindrom metabolik. Hasil penelitian menunjukkan bahwa algoritma KNN dengan pembagian data 50:50 memiliki akurasi tertinggi sebesar 82%, precision 85%, dan recall 87%. Sementara itu, algoritma Naïve Bayes menunjukkan akurasi terbaik sebesar 79%. Hasil ini menyimpulkan bahwa algoritma KNN lebih efektif dibandingkan NB dalam kasus ini. Penelitian selanjutnya diharapkan dapat menggunakan algoritma tambahan untuk meningkatkan performa model [11] .

2.2 Landasan Teori

Landasan Teori ini adalah beberapa referensi atau pengertian dari beberapa kata kunci yang akan dibahas dalam penelitian ini. Berikut adalah beberapa pengertian tersebut:

2.2.1 Data Mining

Pengertian data mining adalah proses penggalian informasi dan pola yang bermanfaat dari data yang sangat besar. Data mining mencakup pengumpulan data, ekstraksi data, analisis data. Data mining juga merupakan proses logis untuk

menemukan informasi yang berguna. Setelah ditemukan informasi yang berguna. Setelah ditemukan informasi dan pola dapat digunakan untuk alat pendukung dalam pengambilan keputusan dalam mengembangkan bisnis. Data mining juga dapat digunakan untuk meramalkan tren masa depan yang memungkinkan untuk membuat keputusan yang efektif, proaktif, dan dinamis. Data-data yang diolah dengan menggunakan teknik data mining juga mampu menghasilkan pengetahuan yang sesuai dengan harapan. Misalnya pada bidang kesehatan, cukup banyak data yang dimiliki oleh rumah sakit, seperti data *medical record* dan radiologi, tetapi karena belum adanya standar koleksi data maka data-data tersebut sukar untuk diolah sehingga dengan kehadiran data mining maka diharapkan data-data yang dimiliki oleh pihak kesehatan dapat diolah sesuai dengan keperluan hingga menghasilkan informasi dan pengetahuan yang dapat dimanfaatkan oleh para pengambil kebijakan terutama pemerintah [12].

Data mining bertujuan untuk menemukan pola yang sebelumnya tidak diketahui. Jika pola-pola tersebut telah diperoleh maka dapat digunakan untuk menyelesaikan berbagai macam permasalahan. Data mining saat ini juga telah menjadi suatu teknologi baru yang kuat dengan potensi besar, untuk membantu perusahaan fokus pada informasi paling penting dalam perusahaan[13].

Menurut Gartner Group, data mining adalah penggunaan teknologi pengenalan pola, seperti metode statistik dan matematika, untuk mengklasifikasikan sejumlah besar data yang disimpan pada media penyimpanan untuk menemukan hubungan baru dengan makna, pola, dan kebiasaan menemukan. Data mining adalah kombinasi berbagai disiplin ilmu yang menggabungkan teknik

seperti pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi untuk mengatasi masalah pengambilan informasi dari database besar [14].

2.2.2 Algoritma Naive Bayes

Naive Bayes merupakan suatu algoritma yang dapat mengklasifikasikan suatu variable tertentu dengan menggunakan metode probabilitas dan statistic. Naive bayes menggunkan sebuah ilmu cabang matematika yang dikenal juga dengan teori probabilitas untuk mencari peluang terbesar dari kemungkinan klasifikasi, dengan cara melihat frekuensi tiap klasifikasi pada data *training* [15].

2.2.3 Algoritma K-NN

K-Nearest Neighbors (knn) merupakan algoritma yang mengklasifikasikan data berdasarkan data pembelajaran (training data set) yang diambil dari k-tetangga terdekat (nearest neighbours). dimana k adalah banyaknya tetangga terdekat. Metode K-Nearest Neighbors melakukan 14 klasifikasi dengan memproyeksikan data latih ke dalam ruang multidimensi. Area ini dibagi menjadi beberapa bagian yang mewakili data dasar pelatihan. Semua data pelatihan direpresentasikan sebagai titik c dalam ruang multidimensi.

K-Nearest Neighbors (KNN) merupakan algoritma klasifikasi yang menggunakan himpunan nilai K dari data terdekat (tetangganya) sebagai acuan untuk menentukan kelas data baru. KNN mengklasifikasikan data berdasarkan kemiripan atau kedekatannya dengan data lain. Algoritma KNN ini adalah pembelajaran yang malas. Artinya, tidak menggunakan titik data pelatihan untuk membangun model. Dengan kata lain algoritma KNN mempunyai fase pelatihan

yang sangat minim. Tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel pada data pelatihan[16].

$$(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Setelah menghitung jarak Euclidean langkah selanjutnya adalah menentukan K- Neighborsnya dengan cara mengurutkan dari nilai yang kecil sampai yang terbesar. Dari K neighbors terdekat tentukan label berdasarkan mayoritas dari K tetangga terdekat untuk mengevaluasi model dari algoritma K-NN tersebut, setelah mengevaluasi mode langkah selanjutnya adalah menentukan kelas dari dataset, Dari seluruh perhitungan tersebut terbagi kepada perhitungan dengan *class* label *Normal*, *Suspect* dan *Phatologic* kemudian hasil yang terbesar dari perbandingan ketiga label tersebut merupakan hasil dari prediksi algoritma *K-Nearest Neighbors* tersebut.

2.2.4 Dataset

Data mining tidak bisa dipisahkan dari data set karena proses data mining sebenarnya membutuhkan data set sebagai objek ekstraksi pengetahuan [17]. Dalam terminologi statistik, kumpulan data adalah kumpulan objek dengan atribut atau variabel tertentu, dan setiap objek merupakan bagian data dengan sekumpulan atribut atau variabel. Nama lain yang umum digunakan untuk objek termasuk catatan, titik, vektor, pola, peristiwa, observasi, dan kasus. Sebaliknya, baris yang mewakili objek data atau kolom disebut atribut. Atribut terkadang disebut variabel, bidang, fitur, atau dimensi [18].

Karakteristik umum kumpulan data yang dapat memengaruhi proses penambangan data adalah dimensi, ketersebaran, dan resolusi. Saat ini, catatan diklasifikasikan menjadi tiga jenis tergantung pada jenisnya: Yang pertama adalah record yang berupa record data. Artinya, data adalah kumpulan kumpulan data yang masing-masing kumpulan atributnya tetap. Yang kedua adalah data grafik, Dengan kata lain merupakan data yang berbentuk grafik yang terdiri dari node dan edge. Misalnya link HTML (di WWW), struktur molekul, dll. Yang terakhir adalah data terurut, yaitu data yang berisi sekumpulan nilai, Atau menampilkan data yang diurutkan menurut pola tertentu. Contohnya termasuk data geonomic sequence atau spatio-temporal[19].

2.2.5 Atribut

Atribut adalah simbol yang menggambarkan identitas atau sifat suatu objek. Atribut yang menggambarkan suatu objek rawat inap antara lain nama, umur, golongan darah, dan tekanan darah. Atribut diklasifikasikan menjadi empat jenis: Atribut nominal, yaitu atribut yang diperoleh melalui klasifikasi untuk menggambarkan kategori, kode, atau status yang tidak berurutan. Yang kedua adalah atribut ordinal, yaitu atribut yang menggambarkan urutan atau peringkat. Namun besarnya selisih antara dua nilai yang berurutan tidak diketahui. Atribut spasi (jarak) merupakan atribut numerik yang ditentukan oleh pengukuran. Oleh karena itu, jarak antara dua titik pada skala diketahui dan tidak ada nol mutlak. Yang terakhir adalah atribut rasio (mutlak), yaitu atribut 16 numerik yang mempunyai titik nol mutlak. Artinya, Anda dapat menghitung perkalian atau perbandingan antara satu nilai dengan nilai lainnya[14].

2.2.6 Perbandingan Algoritma

Perbandingan algoritma adalah proses menganalisis kelebihan dan kekurangan berbagai algoritma berdasarkan kriteria tertentu, seperti akurasi, kecepatan, kompleksitas, skalabilitas, dan kemampuan mengatasi overfitting, untuk menentukan algoritma yang paling cocok dengan kebutuhan atau masalah tertentu. Dalam data mining dan machine learning, perbandingan ini bertujuan untuk mengevaluasi performa algoritma dalam memproses data, baik dari segi efisiensi waktu, ketepatan prediksi, hingga interpretabilitas hasilnya. Perbandingan juga mencakup analisis kemampuan algoritma dalam menangani data besar, data yang tidak seimbang, atau data dengan banyak noise. Selain itu, aspek seperti kemudahan implementasi, kebutuhan sumber daya komputasi, dan stabilitas hasil juga menjadi pertimbangan penting dalam membandingkan algoritma. Melalui proses perbandingan, algoritma yang paling efektif untuk konteks spesifik baik untuk klasifikasi, prediksi, atau pengelompokan dapat dipilih untuk menghasilkan solusi yang optimal [14].

2.2.7 Kesehatan

Kesehatan didefinisikan sebagai kondisi kesejahteraan fisik, mental, dan sosial yang utuh, bukan sekadar bebas dari penyakit atau disabilitas. Selain itu, kesehatan juga mencakup pencapaian potensi total anak, menekankan pentingnya mendukung perkembangan anak sehat selain menangani penyakit atau trauma. Kesehatan dipandang sebagai keseimbangan antara individu, agen (seperti bakteri, virus, dan toksin), serta lingkungan. Interaksi ini tidak hanya mencakup hubungan individu dengan agen, tetapi juga dengan lingkungannya untuk menciptakan

kesejahteraan. Kesehatan dapat dipahami sebagai proses dinamis untuk menjaga dan mendukung keseimbangan fisik, mental, dan adaptasi optimal dengan lingkungan sekitar. Dalam perspektif penyakit, sehat adalah kondisi keutuhan kemampuan fungsional dan keadaan yang lebih baik, di mana seseorang memiliki fungsi tubuh yang baik, mampu beradaptasi dengan lingkungan, serta merasa lebih baik secara subjektif. Kesehatan juga melibatkan keseimbangan antara pertumbuhan, fungsi, keutuhan, dan pemberdayaan sumber daya yang dimiliki. Seseorang dianggap sehat jika ia merasa lebih baik, kuat, memiliki fungsi tubuh yang baik, dan mampu beradaptasi secara memadai dengan lingkungannya [20].

2.2.8 Klasifikasi

Klasifikasi adalah proses analisis data yang bertujuan untuk mengelompokkan atau memetakan data ke dalam kategori atau kelas tertentu berdasarkan pola atau karakteristik yang ada dalam data tersebut. Dalam konteks yang lebih luas, klasifikasi mencakup berbagai bidang seperti statistik, machine learning, kecerdasan buatan, biologi, keuangan, hingga sistem informasi, dengan pendekatan dan tujuan yang disesuaikan untuk setiap kebutuhan[19].

2.1 Perangkat Lunak Yang Digunakan

a. Google Colab

Google colaboratory (*Google colab*) atau google interactive notebook adalah coding environment Bahasa pemprograman python dengan format “notebook” (mirip dengan jupyter notebook), atau dengan kata lain Google seakan meminjami lita komputer secara gratis untuk membuat program pyhton. Dengan google colab dapat mengeksekusi code pyhton di browser seperti halnya jupyter

notebook. Selain itu Goggle Colab bias dimanfaatkan guna menyimpan, menulis, dan membagikan program yang sudah ditulis dari Google Drive [21].

b. Bahasa Pemrograman Python

Bahasa pemrograman python Adalah bahasa pemrograman Tingkat tinggi, berjalan dengan interpreted, dan bisa dipakai untuk berbagai jenis tujuan. Python dikatakan bahasa pemrograman tingkat tinggi karena bahasa program yang dipakai sudah mirip dengan bahasa manusia, kode program akan diproses baris per baris langsung dari kode program artinya tidak butuh proses compile [22].

c. Framework Streamlit

Streamlit adalah sebuah framework berbasis python dan bersifat open-source yang dibuat untuk memudahkan dalam membangun aplikasi web di bidang sains data dan machine learning yang interaktif. Salah satu hal menarik dari framework ini adalah tidak perlu keahlian dalam HTML, CSS, atau javascript, bisa langsung dibuat dengan python dalam beberapa baris kode.

Streamlit adalah kerangka kerja web yang ditujukan untuk menyebarkan model dan visualisasi dengan mudah menggunakan bahasa python, yang cepat dan minimalis tetapi juga memiliki tampilan yang cukup baik serta ramah pengguna. Selain itu, streamlit adalah framework berbasis python yang digunakan untuk membangun antarmuka pengguna grafis (GUI) dengan cara sederhana dan cepat. Streamlit memungkinkan pengembang membuat aplikasi berbasis data, analitik dan machine learning hanya dengan beberapa baris python, tanpa menggunakan HTML, CSS, atau Javascript. Keunggulannya adalah kemampuannya dalam membuat

aplikasi interaktif yang dapat menampilkan visualisasi data, menrima input pengguna dan menampilkan hasil komputasi dengan mudah [23].

BAB III

ANALISIS PERANCANGAN SISTEM

3.1 Analisis sistem

Analisis sistem merupakan tahap awal yang penting dalam penelitian klasifikasi status kesehatan menggunakan algoritma Naive Bayes dan K-Nearest Neighbor (K-NN). Tahapan ini bertujuan untuk mengidentifikasi permasalahan, seperti perlunya metode yang akurat dan efisien dalam menentukan status kesehatan berdasarkan data yang tersedia. Pemilihan algoritma Naive Bayes dan K-NN didasarkan pada perbedaan pendekatan keduanya dalam proses klasifikasi, sehingga memungkinkan dilakukan analisis perbandingan performa yang komprehensif. Analisis ini mencakup identifikasi kebutuhan penelitian, persiapan dan pengolahan dataset, serta perancangan eksperimen di *Google Colaboratory* untuk menguji dan membandingkan kinerja kedua algoritma. Selain itu, hasil dari perbandingan diintegrasikan ke dalam aplikasi berbasis Streamlit sehingga pengguna dapat melakukan prediksi status kesehatan secara interaktif dan mudah diakses melalui antarmuka web yang sederhana[24].

3.1.1 Analisis Masalah

Masalah pokok dalam penelitian ini adalah perlunya metode yang akurat dan efisien dalam mengklasifikasikan status kesehatan, mengingat kesehatan merupakan faktor penting yang berpengaruh langsung terhadap kualitas hidup seseorang. Gaya hidup memiliki pengaruh yang besar terhadap status kesehatan seseorang. Gaya hidup tidak sehat, seperti pola makan yang buruk, kurangnya aktivitas fisik, serta kebiasaan yang merugikan kesehatan, dapat menurunkan

kualitas kesehatan dan meningkatkan risiko munculnya berbagai penyakit. Sebaliknya, gaya hidup sehat dengan menjaga pola makan bergizi seimbang, rutin melakukan aktivitas fisik, serta menghindari kebiasaan yang berisiko, menjadi faktor penting dalam menjaga kesehatan tubuh. Menjaga asupan makanan bergizi dan melakukan aktivitas fisik secara teratur merupakan fondasi utama dalam mewujudkan derajat kesehatan yang optimal. Sayangnya, banyak individu yang masih kurang peduli untuk meluangkan waktu dalam menerapkan pola hidup sehat tersebut [25].

Ketersediaan data kesehatan yang semakin melimpah membuka peluang untuk memanfaatkan teknologi data mining dan machine learning sebagai solusi prediktif. Dalam penelitian ini digunakan algoritma Naive Bayes dan K-Nearest Neighbor (K-NN), yang memiliki karakteristik dan pendekatan berbeda dalam proses klasifikasi. Perbandingan kedua algoritma ini diharapkan dapat menghasilkan gambaran yang jelas mengenai metode mana yang memiliki kinerja lebih baik dalam menentukan status kesehatan, sehingga dapat memberikan kontribusi pada pengembangan metode prediksi yang efektif dan berbasis data.

3.1.2 Analisis Kebutuhan Sistem

Berikut adalah uraian komponen-komponen yang diperlukan guna mendukung perbandingan algoritma Naive Bayes dan K-NN.

a. *Hardware*

Hardware (perangkat keras) merupakan komponen utama dalam membangun sebuah sistem. Spesifikasi hardware yang dibutuhkan untuk

menunjang perbandingan algoritma Naive bayes dan K-NN untuk mengklasifikasikan status kesehatan adalah sebagai berikut :

1. Processor dengan spesifikasi minimum, intel core i3 / AMD Ryzen 3 atau setara
2. RAM dengan spesifikasi minimum 4GB, disarankan 8GB untuk proses lebig cepat
3. Penyimpanan 256GB HDD/ SSD
4. Koneksi internet dengan kecepatan stabil minimal 10 Mbps
5. Perangkat tambahan berupa, keyboard, Mouse, dan Layar Monitor.

b. Software

Software merupakan perangkat lunak komputer yang berfungsi sebagai jembatan penghubung antara sistem dengan pengguna. Dalam konteks penelitian ini, software memiliki peran penting dalam mewujudkan perbandingan algoritma Naive bayes untuk mengklasifikasikan status kesehatan. Perangkat lunak yang digunakan membantu dalam proses pengolahan data, penerapan algoritma klasifikasi, serta evaluasi hasil analisis.

1. Google chrome digunakan untuk mengakses platform *google colaboratory*
2. Google colaboratory platform berbasis cloud untuk menulis dan mengeksekusi kode python secara interaktif

3.1.3 Analisis Data

Analisis data pada penelitian ini menggunakan dataset yang diambil dari situs *kaggle .com,* yang dikenal sebagai salah satu sumber terpercaya dalam bidang

data science dan machine learning. Dataset tersebut mencakup 389 catatan *record*. Terdapat sembilan atribut didalam dataset ini dan atribut ‘hasil’ adalah label dari kesembilan atribut yang ada. Berikut adalah beberapa atribut dalam dataset status yang digunakan untuk menjalankan pengklasifikasian.

Atribut yang digunakan dalam mengklasifikasikan status gizi balita adalah sebagai berikut:

1. Usia

Atribut ini mengelompokkan data berdasarkan kategori umur, seperti "Muda", yang memberikan gambaran tentang rentang usia responden atau subjek dalam penelitian. dimana usia merupakan faktor penting karena dapat memengaruhi kondisi kesehatan, pola hidup, dan risiko terhadap berbagai penyakit. Kriteria usia dalam penelitian ini dikelompokkan dimana usia 32 tahun adalah batasan orang dikatakan muda, sementara usia orang yang tergolong tua adalah yang berusia di atas 54 tahun [26].

Tabel 3. 1 Atribut Usia

No	Usia	Jumlah
1	Muda	185
2	Tua	204

2. Jenis Kelamin

Atribut ini mencerminkan jenis kelamin dari subjek data, baik laki-laki (Pria) maupun perempuan (Wanita). Perbedaan jenis kelamin dapat

mempengaruhi kebutuhan kalori, metabolisme, serta risiko terhadap penyakit tertentu.

Tabel 3. 2 Atribut Jenis Kelamin

No	Jenis Kelamin	Jumlah
1	Pria	114
2	Wanita	275

3. Merokok

Atribut ini menunjukkan kebiasaan merokok dari responden, dikategorikan sebagai "Aktif", "Pasif", atau tidak merokok. Selanjutnya variabel kebiasaan merokok.

Perokok aktif adalah seseorang yang mengkonsumsi rokok secara rutin atau tidak, bahkan hanya sesekali atau coba-coba, termasuk jika hanya menghembuskan asap tanpa mengispnya ke paru-paru.

Perokok pasif adalah orang yang tidak merokok namun terpapar asap rokok dari lingkungan sekitar, misalnya di ruangan tertutup bersama perokok [27].

Tabel 3. 3 Atribut Merokok

No	Merokok	Jumlah
1	Aktif	209
2	Pasif	180

4. Bekerja

Atribut ini menunjukkan apakah subjek memiliki pekerjaan atau tidak. Aktivitas bekerja juga menjadi variabel penting. Bekerja (yes) yaitu seseorang yang sedang memiliki pekerjaan, baik secara tetap maupun tidak tetap, termasuk pekerja penuh waktu, paruh waktu dan buruh harian lepas. Termasuk juga dalam kategori ini adalah mereka yang membantu usaha keluarga tanpa upah. Untuk tidak bekerja (No) yaitu seseorang yang belum memiliki pekerjaan, namun aktif mencari kerja, atau sedang mempersiapkan usaha baru [28].

Tabel 3. 4 Atribut Bekerja

No	Bekerja	Jumlah
1	No	155
2	Yes	234

5. Aktivitas Begadang

Atribut ini menggambarkan kebiasaan begadang atau kurang tidur dari responden. Seseorang dianggap begadang jika tetap terjaga hingga larut malam atau dini hari, melebihi rentang waktu tidur yang ideal antara pukul 22.00 hingga 06.00 [29].

Tabel 3. 5 Atribut Aktivitas Begadang

No	Aktivitas Begadang	Jumlah
1	Iya	223
2	Tidak	166

6. Aktivitas Olahraga

Atribut ini mengukur seberapa sering responden melakukan aktivitas fisik, seperti "Jarang" atau "Sering". Seseorang dikatakan jarang berolahraga jika hanya melakukannya 1 hingga 2 kali per minggu atau kurang dari 150 menit dalam seminggu, dengan durasi kurang dari 30 menit per sesi dan aktivitas berintensitas ringan hingga sedang seperti jalan santai atau yoga ringan.

Sebaliknya, aktivitas olahraga yang rutin ditandai dengan frekuensi minimal 3 kali per minggu atau lebih dari 150 menit per minggu. Durasi minimal 30 menit per sesi dengan jenis olahraga berintensitas sedang hingga tinggi, seperti jogging, bersepeda, berenang, atau latihan beban, yang berdampak pada peningkatan daya tahan tubuh, kekuatan otot, dan kesehatan jantung [30].

Tabel 3. 6 Atribut Aktivitas Olahraga

No	Aktivitas Olahraga	Jumlah
1	Jarang	225
2	Sering	164

7. Pola makan teratur

Atribut ini menunjukkan keteraturan dalam pola makan responden, seperti makan secara "Teratur" atau "Kurang". Pola makan teratur dicirikan dengan makan utama yang tidak pernah dilewatkan, jadwal yang konsisten, serta komposisi nutrisi yang seimbang dengan asupan cairan yang cukup, minimal 8 gelas air putih setiap hari. Sebaliknya, pola makan tidak teratur

ditandai dengan frekuensi makan yang berantakan, sering melewatkan waktu makan, serta konsumsi makanan tinggi gula, garam, atau lemak jenuh tanpa memperhatikan keseimbangan gizi [31].

Tabel 3. 7 Atribut Pola Makan Teratur

No	Pola Makan Teratur	Jumlah
1	Teratur	272
2	Kurang	117

8. Penyakit bawaan

Atribut ini mencatat apakah responden memiliki penyakit bawaan, yang dapat memengaruhi asupan dan penyerapan gizi, serta respons tubuh terhadap nutrisi. Seseorang dikategorikan memiliki penyakit bawaan jika memiliki riwayat penyakit sejak lahir atau akibat faktor genetik, serta mendapat diagnosis dari tenaga medis sebagai penyakit kronis atau berulang. Sebaliknya, individu yang tidak memiliki riwayat penyakit kronis sejak lahir, tidak memiliki faktor keturunan yang menyebabkan kondisi medis tertentu, serta tidak pernah didiagnosis penyakit bawaan oleh tenaga medis, dikategorikan sebagai tidak memiliki penyakit bawaan [32].

Tabel 3. 8 Atribut Penyakit Bawaan

No	Penyakit Bawaan	Jumlah
1	Tidak Ada	149
2	Ada	240

9. Hasil

Atribut ini merupakan label atau target klasifikasi dalam penelitian, menunjukkan apakah seseorang berada dalam kondisi "Ya" atau "Tidak".

Tabel 3. 9 Atribut Hasil

No	Hasil	Jumlah
1	Ya	186
2	Tidak	203

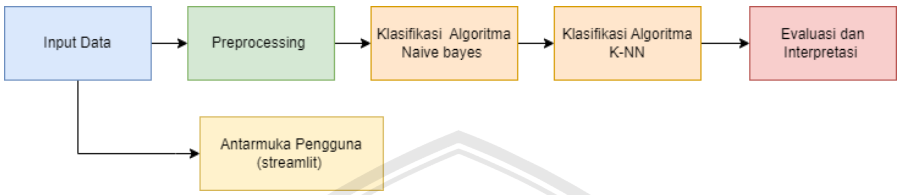
3.2 Perancangan sistem

Tahap perancangan sistem memegang peranan penting dalam penelitian Perbandingan Algoritma Naive Bayes dan K-NN untuk Mengklasifikasikan Status Kesehatan. Tahapan ini bertujuan untuk memvisualisasikan alur proses pengolahan data, mulai dari tahap pengumpulan dan pemrosesan dataset, pembagian data menjadi data latih dan data uji, penerapan algoritma Naive Bayes dan K-NN, hingga menghasilkan output berupa hasil klasifikasi dan evaluasi kinerja masing-masing algoritma [6].

3.2.1 Arsitektur Aplikasi

Arsitektur sistem untuk klasifikasi status kesehatan menggunakan algoritma Naive Bayes dan K-NN dirancang dengan fokus pada keakuratan hasil klasifikasi serta kemudahan implementasi pada lingkungan pemrograman Google Colab. Sistem ini terdiri dari beberapa komponen utama yang saling terintegrasi untuk menghasilkan prediksi yang andal. Komponen pertama adalah modul input data, yang menerima dan memverifikasi dataset status kesehatan. Informasi yang

digunakan meliputi variabel-variabel seperti tekanan darah, kadar gula, indeks massa tubuh (BMI), usia, serta parameter kesehatan lainnya yang relevan. Modul ini juga menangani proses pra-pemrosesan data, seperti pembersihan data (data cleaning), normalisasi, dan pembagian data menjadi data latih dan data uji untuk memastikan kualitas serta konsistensi data sebelum dilakukan proses klasifikasi.



Gambar 3. 1 Arsitektur Aplikasi

Diagram alur sistem untuk klasifikasi status kesehatan menggunakan algoritma Naive Bayes dan K-NN dirancang untuk menggambarkan proses penelitian dari tahap awal hingga evaluasi hasil. Berikut uraian dari masing-masing komponen dalam diagram tersebut:

1. Input Data
    a. Tahap awal di mana dataset status kesehatan dimasukkan ke dalam lingkungan Google Colab.
    b. Data ini memuat variabel-variabel kesehatan seperti usia, jenis kelamin, merokok, bekerja, aktivitas begadang, aktivitas olahraga, pola makan, penyakit bawaan dan hasil sebagai label.
2. Preprocessing
    a. Tahap praproses data bertujuan untuk membersihkan dan mempersiapkan data agar siap digunakan oleh model klasifikasi.

b. Proses ini mencakup:

1) Pembersihan data *(data cleaning)*

Pembersihan data adalah tahap awal dalam preprocessing untuk memastikan dataset yang digunakan bebas dari kesalahan atau ketidakkonsistenan. Proses ini meliputi: Menghapus data duplikat (baris yang sama persis muncul berulang). Memperbaiki kesalahan penulisan (misalnya "Male", "male", "M" diseragamkan). Menghilangkan data yang tidak relevan atau outlier (nilai yang terlalu ekstrem dan tidak masuk akal). Memastikan format data seragam (misalnya tanggal dengan format DD/MM/YYYY, bukan bercampur dengan MM-DD-YYYY). Tujuannya agar data lebih konsisten dan siap diproses tanpa menimbulkan bias atau error

2) Penanganan nilai kosong *(missing value handling)*

Dalam dataset sering ditemukan nilai yang hilang (kosong). Hal ini bisa terjadi karena kesalahan input, data tidak tercatat, atau masalah teknis. Ada beberapa cara menanganinya: Menghapus baris/kolom yang banyak nilai kosongnya (jika jumlahnya sedikit dan tidak berpengaruh). Mengisi (imputasi) dengan nilai tertentu, misalnya: Rata-rata (mean) untuk data numerik. Median jika ada outlier. Modus untuk data kategorikal. Menggunakan algoritma khusus yang bisa menangani missing value. Tujuannya agar model tidak bias atau error akibat data yang tidak lengkap.

3) Normalisasi atau standarisasi data

Normalisasi data adalah proses mengubah skala nilai pada setiap fitur agar berada dalam rentang yang sama, biasanya antara 0 sampai 1. Tujuannya adalah supaya tidak ada fitur yang memiliki pengaruh lebih besar hanya karena skala angkanya lebih tinggi. Misalnya, atribut umur memiliki nilai asli dalam satuan tahun, seperti tua dan muda. Jika langsung digunakan, nilai umur akan lebih besar dibandingkan atribut lain yang berbentuk kategori, seperti jenis kelamin (laki-laki/perempuan) atau kebiasaan merokok (ya/tidak). Demikian pula dengan atribut aktivitas begadang, aktivitas olahraga, pola makan teratur, dan penyakit bawaan yang umumnya dinyatakan dalam bentuk kategorikal (misalnya 0 untuk tidak, 1 untuk ya). Melalui normalisasi, nilai numerik seperti umur diubah ke dalam skala yang sama dengan atribut kategorikal, biasanya dalam rentang 0 sampai 1. Dengan begitu, perbedaan skala antaratribut tidak menimbulkan bias saat proses klasifikasi. Atribut lain seperti merokok, olahraga, atau pola makan teratur juga sudah berada dalam skala kecil (0 dan 1), sehingga hasil akhirnya semua atribut dapat diperlakukan secara seimbang oleh algoritma.

4) Pembagian data menjadi data latih dan data uji

Setelah data siap, dataset dibagi menjadi dua: Data Latih digunakan untuk melatih model agar mengenali pola dari data. Data Uji digunakan untuk menguji sejauh mana model mampu melakukan prediksi pada

data baru yang belum pernah dilihat sebelumnya. Biasanya pembagian dilakukan dengan perbandingan: 70:30 (70%) untuk latih, 30% untuk uji. 80:20 (80%) latih, 20% uji. Tujuannya adalah mengukur performa model agar tidak hanya bagus pada data latih (*overfitting*), tapi juga pada data uji yang baru.

3. Klasifikasi Algoritma Naive Bayes dan K-NN
    a. Data yang telah diproses diklasifikasikan menggunakan dua algoritma: Naive Bayes dan K-Nearest Neighbor (K-NN).
    b. Kedua algoritma akan memprediksi status kesehatan responden berdasarkan variabel-variabel yang diberikan.
4. Evaluasi & Interpretasi
    a. Hasil prediksi dievaluasi menggunakan metrik evaluasi seperti:
        1) Akurasi
        2) Presisi
        3) Recall
        4) F1-Score
    b. Interpretasi hasil ini digunakan untuk menilai kinerja masing-masing algoritma serta membandingkan performa keduanya.
5. Antar muka Pengguna (streamlit)
    a. *Streamlit* adalah *framework Python* yang digunakan untuk membangun antarmuka pengguna secara cepat dan interaktif.
    b. Fungsi dari diagram ini:
        1) Menerima input dari pengguna

2) Menampilkan hasil prediksi

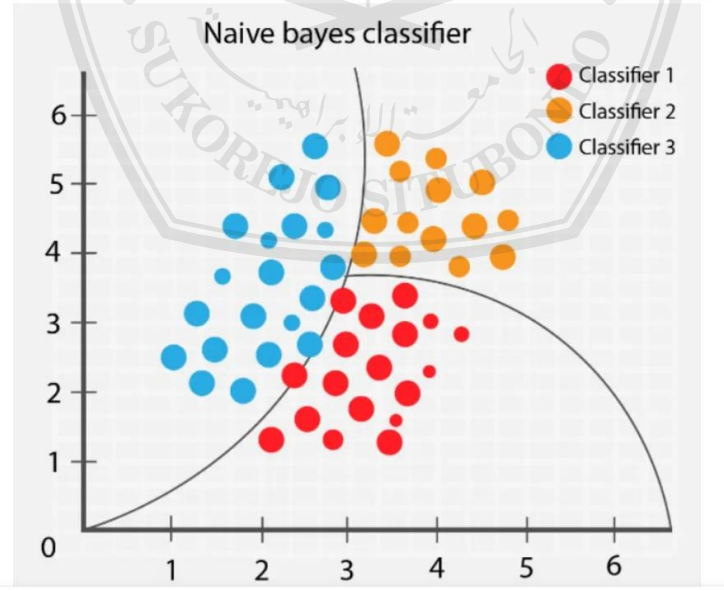
3) Menyediakan tampilan yang mudah digunakan untuk menjalankan seluruh proses dari input hingga evaluasi

3.2.2 Perancangan Model Machine Learning

Perancangan model machine learning menggunakan dua algoritma sebagai berikut :

1. Algoritma Naive Bayes

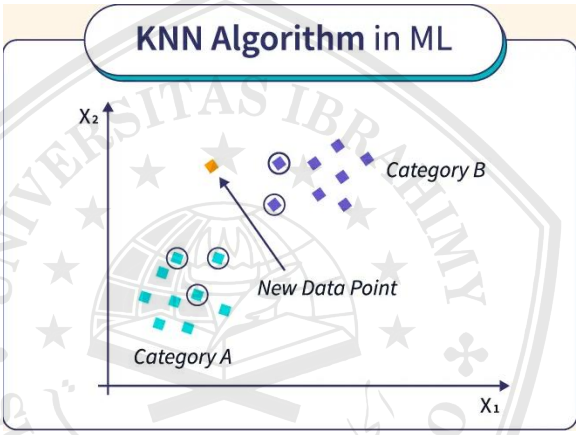
Naive Bayes adalah algoritma klasifikasi yang berbasis pada teorema Bayes, dengan asumsi bahwa setiap fitur bersifat independen satu sama lain. Algoritma ini menghitung probabilitas suatu data masuk ke dalam kelas tertentu berdasarkan nilai atribut yang dimilikinya. Naive Bayes banyak digunakan karena sederhana, cepat, dan cukup efektif, terutama pada dataset berukuran besar [33].



Gambar 3. 2 Model Algoritma Naive Bayes

2. Algoritma K-NN

K-Nearest Neighbor (K-NN) adalah algoritma klasifikasi yang bekerja dengan cara mencari sejumlah tetangga terdekat dari data baru berdasarkan jarak tertentu (umumnya Euclidean distance). Data baru tersebut kemudian diklasifikasikan ke dalam kelas yang paling banyak muncul dari tetangga terdekatnya. K-NN termasuk metode yang sederhana namun efektif, terutama untuk dataset dengan distribusi yang jelas [34].



Gambar 3. 3 Model Algoritma K-NN

3.3 Implementasi Sistem

3.3.1 Library yang digunakan

1. *NumPy*

NumPy merupakan pustaka dasar yang sangat penting untuk melakukan komputasi numerik di *Python.* Dalam proyek ini, *NumPy* dimanfaatkan untuk memanipulasi array dan menjalankan berbagai operasi matematika secara efisien. *Library* ini menyediakan struktur data berupa array multi-dimensi yang sangat kuat, sehingga sangat berguna untuk merepresentasikan dan mengolah data numerik dalam skala besar. Selain itu, *NumPy* juga dilengkapi dengan

beragam fungsi matematika dan statistik yang dapat diterapkan langsung pada array, memungkinkan pelaksanaan operasi vektor dan matriks secara cepat dan efisien.

2. Pandas

Pandas merupakan *library* yang krusial dalam analisis dan manipulasi data menggunakan *Python*. Pada proyek ini, Pandas digunakan untuk memuat, mengelola, dan memodifikasi dataset diabetes. *Library* ini menyediakan struktur data seperti Data*Frame* dan *Series* yang memungkinkan pengolahan data terstruktur secara efisien. Dengan berbagai fungsi yang dimilikinya, Pandas mempermudah proses pembacaan file CSV, manipulasi data, penanganan data yang hilang *(missing values),* hingga agregasi data. Hal ini sangat berguna terutama dalam tahap pra-pemrosesan data sebelum dilakukan analisis lebih lanjut.

3. *Scikit-learn*

Scikit-learn merupakan pustaka *machine learning* yang lengkap dan banyak digunakan dalam bahasa *Python*. Dalam proyek ini, *scikit-learn* dimanfaatkan untuk berbagai keperluan *machine learning,* seperti pra-pemrosesan data, pembagian dataset, pembuatan model, hingga evaluasi performa model. *Library* ini menyediakan beragam algoritma *machine learning,* termasuk algoritma Naïve Bayes dan K-NN. Selain itu, *scikit-learn* juga menyediakan alat evaluasi model seperti confusion matrix, akurasi, presisi, recall, dan F1-score. Dengan *scikit-learn,* proses membangun, melatih, dan

mengevaluasi model prediksi diabetes dapat dilakukan secara praktis dan efisien.

4. ***Matplotlib***

Matplotlib merupakan *library* visualisasi data yang kuat dan fleksibel dalam ekosistem *Python.* Dalam proyek ini, *Matplotlib* digunakan untuk membuat berbagai visualisasi grafik, seperti histogram distribusi kelas dan plot *confusion matrix. Library* ini mendukung pembuatan beragam jenis grafik, mulai dari yang sederhana hingga yang lebih kompleks. *Matplotlib* juga menyediakan kontrol penuh terhadap elemen-elemen visualisasi, sehingga Anda dapat menyesuaikan tampilan grafik sesuai dengan kebutuhan analisis dan presentasi data.

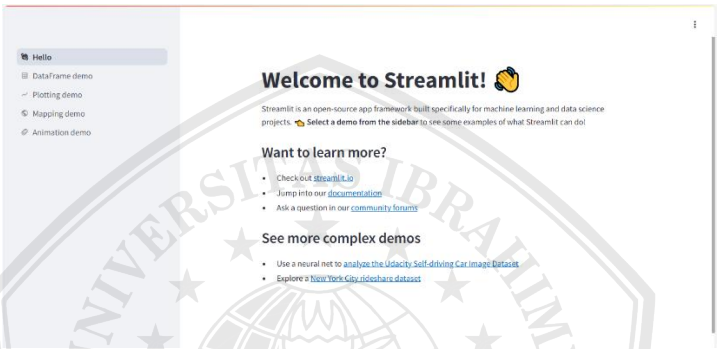
5. ***Seaborn***

Seaborn merupakan *library* visualisasi statistik yang dikembangkan di atas *Matplotlib*. Dalam proyek ini, Seaborn dimanfaatkan untuk membuat heatmap dari *confusion matrix.* Library ini menawarkan antarmuka tingkat tinggi yang memudahkan pembuatan grafik statistik yang informatif dan menarik secara visual. *Seaborn* sangat berguna untuk menyajikan visualisasi data yang lebih kompleks dan estetis, berkat fitur-fitur seperti palet warna yang telah disesuaikan serta pengaturan tema yang intuitif dan mudah diterapkan.

3.3.2 Implementasi Streamlit

Penelitian ini memanfaatkan *Streamlit,* sebuah *framework open-source* berbasis *Python,* untuk menampilkan hasil klasifikasi penyakit diabetes dalam bentuk aplikasi *web* yang interaktif dan *user-friendly.* Proses implementasi

Streamlit diawali dengan menginstal framework menggunakan perintah "*pip install streamlit*" melalui terminal. Setelah semua komponen tersedia, aplikasi dapat dijalankan dengan perintah "*streamlit run sehat.py.*" Streamlit kemudian akan membuka aplikasi secara otomatis di browser default, dengan alamat url localhost:8502 memungkinkan pengguna untuk memasukkan data dan memperoleh hasil prediksi secara langsung dan *real-time*.



Gambar 3. 4 Aplikasi Streamlit

a. Input Data Pengguna

Pengguna diminta untuk mengisi sejumlah parameter yang relevan dengan risiko diabetes. Parameter ini diadaptasi dari *Pima Indians Diabetes Dataset* dan terdiri atas 9 fitur utama, yaitu:

1) Usia
2) Jenis kelamin
3) Merokok
4) Bekerja
5) Aktivitas begadang
6) Aktivitas olahraga
7) Pola makan teratur

8) Penyakit bawaan

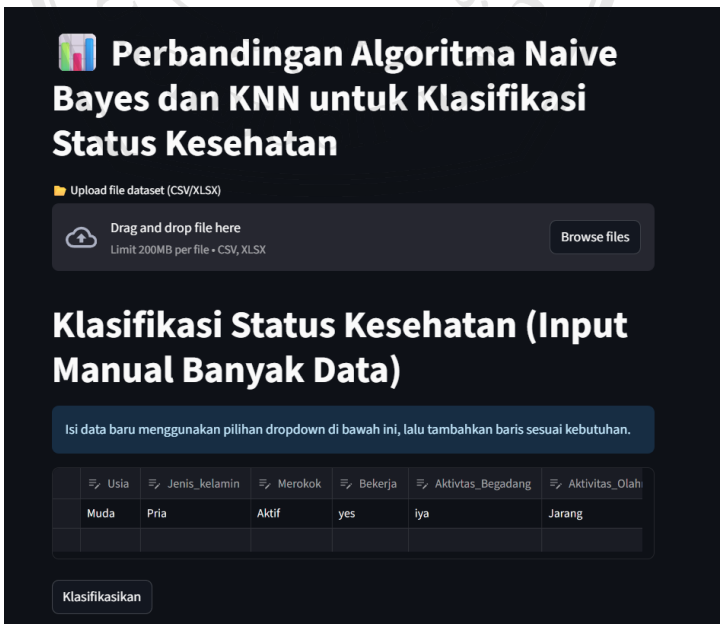
b. Proses Klasifikasi Model Algoritma

Setelah pengguna mengisi semua data yang diminta, informasi tersebut akan dianalisis oleh model klasifikasi yang sebelumnya telah dilatih menggunakan algoritma Naïve Bayes dan K-NN. Model ini akan menentukan apakah data yang diberikan termasuk dalam kategori "Tidak" artinya tidak ada gangguan (Sehat) atau "Ya" artinya ada gangguan (Tidak sehat), berdasarkan pola-pola yang telah dikenali dari data pelatihan.

c. output : Hasil klasifikasi

Setelah pemrosesan selesai, hasil prediksi akan ditampilkan secara langsung di halaman aplikasi. Output ditampilkan dalam bentuk:

1) Hasil prediksi model kedua algoritma, Tidak artinya sehat (tidak ada gangguan) atau Ada artinya tidak sehat (ada gangguan).



Gambar 3. 5 Output Hasil Klasifikasi

3.3.3 Uji Coba Sistem Oleh Pengguna (user validator)

Untuk memastikan bahwa sistem prediksi yang dibangun dapat digunakan secara efektif dan mudah dipahami oleh pengguna umum, dilakukan proses uji coba awal *(user validation)* kepada beberapa mahasiswa sebagai representasi pengguna non-teknis. Tujuan dari tahap ini adalah untuk mengetahui apakah sistem sudah berjalan dengan baik, mudah digunakan, serta memberikan hasil yang dapat dimengerti oleh pengguna awam. Responden merupakan 3 mahasiswa dari program studi Teknologi Informasi, yang belum pernah menggunakan *Streamlit* atau sistem prediksi serupa sebelumnya, dan 2 mahasiswa sudah pernah menggunakan *streamlit* atau sistem prediksi. Adapun proses uji coba sebagai berikut:

a. Mahasiswa diberikan akses ke aplikasi prediksi berbasis *web.*

b. Mereka diminta untuk mengisi data secara mandiri sesuai dengan parameter yang tersedia.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini berasal dari *platform* https://www.kaggle.com/ ,sebuah situs penyedia data terbuka yang sangat populer dan banyak digunakan oleh komunitas data *science* di seluruh dunia. Penelitian ini memanfaatkan data sekunder yang telah tersedia di *Kaggle,* sehingga proses pengumpulan data menjadi lebih efisien dan praktis. Dengan menggunakan dataset dari sumber terpercaya ini, penelitian dapat fokus pada analisis dan pengolahan data secara optimal untuk mencapai hasil yang akurat dan relevan.

(https://www.kaggle.com/datasets/rustaas/kebiasaan-buruk-berdampak-ke-kesehatan) *Kaggle* adalah sebuah platform daring yang sangat populer di kalangan praktisi data *science*, peneliti, maupun pengembang machine learning ataupun *deep learnng*. *Platform* ini menyediakan berbagai macam dataset yang dapat diakses secara gratis oleh siapa saja yang ingin belajar, melakukan eksperimen, atau mengembangkan model analisis data. Selain sebagai sumber data, *Kaggle* juga menjadi wadah bagi komunitas global untuk berbagi pengetahuan, mengikuti kompetisi, serta berdiskusi mengenai berbagai topik terkait data *science* dan *machine learning.* Banyak perusahaan dan institusi ternama juga memanfaatkan *Kaggle* untuk mengadakan lomba pemecahan masalah nyata

berbasis data, sehingga *platform* ini menjadi sangat relevan bagi siapa saja yang ingin mengasah kemampuan analisis data dan algoritma prediksi.

Dalam penelitian ini, data yang digunakan diambil dari salah satu dataset yang tersedia di Kaggle, yaitu dataset *Lifestyle Habits*. Dataset ini sering digunakan dalam penelitian dan pengembangan model klasifikasi kesehatan karena memiliki struktur data yang jelas dan jumlah sampel yang memadai. Secara spesifik, dataset ini terdiri dari 389 entri, di mana setiap baris mewakili satu responden. Setiap sampel memiliki beberapa kolom fitur prediktor dan satu kolom target. Fitur prediktor mencakup berbagai parameter kebiasaan hidup dan kondisi kesehatan seperti usia, jenis kelamin, merokok, bekerja, aktivitas begadang, aktivitas olahraga, pola makan teratur dan penyakit bawaan, dengan satu

Sementara itu, kolom target atau label pada dataset ini adalah “hasil”, yang menunjukkan status kesehatan masing-masing individu. Dengan struktur data seperti ini, dataset *Lifestyle Habits* dari Kaggle sangat cocok digunakan untuk membangun dan menguji model klasifikasi di bidang kesehatan, khususnya untuk memprediksi status kesehatan berdasarkan kebiasaan hidup menggunakan metode *machine learning*.

Dataset ini memuat 389 data indiidu yang masing-masing memiliki 9 fitur penting yang digunakan dalam proses klasifikasi status kesehatan. fitur-fitur tersebut antara lain: usia, jenis kelamin, merokok, bekerja, aktivitas begadang, aktivitas olahraga, pola makan teratur, penyakit bawaan, dan hasil sebagai label kelas.

Tabel 4. 1 Dataset

No	Usia	Jenis Kelamin	Merokok	Bekerja	Aktivtas Begadang	Aktivitas Olahraga	Pola Makan Teratur	Penyakit Bawaan	Hasil
1.	Muda	Pria	Aktif	No	Iya	Jarang	Teratur	Tidak ada	Tidak
2.	Muda	Wanita	Pasif	Yes	Tidak	Sering	Kurang	Ada	Ya
3.	Muda	Wanita	Pasif	Yes	Tidak	Sering	Kurang	Ada	Ya
4.	Muda	Pria	Aktif	No	Iya	Sering	Kurang	Tidak ada	Ya
5.	Muda	Pria	Aktif	No	iya	Jarang	Teratur	Ada	Tidak

Tahap pertama dalam penelitian ini adalah melakukan preprocessing data, yaitu proses menyiapkan dataset agar siap untuk dianalisis secara lebih mendalam. Dataset yang digunakan berasal dari *Lifestyle Habits* yang disimpan dalam format Excel (.xlsx). Untuk memuat file Excel tersebut ke dalam lingkungan *Google Colab*, digunakan perintah pemrograman Python dengan pustaka pandas, sehingga data dapat dibaca dalam bentuk dataframe dan siap diproses lebih lanjut.

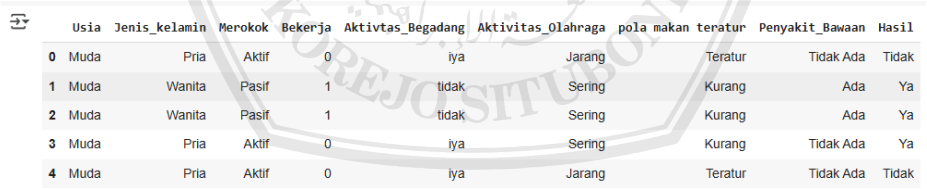
```
dataset =pd.read_excel('predic_tablle.xlsx')
dataset.head()
```

Segmen Program 4. 1 *Load Dataset*

menyediakan fungsi *read_excel* yang sangat berguna untuk membaca dan mengimpor data dari file excel ke dalam struktur *data frame*, sehingga memudahkan pengolahan data selanjutnya. dari dataset. Fungsi *head()* dari DataFrame digunakan untuk menampilkan beberapa baris awal dari dataset. Hal ini memungkinkan untuk melihat sekilas fitur-fitur yang ada dalam dataset dan tipe datanya. Dari output yang ditampilkan, terlihat bahwa dataset diabetes terdiri dari beberapa fitur seperti usia, jenis kelamin,

merokok, bekerja, aktivitas begadang, aktivitas olahraga, pola makan teratur, penyakit bawaan dan hasil.

Setelah dataset berhasil dimuat, tahap berikutnya adalah melakukan eksplorasi untuk memahami struktur dan isi data secara menyeluruh. Proses ini meliputi pemeriksaan tipe variabel, distribusi nilai, serta identifikasi nilai yang hilang atau anomali. Eksplorasi data membantu mengenali pola, hubungan antar variabel, dan potensi masalah dalam dataset sehingga analisis selanjutnya dapat dilakukan dengan lebih akurat dan efektif. Dengan memahami karakteristik data secara mendalam, kita dapat menentukan metode pemrosesan dan teknik analisis yang paling sesuai untuk mencapai hasil yang optimal. Tahapan eksplorasi ini sangat penting sebagai fondasi awal dalam proses analisis data yang sistematis dan terstruktur.



	Usia	Jenis_kelamin	Merokok	Bekerja	Aktivtas_Begadang	Aktivitas_Olahraga	pola makan teratur	Penyakit_Bawaan	Hasil
0	Muda	Pria	Aktif	0	iya	Jarang	Teratur	Tidak Ada	Tidak
1	Muda	Wanita	Pasif	1	tidak	Sering	Kurang	Ada	Ya
2	Muda	Wanita	Pasif	1	tidak	Sering	Kurang	Ada	Ya
3	Muda	Pria	Aktif	0	iya	Sering	Kurang	Tidak Ada	Ya
4	Muda	Pria	Aktif	0	iya	Jarang	Teratur	Tidak Ada	Tidak

Gambar 4. 1 Tampilan Dataset

Setelah melaukukan eksplorasi awal terhadap dataset lifestyle habits, langkah selanjutnya adalah melakukan distribusi data untuk fitur *" Hasil"* fitur *"Hasil"* merupakan variabel target yang menunjukkan apakah seseorang di diagnosis sehat atau tidak.

Untuk memvisualisasikan distribusi data “Hasil”, digunakan *library matplotib* dan *PyPlot* dalam Phyton. Pertama, dilakukan perhitungan jumlah data untuk setiap nilai dalam fitur “Hasil”.

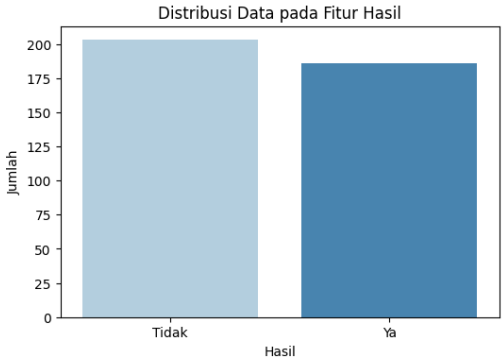
```
import matplotlib.pyplot as plt
import seaborn as sns

# Distribusi data untuk fitur Hasil
plt.figure(figsize=(6,4))
sns.countplot(x='Hasil', data=dataset, palette='Blues')

plt.title("Distribusi Data pada Fitur Hasil")
plt.xlabel("Hasil")
plt.ylabel("Jumlah")
plt.show()
```

Segmen Program 4. 2 Distribusi Data

Grafik ditampilkan menggunakan fungsi *plt.show()*. Dari grafik yang dihasilkan, terlihat bahwa jumlah data dengan nilai “Tidak” (artinya sehat/ Tidak ada gangguan) lebih banyak dibandingkan dengan jumlah data dengan “ ya” (artinya tidak sehat / ada gangguan).



Gambar 4. 2 Visualisasi Distribusi

Dataset ini menyajikan campuran variabel numerik dan kategorik, dengan "Hasil" sebagai variabel target yang menunjukkan hasil status kesehatan. Sebelum melakukan analisis lebih lanjut, penting untuk memahami karakteristik dan distribusi detiap fitur dalam dataset. Untuk kategorik "Hasil" distribusinya sebagai berikut :

1. Kelas 0 (Tidak / artinya sehat atau tidak ada gangguan) = 200 sampel
2. Kelas 1 (ya / artinya tidak sehat atau ada gangguan) = 179 sampel

4.2 *Preprocessing*

4.2.1 *Missing Value*

Dilakukan pengecekan terhadap missing values pada dataset dengan menggunakan fungsi *isnull()* dari pandas yang diikuti dengan fungsi *sum()*. Fungsi *isnull()* akan memeriksa setiap sel dalam dataset dan memberikan nilai *True* apabila terdapat data yang kosong *(null)* serta *False* jika data terisi. Selanjutnya, fungsi *sum()* digunakan untuk menghitung jumlah nilai *True* pada setiap kolom, sehingga diperoleh total jumlah data yang hilang pada masing-masing fitur. Hasil dari perintah ini menampilkan daftar setiap kolom dalam dataset beserta jumlah *missing values* yang dimilikinya. Informasi ini sangat penting untuk mengetahui kualitas data dan menentukan langkah selanjutnya, seperti melakukan penghapusan data yang hilang atau menggantinya dengan teknik imputasi.

```
[ ]  dataset.isnull().sum ()
```

Gambar 4. 3 *Missing Value*

4.2.2 Label Encoding

Setelah proses missing values dilakukan, langkah selanjutnya dalam pra-pemrosesan adalah *encoding*, yaitu proses mengubah data kategorikal menjadi bentuk numerik agar dapat diproses oleh algoritma machine learning. Hal ini penting karena sebagian besar algoritma klasifikasi, termasuk Naive Bayes dan K-Nearest Neighbor (K-NN), hanya dapat bekerja dengan data numerik. Data kategori seperti "Pria" dan "Wanita" harus diubah menjadi nilai numerik, misalnya 0 dan 1, agar sistem dapat mengenalinya dan melakukan perhitungan. Dalam penelitian ini, proses encoding dilakukan menggunakan *LabelEncoder* dari *library sklearn.preprocessing*. Potongan kode berikut digunakan untuk melakukan encoding pada kolom yang bertipe kategori.

```
Encoding

[4] from sklearn.preprocessing import LabelEncoder

en = LabelEncoder()

dataset['Usia'] = en.fit_transform(dataset['Usia'])
dataset['Jenis_kelamin'] = en.fit_transform(dataset['Jenis_kelamin'])
dataset['Merokok'] = en.fit_transform(dataset['Merokok'])
dataset['Aktivtas_Begadang'] = en.fit_transform(dataset['Aktivtas_Begadang'])
dataset['Aktivitas_Olahraga'] = en.fit_transform(dataset['Aktivitas_Olahraga'])
dataset['pola makan teratur'] = en.fit_transform(dataset['pola makan teratur'])
dataset['Penyakit_Bawaan'] = en.fit_transform(dataset['Penyakit_Bawaan'])
dataset['Hasil'] = en.fit_transform(dataset['Hasil'])
dataset.head()
```

Segmen Program 4. 3 Label Encoding

4.2.3 *Drop Data*

Dilakukan pemisahan data dengan menggunakan fungsi *drop()* dari pandas untuk menghilangkan kolom "Hasil" dari dataset. Kolom "Hasil" merupakan variabel target yang menunjukkan status kesehatan responden (misalnya Sehat atau Tidak Sehat). Hasil dari fungsi *drop(columns=['Hasil'])* disimpan ke dalam variabel *trs_data*, sehingga tersisa hanya fitur-fitur independen seperti usia, jenis kelamin, kebiasaan merokok, aktivitas begadang, aktivitas olahraga, pola makan, serta penyakit bawaan. Untuk memastikan hasil pemisahan berjalan dengan benar, digunakan fungsi head() yang menampilkan lima baris pertama dari data. Dengan demikian, diperoleh DataFrame baru yang berisi fitur-fitur prediktor tanpa menyertakan variabel target, sehingga dapat digunakan pada tahap preprocessing dan pemodelan selanjutnya.

```
trs_data = dataset.drop(columns=['Hasil'])
trs_data.head()
```

Segmen Program 4. 4 *Drop Data*

4.2.4 *Splitting Data*

Langkah selanjutnya adalah melakukan pemisahan data menjadi dua bagian, yaitu data latih *(training data)* dan data uji *(testing data)*. Proses ini bertujuan untuk melatih model menggunakan sebagian data, lalu menguji performa model tersebut menggunakan data yang belum pernah dilihat sebelumnya. Pembagian ini dilakukan menggunakan fungsi *train_test_split* dari pustaka *sklearn.model_selection*.

Segmen Program 4. 5 *Split Data*

Pada kode yang tersedia, proses pemisahan data dilakukan menggunakan fungsi train_test_split dari model *sklearn.* Fungsi ini bertugas membagi dataset X dan Y menjadi dua bagian, yaitu data pelatihan *(X_train dan Y_train)* serta data pengujian *(X_test dan Y_test)*, sesuai dengan proporsi yang telah ditentukan. Dalam data ini, data pengujian mengambil porsi 20% dari keseluruhan data *(test_size=0.2)*, sehingga sisanya, yaitu 80%, digunakan sebagai data pelatihan. Dengan membagi data seperti ini, model dapat dilatih menggunakan data pelatihan dan diuji performanya pada data pengujian, sehingga kemampuan model dalam mendeteksi penyakit diabetes dapat dievaluasi secara objektif.

Pra-pemrosesan data merupakan tahap krusial dalam penelitian ini guna menjamin keandalan serta ketepatan data yang digunakan. Setelah melewati proses pra-pemrosesan, dataset tersebut siap untuk dikembangkan menjadi model klasifikasi menggunakan algoritma Naïve bayes dan K-NN. Dataset yang telah dibersihkan dan divalidasi ini menjadi fondasi yang kokoh untuk analisis lanjutan serta pemodelan prediktif dalam upaya menghasilkan klasifikasi status Kesehatan yang akurat.

4.3 Implementasi Algoritma

4.3.1 Naive Bayes

Proses pemodelan dilakukan dengan menggunakan pustaka *sklearn.naive_bayes*. *GaussianNB()* digunakan untuk membuat objek model *Naive Bayes* dengan distribusi *Gaussian*. Fungsi *fit(X_train, y_train)* digunakan untuk melatih model menggunakan data latih yang telah dipisahkan sebelumnya. Setelah proses pelatihan selesai, model Naive Bayes siap untuk digunakan dalam proses prediksi terhadap data uji. Model ini akan mempelajari hubungan antara fitur-fitur input (seperti usia, jenis kelamin dan fitur lainnya) dengan label target (Hasil) untuk menentukan kemungkinan klasifikasi status kesehatan.

Pemodelan Algoritma Naive Bayes

```
classifier = GaussianNB() # Instantiate the GaussianNB class
classifier.fit(x_train, y_train) # Use x_train and y_train, not X_train and Y_train
y_pred = classifier.predict(x_test)
```

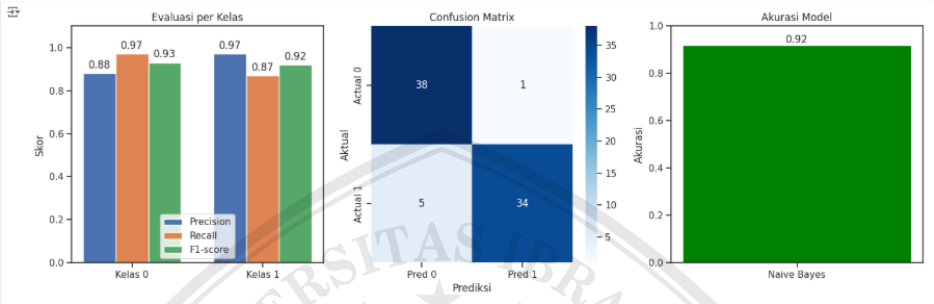
Segmen Program 4. 6 Algoritma Naive Bayes

Setelah proses pelatihan model Naïve Bayes rampung, tahap evaluasi kinerja dilakukan dengan menggunakan matriks konfusi. *Matriks* ini memberikan gambaran menyeluruh mengenai kemampuan prediksi model. Selain itu, *matriks* konfusi juga menjadi dasar penting untuk menghitung berbagai metrik evaluasi, seperti akurasi, presisi, dan recall, yang menilai kualitas model secara keseluruhan.

Evaluasi Model Naive Bayes

```
[9]  print('accuracy of naive bayes kontinu classifier on test set: {:.2f}'.format(classifier.score(x_test, y_test)))
     print(confusion_matrix(y_test, y_pred))
     print(accuracy_score(y_test, y_pred))
     hasilpengujian = classification_report(y_test, y_pred)
     print(hasilpengujian)
```

Segmen Program 4. 7 Evaluasi Model Naive Bayes



Gambar 4. 4 Visualisasi Hasil Evaluasi Model Naive Bayes

Gambar ini menunjukkan hasil evaluasi kemampuan model Naive Bayes dalam mengklasifikasikan data. Pengecekan dilakukan dalam tiga bagian utama, yaitu:

Evaluasi per kelas, *confusion matrix*, dan tingkat akurasi model. Pengecekan Per Kelas Grafik batang di sisi kiri menunjukkan angka *Precision, Recall, dan F1-score* untuk setiap kelas (Kelas 0 dan Kelas 1).

Untuk Kelas 0 (Tidak), nilai Precision adalah 0.88, Recall 0.97, dan F1-score 0.93. Hal ini menunjukkan model cukup baik dalam mengenali data kelas 0, dengan prediksi yang hampir tepat.

Sementara itu, untuk Kelas 1(ya), nilai Precision lebih tinggi yaitu 0.97, Recall 0.87, dan F1-score 0.92. Artinya, model sangat akurat dalam memprediksi kelas 1, meskipun masih ada sedikit data yang salah dikelompokkan. Secara keseluruhan, angka F1-score di kedua kelas

menunjukkan keseimbangan yang baik antara tingkat ketepatan dan kemampuan menangkap data yang benar.

Confuison matrix Bagian tengah menampilkan *Confuison matrix*. Dari 78 data uji: 38 data kelas 0 berhasil diprediksi dengan benar, sedangkan 1 data salah dikelompokkan ke kelas 1. 34 data kelas 1 berhasil diprediksi dengan tepat, namun 5 data salah dikelompokkan ke kelas 0. Ini menunjukkan bahwa model memiliki kesalahan yang terbatas pada kedua kelas. Akurasi Model Grafik batang di sebelah kanan menunjukkan bahwa model Naive Bayes memiliki akurasi keseluruhan sebesar 0.92 atau 92%. Hal ini berarti dari seluruh data uji yang digunakan, 92% di antaranya berhasil diprediksi secara benar.

4.3.2 K-Nearest Neighbor

Pada bagian ini digunakan untuk menguji performa algoritma K-Nearest Neighbor (K-NN) dengan berbagai nilai K (jumlah tetangga terdekat) untuk menentukan parameter terbaik.

Pertama, variabel *k_rng = range(2,10)* menentukan bahwa nilai K yang akan diuji berada pada rentang 2 hingga 9. Selanjutnya, variabel sse diinisialisasi sebagai list kosong untuk menyimpan hasil perhitungan *sum of squared errors (SSE)* pada setiap percobaan.

Kemudian, melalui perulangan *for k in k_rng:*, dilakukan langkah-langkah berikut untuk setiap nilai K: Membuat objek *K-NeighborsClassifier* dengan parameter *n_neighbors=k*, yang berarti jumlah tetangga terdekat diatur sesuai nilai K yang sedang diuji. Melatih model

menggunakan data latih *(x_train dan y_train)* dengan fungsi *.fit()*. Menggunakan model untuk memprediksi data uji *(x_test)* melalui fungsi *.predict()*, dan menyimpan hasilnya di *y_pred*. Mencetak tingkat akurasi model untuk nilai K tersebut menggunakan *accuracy_score(y_test, y_pred)*. Menghitung *sum of squared errors (SSE)* antara hasil prediksi dan data aktual, lalu menyimpannya ke dalam list *sse*.

Pemodelan K-NN

```
k_rng = range(2,10)
sse = []
for k in k_rng:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train, y_train)
    y_pred = knn.predict(x_test)
    k=+k
    print ('Accuracy of K-NN classifier on test set with K = ', k, ' : ', accuracy_score(y_test, y_pred))
    sse.append(np.sum((y_pred-y_test)**2))
```

Segmen Program 4. 8 Pemodelan Algoritma K-NN

Pada tahap selanjutnya adalah dilakukan proses pemodelan menggunakan algoritma K-Nearest Neighbors (K-NN) dengan nilai parameter *k = 3*. Pertama, model diinisialisasi menggunakan perintah *knn = KNeighborsClassifier(n_neighbors=3)*, yang berarti bahwa penentuan kelas suatu data uji akan didasarkan pada tiga tetangga terdekat di dalam ruang fitur.

Selanjutnya, dilakukan proses pelatihan model dengan menggunakan data latih melalui perintah *knn.fit(x_train, y_train)*. Pada tahap ini, algoritma K-NN menyimpan seluruh data latih untuk digunakan sebagai acuan dalam menghitung jarak ketika proses klasifikasi data uji dilakukan.

Setelah model terlatih, dilakukan prediksi terhadap data uji dengan perintah *y_pred = knn.predict(x_test)*. Proses prediksi ini melibatkan

pencarian tiga tetangga terdekat dari setiap data uji, kemudian menentukan kelas berdasarkan mayoritas kelas dari tetangga tersebut. Hasil prediksi ini nantinya akan digunakan pada tahap evaluasi untuk mengukur kinerja model K-NN.

Penentuan Nilai K

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)

y_pred = knn.predict(x_test)
```

Segmen Program 4. 9 Penentuan Nilai K

Pada tahap ini dilakukan evaluasi kinerja algoritma *K-Nearest Neighbors* (K-NN) terhadap data uji. Nilai akurasi model dihitung menggunakan fungsi *knn.score()* dan ditampilkan dalam format persentase untuk menunjukkan tingkat ketepatan prediksi model.

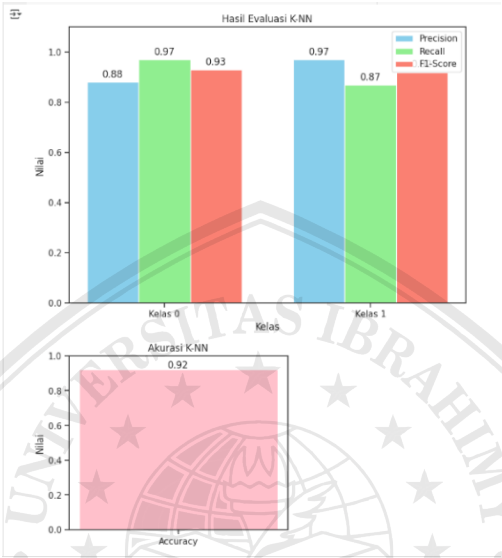
Evaluasi Model K-NN

```
print ('accuracy of K-NN classifier on test set: {:.2f}'.format(knn.score(x_test, y_test)))
print(confusion_matrix(y_test, y_pred))
print(accuracy_score(y_test, y_pred))
hasilpengujian2 = classification_report(y_test, y_pred)
print(hasilpengujian2)
```

Segmen Program 4. 10 Evaluasi Model K-NN

Selanjutnya, *confusion matrix* digunakan untuk menggambarkan jumlah prediksi yang benar dan salah pada masing-masing kelas. Matriks ini membantu mengidentifikasi seberapa akurat model dalam memprediksi setiap kategori. Kemudian, nilai akurasi kembali dihitung dengan fungsi *accuracy_score()* untuk memastikan konsistensi hasil perhitungan. Terakhir, dihasilkan *classification report* yang berisi metrik *precision,*

recall, dan f1-score untuk tiap kelas, serta nilai rata-rata makro dan berbobot. Laporan ini memberikan gambaran yang lebih rinci mengenai performa model, tidak hanya dari sisi akurasi keseluruhan, tetapi juga kemampuan model dalam mengenali setiap kelas secara seimbang.



Gambar 4. 5 Visualisasi Hasil Evaluasi K-NN

Grafik tersebut pada gambar 4.15 menampilkan hasil K-NN berdasarkan tiga ukuran utama, yaitu *Precision, Recall, dan F1-Score* untuk masing-masing kelas.

Kelas 0 (tidak) dengan nilai *precision*: 0.88 dari semua hasil prediksi yang dikategorikan sebagai Kelas 0, sekitar 88% di antaranya benar. *Recall*: 0.97 dari seluruh data sebenarnya yang termasuk dalam Kelas 0, sebanyak 97% berhasil terdeteksi dengan tepat. *F1-Score*: 0.93 nilai ini menggabungkan antara *precision* dan *recall* secara seimbang, menunjukkan performa yang sangat baik.

Kelas 1 (ya) dengan nilai *precision*: 0.97 model ini cukup akurat dalam memprediksi Kelas 1. *Recall*: 0.87 meskipun tingkat keakuratan tinggi, tetap ada sebagian data Kelas 1 yang belum terdeteksi dengan baik. *F1-Score*: 0.92 menunjukkan bahwa model memiliki performa yang stabil untuk kelas ini. Secara keseluruhan, K-NN menunjukkan performa yang seimbang untuk kedua kelas. Meskipun demikian, kecenderungan model ini mirip dengan model sebelumnya (Naive Bayes), yaitu *recall* pada Kelas 1 cenderung lebih rendah dibandingkan dengan tingkat *Precision-nya.*

4.4 Implementasi *Framework Streamlit*

Fase implementasi dalam penelitian ini berfokus pada integrasi model machine learning yang telah dikembangkan, yaitu algoritma Naive Bayes dan K-Nearest Neighbor (K-NN), ke dalam aplikasi berbasis web menggunakan framework Streamlit. Model dilatih dan dijalankan secara real-time di dalam Google Colab serta dihubungkan dengan antarmuka Streamlit.

Melalui pendekatan ini, setiap kali pengguna memberikan input melalui antarmuka aplikasi, sistem akan menjalankan *pipeline preprocessing*, melatih atau memanggil model yang sudah disiapkan, kemudian menghasilkan prediksi secara langsung. Streamlit menyediakan komponen interaktif, seperti input form, tombol, dan tampilan hasil, sehingga pengguna dapat berinteraksi dengan sistem dengan mudah.

Implementasi framework Streamlit ini bertujuan untuk mempermudah pengguna dalam memanfaatkan hasil penelitian, karena

model tidak hanya berhenti pada tahap evaluasi akurasi, tetapi juga dapat diakses secara praktis melalui antarmuka web yang sederhana. Dengan demikian, hasil eksperimen Naive Bayes dan K-NN dapat diaplikasikan secara langsung dalam bentuk prediksi yang lebih mudah dipahami oleh pengguna akhir.

4.4.1 Tampilan Antarmuka Web

Antarmuka aplikasi web yang dikembangkan dalam penelitian ini dirancang dengan tujuan untuk memberikan pengalaman pengguna yang intuitif dan ramah.

Gambar 4. 6 Tampilan Antarmuka Web

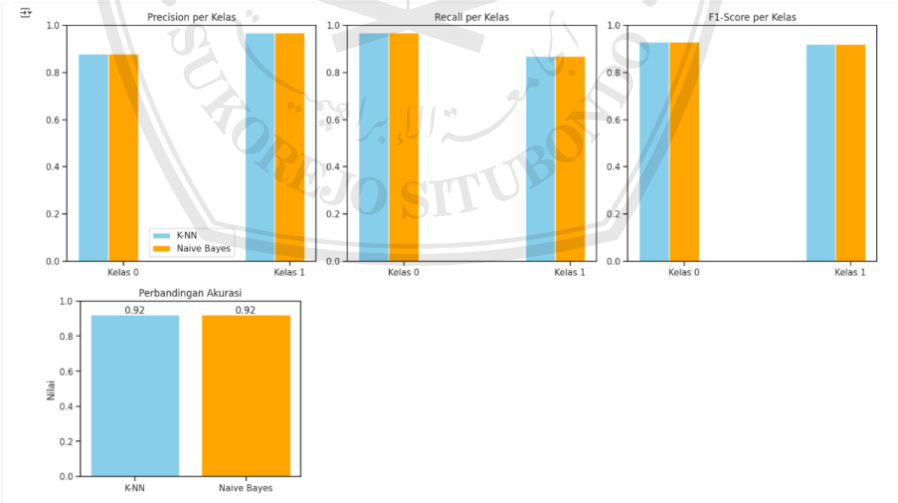
Pada gambar 4.16 menunjukkan tampilan antarmuka web dimana pada bagian atas tampilan terdapat judul aplikasi, "Klasifikasi Status Kesehatan (Input Manual Banyak Data)". Pada bagian bawah judul terdapat teks penjelasan berwarna biru yang memberikan instruksi: "Isi data baru menggunakan pilihan dropdown di bawah ini, lalu tambahkan baris sesuai kebutuhan."

Di bawahnya terdapat sebuah tabel input dengan kolom: Usia, Jenis kelamin, Merokok, Bekerja, Aktivitas Begadang, Aktivitas Olahraga dan penyakit bawaan. Pada tabel terlihat satu baris data contoh: Muda, Pria, Aktif, yes, iya, Jarang.

Kemudian di bagian bawah tabel terdapat sebuah tombol "Klasifikasikan", yang berfungsi untuk memproses data yang telah diinput pada tabel agar bisa dilakukan klasifikasi status kesehatan sesuai dengan model machine learning yang digunakan.

4.4.2 Analisis Hasil Perbandingan

Dari evaluasi hasil analisis kedua algoritma naïve bayes dan K-NN, diperoleh analisis hasil perbandingan dengan visualisasi pada gambar 4.16 dibawah ini:



Gambar 4. 7 Visualisasi Analisis Hasil Perbandingan

Grafik tersebut menampilkan tingkat presisi untuk setiap kelas, yaitu Kelas 0 (tidak) dan Kelas 1(ya), pada kedua algoritma yang digunakan.

Untuk Kelas 0, presisi dari K-NN dan Naive Bayes hampir sama, masing-masing sekitar 0.88. Sementara itu, pada Kelas 1, kedua algoritma juga menunjukkan presisi yang hampir sama, sekitar 0.97. Dengan demikian, baik K-NN maupun Naive Bayes memiliki kemampuan yang cukup baik dalam memprediksi data positif untuk kedua kelas.

Untuk *recall* menunjukkan seberapa baik model mampu menemukan semua data yang benar pada setiap kelas. Di kelas 0, kedua algoritma memiliki tingkat recall yang tinggi, sekitar 0.97. Di kelas 1, *recall* kedua algoritma pun hampir sama, sekitar 0.87. Ini menunjukkan bahwa kedua algoritma mampu mengenali data dari setiap kelas dengan tingkat kemampuan yang hampir sama.

Untuk grafik *F1-Score* yang menggabungkan *precision* dan *recall* menunjukkan hasil yang sama antara kedua algoritma: Untuk kelas 0 sekitar 0.93. Untuk kelas 1 sekitar 0.92. Ini menunjukkan bahwa kedua algoritma memiliki performa yang seimbang pada setiap kelas, tanpa perbedaan yang terlalu besar.

Untuk perbandingan akurasi kedua algoritma diperoleh hasil dimana, algoritma K-NN memiliki akurasi sebesar 0.92 (92%). Algoritma Naive Bayes juga memiliki akurasi sebesar 0.92 (92%). Jadi, secara keseluruhan, tidak ada perbedaan signifikan dalam akurasi antara kedua algoritma pada dataset ini.

Meskipun demikian, setiap algoritma memiliki kelebihan dan kekurangannya masing-masing. Algoritma *K-Nearest Neighbor* (K-NN)

memiliki kelebihan karena tidak memerlukan asumsi distribusi data serta mampu menangkap pola non-linear dengan baik. Namun demikian, K-NN cenderung lebih lambat ketika digunakan pada dataset yang berukuran besar dan sensitif terhadap skala data sehingga memerlukan normalisasi terlebih dahulu.

Sementara itu, algoritma Naive Bayes memiliki keunggulan dalam hal kecepatan baik pada tahap pelatihan maupun prediksi. Algoritma ini juga andal ketika diterapkan pada dataset dengan ukuran kecil hingga besar, bahkan tetap memberikan hasil yang baik meskipun asumsi independensi antar fitur tidak sepenuhnya terpenuhi. Akan tetapi, Naive Bayes sangat bergantung pada asumsi distribusi data dan sensitif terhadap adanya korelasi antar fitur yang tinggi.

Dengan mempertimbangkan karakteristik tersebut, pemilihan algoritma sangat bergantung pada kebutuhan penelitian maupun implementasi. *Naive Bayes* lebih sesuai digunakan apabila kecepatan dan efisiensi menjadi prioritas utama, sedangkan K-NN lebih tepat digunakan ketika fleksibilitas dalam menangkap pola data *non-linear* lebih dibutuhkan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai perbandingan algoritma Naive Bayes dan K-Nearest Neighbor (K-NN) dalam mengklasifikasikan status kesehatan, diperoleh kesimpulan bahwa kedua algoritma menunjukkan performa yang sama baik. Hasil evaluasi menunjukkan bahwa akurasi Naive Bayes dan K-NN sama-sama mencapai 92%. Pada evaluasi per kelas, baik precision, recall, maupun F1-score dari kedua algoritma relatif identik, dengan nilai precision tertinggi pada Kelas 1 (0.97) dan recall tertinggi pada Kelas 0 (0.97). Hal ini menunjukkan bahwa kedua algoritma mampu melakukan klasifikasi dengan tingkat ketepatan yang tinggi dan seimbang.

Dengan demikian, dapat disimpulkan bahwa *Naive Bayes* dan K-NN sama-sama efektif digunakan untuk klasifikasi status kesehatan, tanpa perbedaan performa yang signifikan. Pemilihan algoritma selanjutnya dapat disesuaikan dengan kebutuhan implementasi, di mana *Naive Bayes* lebih unggul dari sisi kecepatan dan kesederhanaan, sementara K-NN lebih fleksibel dalam menangani data dengan pola *non-linear.*

Selain itu, implementasi menggunakan Streamlit berhasil memvisualisasikan hasil perbandingan kedua algoritma dengan lebih interaktif dan mudah dipahami. Streamlit memberikan antarmuka yang sederhana untuk menampilkan input untuk beberapa data baru, dan hasil

klasifikasi dapat di unduh dalam format excel. Dengan adanya visualisasi ini, pengguna dapat memahami performa algoritma secara lebih intuitif tanpa harus langsung berinteraksi dengan kode program.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat menjadi masukan untuk penelitian selanjutnya. Pertama, penggunaan dataset dengan jumlah yang lebih besar dan lebih bervariasi sangat dianjurkan agar model dapat dilatih dengan data yang lebih representatif sehingga hasil yang diperoleh semakin optimal. Kedua, penelitian berikutnya dapat memperluas perbandingan dengan algoritma lain, seperti *Random Forest, Decision Tree*, maupun metode berbasis *Deep Learning*, guna memperoleh gambaran performa yang lebih komprehensif. Selain itu, optimasi parameter pada algoritma juga penting dilakukan, misalnya penentuan nilai k terbaik pada K-NN atau penggunaan varian distribusi berbeda pada Naive Bayes.

Metode evaluasi juga dapat ditingkatkan dengan menerapkan validasi silang *(cross-validation)* agar hasil evaluasi lebih stabil dan tidak hanya bergantung pada satu kali pembagian data. Untuk pengembangan lebih lanjut, aplikasi Streamlit dapat ditingkatkan dengan penambahan fitur autentikasi pengguna, sehingga hanya pengguna tertentu yang dapat mengakses sistem. Selain itu, integrasi dengan database online dapat dilakukan agar data yang dimasukkan tidak hanya disimpan dalam file unduhan, tetapi juga tersimpan secara terpusat dan dapat dikelola kembali.

Aplikasi juga bisa dilengkapi dengan dashboard monitoring kesehatan yang menampilkan tren atau riwayat prediksi pengguna dalam bentuk grafik interaktif. Dengan pengembangan ini, aplikasi tidak hanya berfungsi sebagai alat prediksi, tetapi juga dapat menjadi sistem pemantauan kesehatan yang lebih komprehensif.

DAFTAR PUSTAKA

[1] Fadhillah, R. A'la, And Z. Fatah, "Perbandingan Algoritma Decision Tree Dan Deep Learning Dalam," *Multidiciplinary Scientifict Journal*, Vol. 2, 2024.

[2] S. Emma *Et Al.*, "PADA MASYARAKAT DI DESA X," Vol. 6, No. 1, 2022.

[3] L. E. T. Kusrini, *Algoritma Data Mining*, Vol. 2009. Yogyakarta: PENERBIT ANDI.

[4] P. W. Rahayu *Et Al.*, *Buku Ajar Data Mining*. PT. Sonpedia Publishing Indonesia, 2024.

[5] J. Homepage, S. Kenia, P. Loka, And A. Marsal, "MALCOM: Indonesian Journal Of Machine Learning And Computer Science Comparison Algorithm Of K-Nearest Neighbor And Naïve Bayes Classifier For Classifying Nutritional Status In Toddlers Perbandingan Algoritma K-Nearest Neighbor Dan Naïve Bayes Classifier Untuk Klasifikasi Status Gizi Pada Balita," Vol. 3, Pp. 8–14, 2023.

[6] M. Jannah, M. Arief, H. M. Kom, M. Al Fajar, And M. A. Hasan, "Perbandingan Metode Naïve Bayes Dan K-Nearest Neighbor Dalam Mengklasifikasi Status Pertumbuhan Anak Stunting (Studi Kasus : Posyandu Cemara)".

[7] G. Maulani *Et Al.*, *Machine Learning*. Jawa Barat: CV. Mega Press Nusantara, 2025. Accessed: Jun. 09, 2025. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/Machine_Learning/Rblpeqaaqbaj

[8] S. T. , M. Kom. , G. Urva, *Penerapan Data Mining Di Berbagai Bidang*. PT. Sonpedia Publishing Indonesia, 2023.

[9] S. Kom. , M. Kom. , M. T. Thamrin And D. S. Utsalina, *Dasar- Dasar Data Mining*. Sumatera Barat: Yayasan Tri Edukasi Ilmiah, 2025.

[10] D. Nasien *Et Al.*, "Perbandingan Implementasi Machine Learning Menggunakan Metode KNN, Naive Bayes, Dan Logistik Regression Untuk Mengklasifikasi Penyakit Diabetes," 2024.

[11] F. Sholekhah, A. D. Putri, R. Rahmaddeni, And L. Efrizoni, "Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbors Untuk Klasifikasi Metabolik Sindrom," *MALCOM: Indonesian Journal Of Machine Learning And Computer Science*, Vol. 4, No. 2, Pp. 507–514, Feb. 2024, Doi: 10.57152/Malcom.V4i2.1249.

[12] M. Arhami And M. Nasir, *Data Mining Algoritma Dan Implementasi*. 2020.

[13] M. Arhami And M. Nasir, *Data_Mining_Algoritma_Dan_Implementasi*. 2020. Accessed: Dec. 13, 2024. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/Data_Mining_Algoritma_Dan_Implementasi/Atcceaaaqbaj?Hl=Id&Gbpv=0

[14] R. S. Wahono, *"Data Mining."* 2020. Accessed: Dec. 13, 2024. [Online]. Available: R. Satria Wahono, "Data Mining," Data Mining, May 2020. Https://Romisatriawahono.Net/Dm/

[15] E. A. Novia, I. W. Rahayu, And C. Prianto, *Sistem Perbandingan Algoritma K-Means Dan Naive Bayes Untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan*. Bandung: Kreatif Industri Nusantara, 2020. Accessed: Dec. 13, 2024. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/SISTEM_PERBANDINGAN_ALGORITMA_K_MEANS_DA/MND9DwAAQBAJ?Hl=Id&Gbpv=1&Dq=Naive+Bayes&Pg=PA37&Printsec=Frontcover

[16] W. Ode Nurhayah Kadir And B. Pramono, "Terakreditasi 'Peringkat 4 (Sinta 4)' Oleh Kemenristekdikti PENERAPAN DATA MINING DENGAN METODE K-NEAREST NEIGHBOR (KNN) UNTUK MENGELOMPOKAN MINAT KONSUMEN ASURANSI (PT. JASARAHARJA PUTERA)," Vol. 5, No. 1, Pp. 97–104, Doi: 10.5281/Zenodo.3116132.

[17] ST. M. K. Yahya, *Data_Mining*, Vol. 224. CV. Jejak (Jejak Publisher), 2022. Accessed: Jul. 20, 2025. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/Data_Mining/0j2meaaaqbaj?Hl=Id&Gbpv=0&Kptab=Overview

[18] S. S. M. K. Muhammad Arhami And S. T. M. T. Muhammad Nasir, *Data Mining - Algoritma Dan Implementasi*. Andi Offset, 2020. [Online]. Available: Https://Books.Google.Co.Id/Books?Id=Atcceaaaqbaj

[19] T. Penulis *Et Al.*, *DATA SCIENCE*. 2022. [Online]. Available: Www.Penerbitwidina.Com

[20] Nyimas Sri Wahyuni, "Kesehatan Dan Makna Sehat," KEMENKES - Direktorat Jendral Kesehatan Lanjutan. Accessed: Jan. 21, 2025. [Online]. Available: Https://Yankes.Kemkes.Go.Id/View_Artikel/119/Kesehatan-Dan-Makna-Sehat

[21] D. Renaldi And Edy, *Menjelejahi Bahasa Phyton Dengan Google Colab*. Bogor: Guepedia, 2024. Accessed: Feb. 08, 2025. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/Menjelajahi_Bahasa_Python_Dengan_Google/Gseyeqaaqbaj?Hl=Id&Gbpv=1&Dq=Google+Colab+Adalah&Pg=PA178&Printsec=Frontcover

[22] F. M. K. Sembiring, *Buku_Ajar_Dasar_Pemrograman_Python*. Jawa Barat : Nusa Putra Press. Accessed: May 09, 2025. [Online]. Available: Https://Www.Google.Co.Id/Books/Edition/Buku_Ajar_Dasar_Pemrograman_Python/Za08eaaaqbaj?Hl=Id&Gbpv=0

[23] , Rahmaddeni, Wulandari, A. Denok, S. Adrianto, And F. Pratiwi, *Teori Dan Implementasi Machine Learning Menggunakan Python* , First. PT. Serasi Media Teknologi , 2025.

[24] R. A. Putra, *Langkah Mudah Belajar Membuat Aplikasi Data Interaktif Dengan Streamlit Python Untuk Pemula* . Anak Hebat Indonesia , 2024.

[25] M. Kes. , S.-K. M. R. I. Ikhsan, “Penyakit Yang Timbul Akibat Gaya Hidup Tidak Sehat.”

[26] Kemenkes RI, “Kategori Usia .” Accessed: Aug. 18, 2025. [Online]. Available: Https://Ayosehat.Kemkes.Go.Id/Kategori-Usia

[27] W. Sudirohusodo, “Perokok Aktif Dan Perokok Pasif,” Kemenkes Rs. Wahidin Sudirohusodo. Accessed: Jul. 24, 2025. [Online]. Available: Https://Rsupwahidin.Com/Berita-122-Perokok-Aktif-Dan-Perokok-Pasif.Html

[28] Y. A. Tosepu, “Hakikat Bekerja .” Accessed: Aug. 18, 2025. [Online]. Available: Https://Yusrintosepu.Wixsite.Com/Yoes/Post/Bekerja-Tujuan-Makna-Dan-Hakikat#:~:Text=Tujuan%20utama%20manusia%20bekerja%20adalah,Dan%20mengaktualisasikan%20dirinya%20dalam%20bekerja.

[29] Fakultas Keperawatan, “Mengenal Dampak Yang Timbul Dari Kebiasaan Begadang ,” Universitas Airlangga . Accessed: Aug. 18, 2025. [Online]. Available: Https://Ners.Unair.Ac.Id/Site/Index.Php/News-Fkp-Unair/30-Lihat/1083-Mengenal-Dampak-Yang-Timbul-Dari-Kebiasaan-Begadang#:~:Text=Pengertian%20begadang%20menurut%20KBBI%20adalah,Padahal%20dua%20hal%20ini%20berbeda.

[30] Kemenkes RI, “Mengenal Jenis Aktivitas Fisik .” Accessed: Aug. 18, 2025. [Online]. Available: Https://Ayosehat.Kemkes.Go.Id/Mengenal-Jenis-Aktivitas-Fisik#:~:Text=Secara%20umum%20aktivitas%20fisik%20dibagi,Dengan%20latihan%2C%20dan%20juga%20olahraga.

[31] -, “Pola Makan Sehat ,” Kemenkes Labkesmas Makassar I . Accessed: Aug. 18, 2025. [Online]. Available: Https://Bblabkesmasmakassar.Go.Id/Pola-Makan-Sehat-Dengan-Tumpeng-Gizi-Seimbang/

[32] Dr. K. Adrian, "Berbagai Penyakit Keturunan Yang Perlu Anda Waspadai ," Alodokter . Accessed: Aug. 18, 2025. [Online]. Available: Https://Www.Alodokter.Com/Berbagai-Penyakit-Keturunan-Yang-Perlu-Anda-Waspadai#:~:Text=Orang%20yang%20menderita%20gangguan%20mental,Atau%20tekanan%20psikologis%20yang%20berat.&Text=Anda%20juga%20dapat%20melakukan%20pemeriksaan,Yang%20bisa%20diwariskan%20kepada%20anak.

[33] W. Andriyani, F. Natsir, And H. Lubis, *Perangkat Lunak Data Mining*. Bandung: WIDINA MEDIA UTAMA, 2024. Accessed: Dec. 13, 2024. [Online]. Available: Https://Books.Google.Co.Id/Books?Id=Bdoueqaaqbaj&Newbks=0&Printsec=Frontcover&Pg=PA51&Dq=Random+Forest+Adalah&Hl=Id&Source=Newbks_Fb&Redir_Esc=Y#V=Onepage&Q=Random%20forest%20adalah&F=False

[34] W. Ode Nurhayah Kadir And B. Pramono, "Terakreditasi 'Peringkat 4 (Sinta 4)' Oleh Kemenristekdikti Penerapan Data Mining Dengan Metode K-Nearest Neighbor (Knn) Untuk Mengelompokan Minat Konsumen Asuransi (Pt. Jasaraharja Putera)," Vol. 5, No. 1, Pp. 97–104, Doi: 10.5281/Zenodo.3116132.

CURRICULUM VITAE

Contac Me

Email
nsfh2502@gmail.com

IDENTITTAS

NAMA : Nazhifatul Muthohharoh

PRODI : Teknologi Informasi

FAKULTAS : Sains dan Teknologi

ALAMAT : Curah Jeru, Panji, Situbondo

ORANG TUA

AYAH : Hardiyono

IBU : Kalsum

RIWAYAT PENDIDIKAN

TK ALHIDAYAH IV : 2008

SDN 4 CURAH JERU : 2009

SMP IBRAHIMY 3 : 2015

SMA IBRAHIMY : 2018

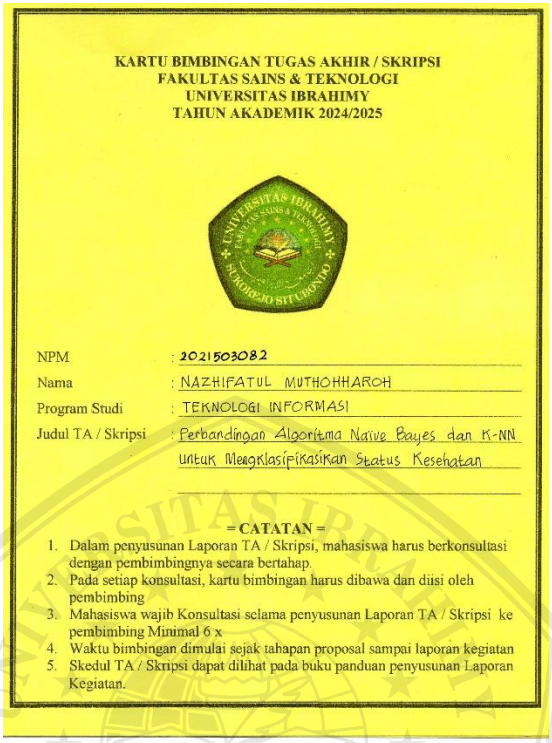
LAMPIRAN-LAMPIRAN

A. Lampiran Dataset

	B	C	D	E	F	G	H	I
1	Jenis_kelamin	Merokok	Bekerja	Aktivtas_Begadang	Aktivitas_Olahraga	pola makan teratur	Penyakit_Bawaan	Hasil
2	Pria	Aktif	No	iya	Jarang	Teratur	Tidak Ada	Tidak
3	Wanita	Pasif	yes	tidak	Sering	Kurang	Ada	Ya
4	Wanita	Pasif	yes	tidak	Sering	Kurang	Ada	Ya
5	Pria	Aktif	No	iya	Sering	Kurang	Tidak Ada	Ya
6	Pria	Aktif	No	iya	Jarang	Teratur	Tidak Ada	Tidak
7	Wanita	Aktif	yes	iya	Jarang	Teratur	Ada	Tidak
8	Wanita	Pasif	yes	iya	Sering	Kurang	Ada	Ya
9	Wanita	Aktif	yes	iya	Jarang	Teratur	Ada	Tidak
10	Pria	Pasif	No	iya	Sering	Teratur	Tidak Ada	Tidak
11	Pria	Aktif	yes	tidak	Jarang	Teratur	Ada	Tidak
12	Wanita	Pasif	yes	tidak	Sering	Kurang	Ada	Ya
13	Wanita	Pasif	yes	tidak	Sering	Teratur	Ada	Ya
14	Wanita	Aktif	yes	iya	Jarang	Teratur	Ada	Tidak
15	Wanita	Pasif	yes	iya	Jarang	Teratur	Ada	Ya
16	Wanita	Pasif	yes	tidak	Sering	Teratur	Ada	Ya
17	Pria	Aktif	yes	tidak	Jarang	Teratur	Ada	Tidak
18	Wanita	Pasif	yes	tidak	Sering	Teratur	Ada	Ya
19	Wanita	Aktif	yes	iya	Jarang	Teratur	Ada	Tidak
20	Wanita	Pasif	No	tidak	Sering	Kurang	Tidak Ada	Ya

Lampiran 1 Dataset Lifestyle Habits

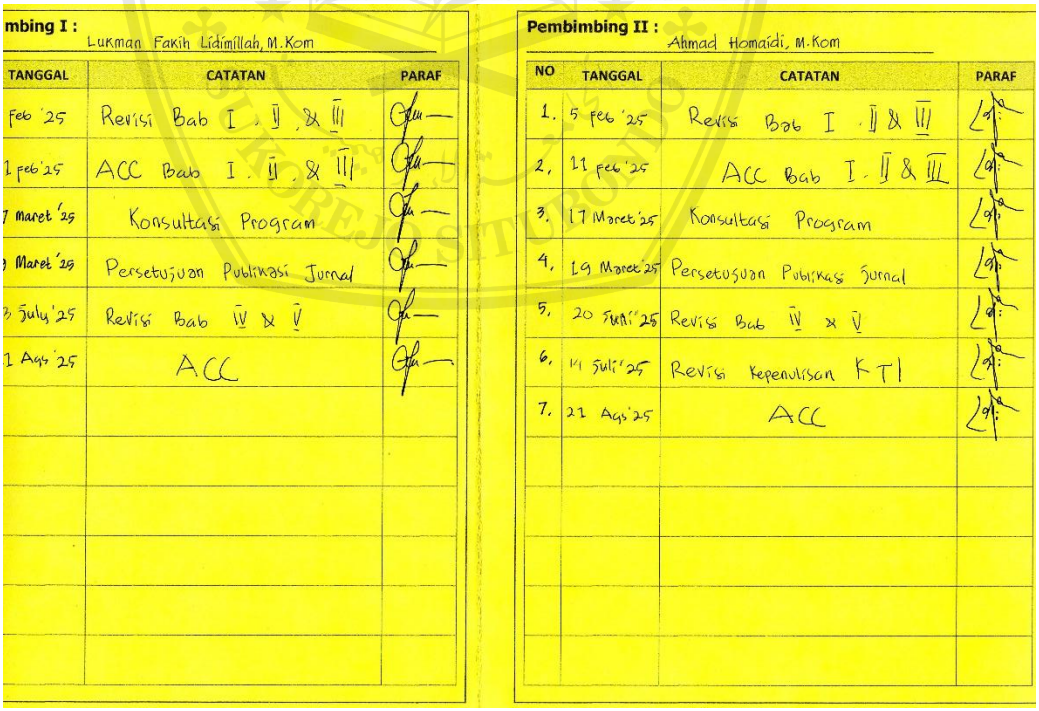
B. Lampiran Kartu Bimbingan



KARTU BIMBINGAN TUGAS AKHIR / SKRIPSI
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS IBRAHIMY
TAHUN AKADEMIK 2024/2025

NPM : 2021503082
Nama : NAZHIFATUL MUTHOHHAROH
Program Studi : TEKNOLOGI INFORMASI
Judul TA / Skripsi : Perbandingan Algoritma Naïve Bayes dan K-NN untuk Meagklasifikasikan Status Kesehatan

= CATATAN =
1. Dalam penyusunan Laporan TA / Skripsi, mahasiswa harus berkonsultasi dengan pembimbingnya secara bertahap.
2. Pada setiap konsultasi, kartu bimbingan harus dibawa dan diisi oleh pembimbing
3. Mahasiswa wajib Konsultasi selama penyusunan Laporan TA / Skripsi ke pembimbing Minimal 6 x
4. Waktu bimbingan dimulai sejak tahapan proposal sampai laporan kegiatan
5. Skedul TA / Skripsi dapat dilihat pada buku panduan penyusunan Laporan Kegiatan.



mbing I : Lukman Fakih Lidimillah, M.Kom

TANGGAL	CATATAN	PARAF
Feb '25	Revisi Bab I, II, & III	
1 Feb '25	ACC Bab I, II, & III	
7 Maret '25	Konsultasi Program	
Maret '25	Persetujuan Publikasi Jurnal	
3 July '25	Revisi Bab IV & V	
1 Ags '25	ACC	

Pembimbing II : Ahmad Homaidi, M.Kom

NO	TANGGAL	CATATAN	PARAF
1.	5 Feb '25	Revisi Bab I, II & III	
2.	11 Feb '25	ACC Bab I, II & III	
3.	17 Maret '25	Konsultasi Program	
4.	19 Maret '25	Persetujuan Publikasi Jurnal	
5.	20 Juli '25	Revisi Bab IV & V	
6.	11 Juli '25	Revisi Kepenulisan KTI	
7.	21 Ags '25	ACC	

Lampiran 2 Kartu Bimbingan

C. Lampiran LoA

Prosiding Seminar Nasional
Karsa Nusantara
Kolaborasi Rekayasa dan Sains Nasional
Untuk Teknologi, Riset, dan Kecerdasan Buatan
ISSN 3090-1154 (Online)

Nomor : 011/LoA/KarsaNusantara/VII/2025

Perihal : Letter of Acceptance (LoA)

Kepada
Yth.
Pemakalah Prosiding Seminar Nasional Karsa Nusantara 2025
Di Tempat

Pemimpin Redaksi Prosiding Seminar Nasional Karsa Nusantara telah melakukan evaluasi terhadap Naskah:

Judul : Perbandingan algoritma Naïve Bayes dan K-Nearest Neighbor untuk mengklasifikasikan status kesehatan

Penulis : Nazhifatul Muthohharoh*1, Lukman Fakih Lidimilah,M.Kom2, Ahmad Homaidi, M.Kom3
Afiliasi : Universitas Ibrahimy

Korespondensi: Nazhifatul Muthohharoh

Berdasarkan penilaian dari Reviewer terhadap Naskah yang telah disubmit, maka diputuskan bahwa Naskah tersebut DITERIMA, dan dapat dimuat pada terbitan prosiding Seminar Nasional Karsa Nusantara pada Volume 2 Tahun 2025.

Terimakasih untuk partisipasinya dalam mempercayakan naskahnya kepada prosiding kami.

Surabaya, 29 Juni 2025
Pemimpin Redaksi

Dr. Ir. Anang Kukuh Adisusilo, ST., MT

Lampiran 3 LoA

D. Lampiran Sertifikat

Lampiran 4 Seritifikat

E. Lampiran Segmen Streamlit

```
 import streamlit as st
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score

st.title("📊 Perbandingan Algoritma Naive Bayes dan KNN
untuk Klasifikasi Status Kesehatan")

# =======================
# Setup Session State
# =======================
if "step" not in st.session_state:
    st.session_state.step = 0
if "df" not in st.session_state:
    st.session_state.df = None
if "df_encoded" not in st.session_state:
    st.session_state.df_encoded = None
if "target_col" not in st.session_state:
    st.session_state.target_col = None
if "X" not in st.session_state:
    st.session_state.X = None
if "y" not in st.session_state:
    st.session_state.y = None

# =======================
# STEP 0: Upload File
# =======================
if st.session_state.step == 0:
    uploaded_file = st.file_uploader("📁 Upload file
dataset (CSV/XLSX)", type=["csv", "xlsx"])
    if uploaded_file:
        try:
            if uploaded_file.name.endswith("csv"):
                df = pd.read_csv(uploaded_file)
            else:
                df = pd.read_excel(uploaded_file)

            st.session_state.df = df
            st.write("✅ Dataset berhasil dimuat!")
            st.dataframe(df.head())
```

```
            if st.button("➡️ Lanjut: Proses Missing
Value"):
                st.session_state.step = 1
                st.rerun()

        except Exception as e:
            st.error(f"Terjadi kesalahan saat membaca
file: {e}")

# ======================
# STEP 1: Missing Value
# ======================
elif st.session_state.step == 1:
    st.subheader("🩹 Proses Missing Value")
    df = st.session_state.df

    st.write("Jumlah missing value di setiap kolom:")
    st.write(df.isnull().sum())

    if st.button("⚙️ Isi Missing Value (dengan
modus/median)"):
        for col in df.columns:
            if df[col].dtype == 'object':
                df[col].fillna(df[col].mode()[0],
inplace=True)
            else:
                df[col].fillna(df[col].median(),
inplace=True)
        st.session_state.df = df
        st.success("✅ Missing value sudah diisi.")
        st.dataframe(df.head())

    if st.button("➡️ Lanjut: Label Encoding"):
        st.session_state.step = 2
        st.rerun()

# ======================
# STEP 2: Label Encoding
# ======================
elif st.session_state.step == 2:
    st.subheader("🔤 Proses Label Encoding")
    df = st.session_state.df.copy()

    if st.button("⚙️ Lakukan Label Encoding"):
        df_encoded = df.copy()
        label_encoders = {}
        for col in
df_encoded.select_dtypes(include='object').columns:
            le = LabelEncoder()
```

```
            df_encoded[col] =
le.fit_transform(df_encoded[col].astype(str))
            label_encoders[col] = le
            st.write(f"Kolom '{col}' telah di-encode.")

        st.session_state.df_encoded = df_encoded
        st.success("✅ Label Encoding selesai.")
        st.dataframe(df_encoded.head())

    if st.button("➡️ Lanjut: Pilih Label (Target)"):
        st.session_state.step = 3
        st.rerun()

# ======================
# STEP 3: Pilih Label
# ======================
elif st.session_state.step == 3:
    st.subheader("🎯 Pilih Kolom Label (Target)")
    df = st.session_state.df_encoded

    target = st.selectbox("Pilih kolom target
(label):", options=df.columns)
    if st.button("✅ Konfirmasi Pilihan Label"):
        st.session_state.target_col = target
        feature_cols = [col for col in df.columns if
col != target]

        st.session_state.X = df[feature_cols]
        st.session_state.y = df[target]

        st.success(f"Label: **{target}** | Fitur:
{feature_cols}")

    if st.button("➡️ Lanjut: Split Data & Pilih K"):
        st.session_state.step = 4
        st.rerun()

# ======================
# STEP 4: Split Data + Pilih K
# ======================
elif st.session_state.step == 4:
    st.subheader("📂 Split Data & Pilih Nilai K untuk
KNN")

    test_size = st.slider("Pilih proporsi data uji:",
0.1, 0.5, 0.2, 0.05)
    k_value = st.slider("Pilih nilai K untuk KNN:", 1,
20, 5)
```

```
    if st.button("⚙️ Lakukan Split Data & Simpan
Parameter"):
        X_train, X_test, y_train, y_test =
train_test_split(
            st.session_state.X, st.session_state.y,
            test_size=test_size, random_state=42
        )

        st.session_state.X_train = X_train
        st.session_state.X_test = X_test
        st.session_state.y_train = y_train
        st.session_state.y_test = y_test
        st.session_state.k_value = k_value

        st.success(f"✅ Data berhasil di-split. Test
size = {test_size}, K = {k_value}")

    if st.button("➡️ Lanjut: Training & Evaluasi"):
        st.session_state.step = 5
        st.rerun()

# ======================
# STEP 5: Training & Evaluasi
# ======================
elif st.session_state.step == 5:
    st.subheader("🤖 Training & Evaluasi Model")

    X_train = st.session_state.X_train
    X_test = st.session_state.X_test
    y_train = st.session_state.y_train
    y_test = st.session_state.y_test
    k_value = st.session_state.k_value

    if st.button("⚙️ Jalankan Training & Evaluasi"):
        # --- Naive Bayes
        nb = GaussianNB()
        nb.fit(X_train, y_train)
        y_pred_nb = nb.predict(X_test)

        acc_nb = accuracy_score(y_test, y_pred_nb)
        pre_nb = precision_score(y_test, y_pred_nb,
average="weighted")
        rec_nb = recall_score(y_test, y_pred_nb,
average="weighted")
        f1_nb = f1_score(y_test, y_pred_nb,
average="weighted")

        # --- KNN
        knn = KNeighborsClassifier(n_neighbors=k_value)
        knn.fit(X_train, y_train)
```

```
        y_pred_knn = knn.predict(X_test)

        acc_knn = accuracy_score(y_test, y_pred_knn)
        pre_knn = precision_score(y_test, y_pred_knn,
average="weighted")
        rec_knn = recall_score(y_test, y_pred_knn,
average="weighted")
        f1_knn = f1_score(y_test, y_pred_knn,
average="weighted")

        # Simpan hasil
        st.session_state.results = {
            "Naive Bayes": (acc_nb, pre_nb, rec_nb,
f1_nb),
            "KNN": (acc_knn, pre_knn, rec_knn, f1_knn)
        }

        st.success("✅ Training & Evaluasi selesai!")

    if st.button("➡️ Lanjut: Bandingkan Algoritma"):
        st.session_state.step = 6
        st.rerun()

# ======================
# STEP 6: Perbandingan Algoritma
# ======================
elif st.session_state.step == 6:
    st.subheader("📊 Perbandingan Hasil Evaluasi")

    results = st.session_state.results
    df_results = pd.DataFrame(results,
index=["Accuracy", "Precision", "Recall", "F1-
Score"]).T
    st.table(df_results)

    if df_results.loc["KNN","Accuracy"] >
df_results.loc["Naive Bayes","Accuracy"]:
        st.success("👍 KNN memiliki performa lebih baik
pada dataset ini.")
    else:
        st.success("👍 Naive Bayes memiliki performa
lebih baik pada dataset ini.")
# === 2. Input data baru manual dengan dropdown ===
st.title("Klasifikasi Status Kesehatan (Input Manual
Banyak Data)")

st.info("Isi data baru menggunakan pilihan dropdown di
bawah ini, lalu tambahkan baris sesuai kebutuhan.")

# Pilihan untuk tiap atribut
```

```
options = {
    "Usia": ["Muda", "Tua"],
    "Jenis_kelamin": ["Pria", "Wanita"],
    "Merokok": ["Aktif", "Pasif"],
    "Bekerja": ["yes", "No"],
    "Aktivtas_Begadang": ["iya", "tidak"],
    "Aktivitas_Olahraga": ["Jarang", "Sering"],
    "pola makan teratur": ["Teratur", "Kurang"],
    "Penyakit_Bawaan": ["Ada", "Tidak Ada"]
}

# Template default
data_template = pd.DataFrame([{
    "Usia": "Muda",
    "Jenis_kelamin": "Pria",
    "Merokok": "Aktif",
    "Bekerja": "yes",
    "Aktivtas_Begadang": "iya",
    "Aktivitas_Olahraga": "Jarang",
    "pola makan teratur": "Teratur",
    "Penyakit_Bawaan": "Tidak Ada"
}])

# Data editor dengan dropdown
new_data = st.data_editor(
    data_template,
    num_rows="dynamic",
    column_config={
        col:
st.column_config.SelectboxColumn(options=opts,
required=True)
        for col, opts in options.items()
    },
    key="editor"
)

# === 3. Prediksi ===
if st.button("Klasifikasikan"):
    if not new_data.empty:
        # Encode data baru
        encoded_new = new_data.copy()
        for col in encoded_new.columns:
            if col in label_encoders:
                encoded_new[col] =
label_encoders[col].transform(encoded_new[col])

        # Prediksi
        nb_pred = nb_model.predict(encoded_new)
        knn_pred = knn_model.predict(encoded_new)

        # Decode hasil
```

```
        hasil_nb =
label_encoders["Hasil"].inverse_transform(nb_pred)
        hasil_knn =
label_encoders["Hasil"].inverse_transform(knn_pred)

        # Tambahkan hasil ke tabel
        new_data["Prediksi_NaiveBayes"] = hasil_nb
        new_data["Prediksi_KNN"] = hasil_knn

        st.success("Hasil Klasifikasi:")
        st.dataframe(new_data)

        # === Rekap jumlah prediksi (contoh: Tidak=4,
Ya=5) ===
        st.subheader("Rekapitulasi Prediksi")
        rekap_nb = pd.Series(hasil_nb).value_counts()
        rekap_knn = pd.Series(hasil_knn).value_counts()

        st.write("**Naive Bayes:**")
        for kelas, jumlah in rekap_nb.items():
            st.write(f"- {kelas} = {jumlah}")

        st.write("**KNN:**")
        for kelas, jumlah in rekap_knn.items():
            st.write(f"- {kelas} = {jumlah}")

        # === 4. Download ke Excel ===
        output = BytesIO()
        with pd.ExcelWriter(output,
engine="xlsxwriter") as writer:
            new_data.to_excel(writer, index=False,
sheet_name="Hasil Prediksi")
        excel_data = output.getvalue()

        st.download_button(
            label="📥 Download Hasil ke Excel",
            data=excel_data,
            file_name="hasil_klasifikasi.xlsx",
            mime="application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet"
        )
    else:
        st.warning("Harap isi data terlebih dahulu.")
```

Lampiran 5 Segmen Streamlit

F. Lampiran Hasil Cek Plagiasi

PONDOK PESANTREN SALAFIYAH SYAFI'IYAH SUKOREJO
UNIVERSITAS IBRAHIMY
PERPUSTAKAAN IBRAHIMY
NPP. 3512142F2006567
Jl. KHR. Syamsul Arifin No. 1-2 PO. Box. 2 Kode Pos. 68374 Phone (0338) 452666 Fax. (0338) 453068
SUMBEREJO BANYUPUTIH SITUBONDO JAWA TIMUR

SURAT KETERANGAN
HASIL PEMERIKSAAN PLAGIASI

Yang bertanda tangan di bawah ini

Nama	: Muhammad Ali Ridla, M.Kom.
Jabatan	: Kepala Perpustakaan

Menyatakan dengan sebenarnya bahwa:

NIM	: 2021503082
Nama	: NAZHIFATUL MUTHOHHAROH
Fakultas	: Sains dan Teknologi
Prodi	: Teknologi Informasi
Kecamatan	: PANJI
Kabupaten	: SITUBONDO
Provinsi	:
Judul Skripsi	: Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbor (Knn) Untuk Mengklasifikasikan Status Kesehatan

Dengan dosen Pembimbing :

1. Lukman Fakih Lidimilah, M.Kom
2. Ahmad Homaidi, M.Kom.

Telah dilakukan cek plagiasi di Perpustakaan Universitas Ibrahimy dengan persentase plagiasi terakhir sebesar **29%** .

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Sukorejo, 20 Agustus 2025
Kepala Perpustakaan,

Dokumen ini telah ditandatangani secara elektronik.

Muhammad Ali Ridla, M.Kom.

UU ITE No.11 Tahun 2008 Pasal 5 Ayat 1
"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti yang sah."

www.lib.ibrahimy.ac.id library@ibrahimy.ac.id Perpustakaan Ibrahimy @ibrahimy_lib

Lampiran 6 Hasil Cek Plagiasi

G. Lampiran Kesediaan Publikasi

LEMBAR PERNYATAAN
KESEDIAAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama	: Nazhifatul Muthohharoh
NIM/NPM	: 2021503082
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Jenis Karya Ilmiyh	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non-eksklusif *(Non-exclusive Royalty-Free Right)* kepada Perpustakaan Universitas Ibrahimy atas karya ilmiah saya berupa Skripsi yang berjudul:

" PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR UNTUK MENGKLASIFIKASIKAN STATUS KESEHATAN"

Dengan Hak Bebas Royalti Non-eksklusif ini Pusat Perpustakaan Universitas Ibrahimy berhak menyimpan, alih media/format, mengelola dalam bentuk pangkalan data *(database)*, merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya seabagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyatan ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Situbondo, 27 Agustus 2025
Yang Menyatakan

NAZHIFATUL MUTHOHHAROH

Lampiran 7 Kesediaan Publikasi Ilmiyah