PERBANDINGAN PERFORMA ALGORITMA KLASIFIKASI C4.5 DAN NAÏVE BAYES UNTUK PREDIKSI DIAGNOSA PENYAKIT DIABETES

SKRIPSI

Oleh:

Febri Basufi Bahtiarullah

2022503114

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY SITUBONDO

2024

PERBANDINGAN PERFORMA ALGORITMA KLASIFIKASI C4.5 DAN NAÏVE BAYES UNTUK PREDIKSI DIAGNOSA PENYAKIT DIABETES

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:
Febri Basufi Bahtiarullah
2022503114

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO
2024

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : **Febri Basufi Bahtiarullah**

NPM : 2022503114

Prodi : S-1 Teknologi Informasi

Fakultas : Fakultas Sains dan Teknologi

Menyatakan dengan sebenarnya, bahwa tugas skripsi ini secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali pada bagian-bagian yang dirujuk sebagai sumber referensi dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil plagiasi, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Situbondo, 16 Agustus 2024

Saya yang menyatakan,

Febri Basufi Bahtiarullah

PERSETUJUAN PEMBIMBING

Nama	:	**Febri Basufi Bahtiarullah**
NPM	:	2022503114
Judul	:	**PERBANDINGAN PERFORMA ALGORITMA KLASIFIKASI C4.5 DAN NAÏVE BAYES UNTUK PREDIKSI DIAGNOSA PENYAKIT DIABETES**

Telah disetujui oleh :

Pembimbing I,

Ahmad Homaidi, M.Kom
NIDN. 0705078901

Pembimbing II,

Firman Santoso, M.Kom
NIDN. 0722129201

PENGESAHAN

SKRIPSI

PERBANDINGAN PERFORMA ALGORITMA KLASIFIKASI C4.5 DAN NAÏVE BAYES UNTUK PREDIKSI DIAGNOSA PENYAKIT DIABETES

FEBRI BASUFI BAHTIARULLAH
2022503114

Telah dipertahankan di depan dewan penguji Sidang/Munaqasyah Skripsi pada hari Senin, Tanggal 19 Agustus 2024 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom) pada Fakultas Sains dan Teknologi Universitas Ibrahimy.

Tim Penguji,

Ketua Sidang,

Abdul Wafi, S.Pi, M.P
NIDN. 0705049103

Sekretaris Sidang,

Abdus Samad, M.Kom
NIDN. 0709099006

Penguji I,

Ahmad Lutfi, M.Kom
NIDN. 0714108803

Penguji II,

Farihin Lazim, M. Tr. T
NIDN. 0711099201

Mengetahui
Dekan,

Abd. Ghofur, M.Kom
NIDN: 0711088303

MOTTO

فَاصْبِرْ إِنَّ وَعْدَ اللَّهِ حَقٌّ ۖ وَلَا يَسْتَخِفَّنَّكَ الَّذِينَ لَا يُوقِنُونَ

"Dan bersabarlah kamu, sesungguhnya janji Allah adalah benar dan sekali-kali janganlah orang-orang yang tidak meyakini (kebenaran ayat-ayat Allah) itu menggelisahkan kamu."

(QS. Ar-Rum [60])

KATA PENGANTAR

Segala puji syukur penulis sampaikan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya, perencanaan, pelaksanaan dan penyelesaian tugas akhir/skripsi dengan judul "Perbandingan Performa Algoritma Klasifikasi C4.5 Dan Naïve Bayes Untuk Prediksi Diagnosa Penyakit Diabetes" sebagai salah satu syarat penyelesaian program diploma/sarjana dapat terselesaikan dengan baik dan lancar.

Kesuksesan ini dapat peneliti peroleh karena dukungan beberapa pihak. Peneliti menyampaikan terima kasih kepada :

1. **KHR. Ach Azaim Ibrahimy, S.Sy, M.H** selaku Pengasuh Pondok Pesantren Salafiyah Syafiiyah Sukorejo Situbondo.
2. **KH. Ach. Fadhail, M.H** selaku Rektor Universitas Ibrahimy.
3. **Bapak Abd. Ghofur, M.Kom** selaku Dekan Fakultas Sains dan Teknologi.
4. **Bapak Firman Santoso, M.Kom**, selaku Ka. Prodi Teknologi Informasi.
5. **Bapak Ahmad Homaidi, M.Kom** dan **Bapak Firman Santoso, M.Kom** selaku pembimbing I dan II.

Semoga amal baik yang telah diberikan oleh Bapak/Ibu kepada peneliti mendapat balasan yang sebaik mungkin dari Allah SWT, Amin.

Situbondo, 16 Agustus 2024

Peneliti

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR SEGMEN PROGRAM

Halaman

ABSTRAK

Febri Basufi Bahtiarullah. 2024. **Perbandingan Performa Algoritma Klasifikasi C4.5 Dan Naïve Bayes Untuk Prediksi Diagnosa Penyakit Diabetes.** Skripsi. Program Studi Teknologi Informasi. Universitas Ibrahimy. Pembimbing: (1) Ahmad Homaidi, M.Kom., (2) Firman Santoso, M.Kom.

Penyakit diabetes mellitus merupakan tantangan kesehatan global yang terus meningkat, ditandai oleh hiperglikemia akibat gangguan sekresi insulin atau resistensi insulin. Dengan meningkatnya jumlah penderita, penting untuk mengembangkan metode efektif dalam mendiagnosis dan memprediksi penyakit ini. Penelitian ini membandingkan performa dua algoritma klasifikasi, C4.5 dan Naïve Bayes, dalam memprediksi diagnosis diabetes menggunakan dataset dari UCI Machine Learning Repository yang terdiri dari 520 baris data dan 17 variabel. Metode penelitian menggunakan pendekatan CRISP-DM (Cross-Industry Standard Process for Data Mining) dan evaluasi performa model menggunakan metrik akurasi, presisi, dan recall yang diimplementasikan ke dalam pemrograman Python berbasis Web. Hasil menunjukkan bahwa C4.5 mencapai akurasi 100%, sedangkan Naïve Bayes mencapai 90,38%. Meskipun Naïve Bayes lebih cepat dalam waktu komputasi, C4.5 lebih unggul dalam akurasi prediksi. Kesimpulan penelitian menegaskan pentingnya pemilihan algoritma yang tepat berdasarkan karakteristik dataset dan tujuan analisis, dengan akurasi sebagai prioritas utama dalam diagnosis medis. Penelitian ini juga merekomendasikan eksplorasi algoritma tambahan seperti Random Forest atau Support Vector Machine untuk memahami performa dalam konteks yang lebih luas. Diharapkan, penelitian ini dapat menjadi acuan bagi praktisi kesehatan dan peneliti dalam meningkatkan akurasi diagnosis serta efektivitas intervensi medis untuk diabetes, serta mendorong penelitian lebih lanjut di bidang ini untuk meningkatkan kualitas layanan kesehatan dan deteksi dini penyakit diabetes.

Kata kunci: Diabetes, C4.5, Naïve Bayes, Python, Prediksi.

ABSTRACT

Febri Basufi Bahtiarullah. 2024. **Comparison of the Performance of C4.5 and Naïve Bayes Classification Algorithms for Predicting Diabetes Diagnosis**. Thesis. Information Technology Study Program. Ibrahimy University. Supervisors: (1) Ahmad Homaidi, M.Kom., (2) Firman Santoso, M.Kom.

Diabetes mellitus is a growing global health challenge characterized by hyperglycemia due to impaired insulin secretion or insulin resistance. With the increasing number of patients, it is crucial to develop effective methods for diagnosing and predicting this disease. This study compares the performance of two classification algorithms, C4.5 and Naïve Bayes, in predicting diabetes diagnosis using a dataset from the UCI Machine Learning Repository, which consists of 520 instances and 17 variables. The research methodology employs the CRISP-DM (Cross-Industry Standard Process for Data Mining) approach, and model performance is evaluated using metrics such as accuracy, precision, and recall, implemented in a web-based Python programming environment. The results indicate that C4.5 achieves an accuracy of 100%, while Naïve Bayes reaches 90.38%. Although Naïve Bayes is faster in computation time, C4.5 demonstrates superior predictive accuracy. The conclusion of this research emphasizes the importance of selecting the appropriate algorithm based on the characteristics of the dataset and the analysis objectives, with accuracy being the primary priority in medical diagnosis. This study also recommends exploring additional algorithms such as Random Forest or Support Vector Machine to gain a broader understanding of performance in various contexts. It is hoped that this research can serve as a reference for healthcare practitioners and researchers in enhancing diagnostic accuracy and the effectiveness of medical interventions for diabetes, as well as encouraging further research in this field to improve healthcare quality and early detection of diabetes.

Keywords: Diabetes, C4.5, Naïve Bayes, Python, Prediction.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Diabetes adalah kondisi di mana pankreas tidak memproduksi cukup insulin atau tubuh tidak dapat menggunakan insulin yang dihasilkan secara efektif. Menurut American Diabetes Association (ADA), diabetes didefinisikan sebagai kondisi yang mengalami hiperglikemia kronis, yang disebabkan oleh kekurangan insulin absolut atau relatif, gangguan kerja insulin, atau peningkatan produksi glukosa [1]. Saat ini, diabetes terus menjadi salah satu penyebab kematian utama di seluruh dunia, karena dapat menyebabkan berbagai komplikasi serius dan bahkan kematian..

Di Indonesia, masalah diabetes semakin mengkhawatirkan. Menurut data dari International Diabetes Federation (IDF), pada tahun 2021, Indonesia menempati peringkat kelima di dunia dalam hal jumlah penderita diabetes, dengan 19,5 juta orang. Proyeksi IDF untuk tahun 2045 menunjukkan bahwa jumlah penderita diabetes di Indonesia diperkirakan akan terus meningkat pesat, mencapai 28,6 juta jiwa dalam waktu kurang dari dua dekade mendatang [2]. Peningkatan ini mencerminkan tren yang mengarah pada epidemi diabetes di negara ini.

Lebih lanjut, laporan survei yang dilakukan oleh Kementerian Kesehatan (Kemenkes) melalui Survei Kesehatan Indonesia (SKI) tahun 2023 menunjukkan bahwa prevalensi diabetes melitus di Indonesia mengalami peningkatan signifikan, terutama di kalangan individu berusia di atas 15 tahun. Pada tahun 2018, prevalensi diabetes di Indonesia tercatat sebesar 10,9%, dan angka ini meningkat menjadi

11,7% pada tahun 2023 [3]. Peningkatan ini mencerminkan pertumbuhan jumlah penderita diabetes yang cukup signifikan dalam kurun waktu lima tahun, yang dapat disebabkan oleh berbagai faktor, termasuk perubahan gaya hidup, pola makan yang tidak sehat, serta kurangnya aktivitas fisik di kalangan masyarakat.

Peningkatan prevalensi diabetes ini tentunya menambah beban pada sistem kesehatan nasional dan menuntut adanya langkah-langkah strategis yang lebih efektif untuk penanggulangan dan pencegahan penyakit ini. Upaya-upaya seperti edukasi kesehatan, promosi pola hidup sehat, serta peningkatan akses terhadap layanan kesehatan menjadi sangat penting untuk mengendalikan laju peningkatan prevalensi diabetes di Indonesia. Tanpa intervensi yang tepat dan segera, risiko komplikasi serius akibat diabetes, seperti penyakit kardiovaskular, kerusakan ginjal, dan gangguan penglihatan, akan semakin tinggi, yang dapat berdampak buruk pada kualitas hidup masyarakat..

Dalam konteks ini, teknologi informasi dan data mining memainkan peran penting. Data mining adalah proses ekstraksi informasi penting dari kumpulan data besar yang beragam dan kompleks, yang dapat membantu dalam diagnosis dan pengelolaan diabetes. Dengan menggunakan algoritma dan teknik pembelajaran mesin, kita dapat menemukan pola, tren, dan hubungan dalam data yang dapat membantu membuat keputusan yang lebih baik dalam penanganan diabetes. Dalam langkah mendukung hal tersebut, penulis melakukan pengumpulan dataset diagnosis diabetes yang didapat dari *UCI Machine Learning Repository* untuk kemudian dilakukan proses data mining. Namun, teridentifikasi kendala bahwa dataset tersebut belum terklasifikasi dengan memadai.

Berdasarkan latar belakang masalah dan metode penanggulangan terhadap masalah yang telah dijabarkan tersebut. Maka penulis bermaksud untuk melakukan sebuah penelitian tentang "**Perbandingan Performa Algoritma Klasifikasi C4.5 Dan Naïve Bayes Untuk Prediksi Diagnosa Penyakit Diabetes".** Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja algoritma C4.5 dan Naive Bayes dalam prediksi diagnosis diabetes karena keduanya memiliki kelebihan dan kekurangan masing-masing.

1.2. Identifikasi Masalah

Berdasarkan latar belakang dari permasalahan tersebut, dapat diidentifikasi beberapa permasalahan, sebagai berikut:

a. Dataset yang didapat belum terklasifikasi dengan baik.
b. Nilai perbandingan performa algoritma klasifikasi C4.5 dan algoritma Naïve Bayes belum diketahui.
c. Referensi yang ada kebanyakan masih belum melakukan implementasi ke bahasa pemrograman python

1.3. Rumusan Masalah

Setelah mengidentifikasi masalah ke dalam beberapa catatan penting, kemudian dapat dirumuskan permasalahan inti yang akan dibahas dalam penelitian ini yaitu membandingkan performa kedua algoritma klasifikasi c4.5 dan algoritma naive bayes dalam memprediksi diagnosa penyakit diabetes.

1.4. Batasan Masalah

a. Klasifikasi Dataset

Penelitian ini dibatasi pada dataset diagnosa diabetes yang belum terklasifikasi dengan baik, sehingga proses klasifikasi awal seperti preprocessing data harus dilakukan terlebih dahulu.

b. Evaluasi Performa Algoritma

Studi ini mengevaluasi performa algoritma C4.5 dan Naïve Bayes menggunakan metrik tertentu seperti akurasi dan F1-scoree.

c. Implementasi dalam Python

Implementasi algoritma dilakukan menggunakan bahasa pemrograman Python dengan keterbatasan pada penggunaan library tertentu seperti scikit-learn dan streamlit.serta library lainnya yang dibutuhkan.

1.5. Tujuan Penelitian

Studi ini bertujuan untuk mengevaluasi dan membandingkan kinerja algoritma klasifikasi C4.5 dan Naive Bayes dalam prediksi diagnosis diabetes. Kemudian penelitian ini juga bertujuan untuk menentukan algoritma mana yang lebih efektif dalam membuat prediksi yang akurat dan efisien dengan menggunakan dataset publik yang disediakan oleh *UC Irvine Machine Learning Repositor*y.

1.6. Manfaat Penelitian

Sedangkan manfaat dari perbandingan performa Algoritma Klasifikasi C4.5 dan Naïve Bayes untuk prediksi diagnosa penyakit diabetes ini diantaranya adalah:

a. Manfaat Teoritis

Penelitian ini dapat menjadi referensi tambahan bagi peneliti selanjutnya yang akan meneliti tentang perbandingan dua atau lebih penggunaan algoritma data mining untuk bidang kesehatan maupun bidang bisnis dan lainnya.

b. Manfaat bagi Universitas

Dapat menjadi salah satu sumber referensi maupun sumbangsih dalam pengakreditasian program studi sehingga dapat meningkatkan taraf akreditasi universitas yang bersangkutan.

c. Manfaat bagi pemangku kepentingan, dalam hal ini Tenaga Kesehatan

Penelitian ini juga bertujuan untuk meningkatkan pemahaman tentang penggunaan teknik data mining dalam bidang kesehatan, khususnya dalam deteksi dan penanganan penyakit kronis seperti diabetes, dan untuk menawarkan solusi praktis dan berbasis data untuk meningkatkan kualitas layanan kesehatan.

d. Manfaat bagi Masyarakat yang berpotensi menderita penyakit diabetes

Dengan pemahaman yang lebih baik tentang teknik data mining dan penerapannya dalam bidang kesehatan, masyarakat dapat lebih menyadari pentingnya data dan teknologi dalam mendukung kesehatan mereka. Ini juga dapat mendorong peningkatan literasi digital dan kesehatan di kalangan masyarakat, sehingga mereka lebih proaktif dalam menjaga kesehatan dan mengelola risiko penyakit..

1.7. Metodologi Penelitian

1.7.1. Jenis Penelitian

a. Penelitian Library Research :

Penelitian kepustakaan diartikan sebagai metode penelitian yang dilakukan dengan cara mengumpulkan dan menganalisis berbagai sumber informasi yang ada di perpustakaan atau sumber tertulis lainnya. Penelitian ini bertujuan untuk mengkaji dan mengeksplorasi teori, konsep, dan hasil penelitian terdahulu yang relevan dengan topik yang diteliti [4].

b. Penelitian Field Research:

Field Research merupakan metode penelitian yang melibatkan observasi langsung dengan dukungan data primer dan sekunder yang diperoleh melalui penelitian kepustakaan [5]. Pengertian lainnya adalah .merupakan metode penelitian yang dilakukan langsung di lokasi atau lingkungan nyata di mana fenomena atau subjek yang diteliti berada. Tujuan utama dari field research adalah untuk mengumpulkan data langsung dari sumber aslinya dalam konteks alami, sehingga peneliti bisa mendapatkan pemahaman yang lebih mendalam dan akurat tentang perilaku, interaksi, atau kondisi yang terjadi.

c. Penelitian Eksperimen:

Dalam penelitian ini, dilakukan eksperimen dengan menggunakan dataset untuk melatih dan menguji model algoritma C4.5 dan Naive Bayes. Penelitian ini melibatkan proses eksperimental yang sistematis untuk membandingkan hasil prediksi dari kedua algoritma tersebut. Melalui

eksperimen ini, peneliti dapat mengevaluasi performa masing-masing algoritma berdasarkan metrik evaluasi tertentu, seperti akurasi, presisi, recall, dan f1-score

1.7.2. Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini adalah studi literatur dengan memanfaatkan data sekunder dari Platform UC Irvine Machine Learning Repository. Dataset berjudul " Early Stage Diabetes Risk Prediction " ini berasal dari Informasi yang dikumpulkan menggunakan kuesioner langsung dari pasien Rumah Sakit Diabetes Sylhet di Sylhet, Bangladesh dan disetujui oleh seorang dokter.

Tautan yang dapat diakses adalah sebagai berikut https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset. Data terdiri dari 520 baris data dengan memiliki 17 variabel yang terdiri dari 16 atribut dan 1 kelas yang berpengaruh terhadap diagnosis penyakit diabetes.

1.8. Sistematika Pembahasan

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

BAB I: PENDAHULUAN

Bab ini menyajikan latar belakang penelitian, identifikasi masalah yang dihadapi, serta perumusan dan batasan masalah yang akan dijelajahi. Selain itu, bab ini juga menjelaskan tujuan dan manfaat penelitian, menjelaskan jenis metode serta teknik yang digunakan dalam penelitian, serta mendetailkan teknik pengumpulan data dan teknik pengembangan sistem. Sistematika pembahasan juga akan

dijelaskan untuk memberikan panduan yang jelas tentang bagaimana pembahasan dalam penelitian ini disusun.

BAB II: TINJAUAN PUSTAKA

Bab ini mengulas secara komprehensif penelitian-penelitian sebelumnya yang relevan dengan topik penelitian ini. Selain itu, bab ini juga membahas landasan teori yang mendasari penelitian, metodologi pemodelan yang diterapkan, serta perangkat lunak yang digunakan dalam proses penelitian. Penjelasan ini bertujuan untuk memberikan konteks dan dasar yang kuat bagi penelitian yang dilakukan.

BAB III: ANALISIS DAN PERANCANGAN SISTEM

Bab ini menguraikan secara rinci bagaimana peneliti mengumpulkan data yang diperlukan untuk penelitian ini dan bagaimana proses pengolahan data dilakukan sebelum melanjutkan ke tahap analisis. Proses ini melibatkan teknik pengolahan data yang digunakan untuk memastikan bahwa data siap dan sesuai untuk analisis menggunakan algoritma data mining yang telah dipilih.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari analisis data yang telah dilakukan, lengkap dengan penjelasan mendetail dan perhitungan dari data yang telah diproses. Penjelasan ini mencakup pembuatan prediksi berdasarkan hasil analisis serta evaluasi tingkat akurasi dari prediksi yang dihasilkan. Bab ini bertujuan untuk memberikan wawasan yang jelas mengenai temuan dari penelitian dan interpretasinya.

BAB V: PENUTUP

Bab terakhir ini membahas hasil penelitian secara keseluruhan dan memberikan rekomendasi berdasarkan sistem yang telah diteliti, dirancang, dan diterapkan. Rekomendasi ini bertujuan untuk memberikan saran yang konstruktif mengenai pengembangan lebih lanjut dan implementasi sistem, serta dampak potensial dari penelitian terhadap praktik dan kebijakan terkait.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Analisis Performa Algoritma Klasifikasi Naive Bayes dan C4.5 untuk Prediksi Penerima Bantuan Jaminan Kesehatan.

Penelitian ini dilakukan oleh Nurfazriah Attamami , Agung Triayudi , Rima Tamara Aldisa, mahasiswa dari Universitas Nasional, Kota Jakarta Selatan. Penelitian ini dipublikasikan pada bulan April 2023 melalui Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi).

Penelitian ini berfokus pada analisis performa dua algoritma klasifikasi, yaitu Naive Bayes dan C4.5, dalam memprediksi kelayakan calon penerima bantuan jaminan kesehatan. Latar belakang penelitian ini didasari oleh tantangan yang dihadapi oleh petugas dalam mengidentifikasi calon penerima bantuan kesehatan di tengah banyaknya data pasien yang harus diperiksa. Dengan meningkatnya jumlah individu yang membutuhkan akses ke jaminan kesehatan, penting untuk memiliki metode yang efisien dan akurat dalam menentukan siapa yang berhak menerima bantuan tersebut.

Dalam penelitian ini, peneliti menggunakan dataset yang terdiri dari 1043 record, yang dibagi menjadi dua bagian: 730 record digunakan sebagai data latih dan 313 record sebagai data uji. Proses pengolahan data dimulai dengan konversi nilai non-numerik menjadi nilai numerik agar dapat diproses oleh algoritma. Selain itu, kolom yang tidak relevan, seperti nama, usia, dan pekerjaan, dihapus untuk

memastikan bahwa hanya atribut yang diperlukan yang dipertimbangkan dalam analisis.

Hasil pengujian menunjukkan bahwa algoritma C4.5 memiliki performa yang lebih baik dibandingkan dengan Naive Bayes. C4.5 mencapai akurasi sebesar 99.04%, dengan hanya 3 record yang diprediksi salah, sehingga tingkat kesalahan atau error rate-nya hanya 0.96%. Dalam hal ini, C4.5 juga menunjukkan nilai presisi sebesar 57.74%, recall sebesar 99.44%, specificity sebesar 98.50%, dan F1 scoree sebesar 73.06%. Ini menunjukkan bahwa algoritma C4.5 sangat efektif dalam mengidentifikasi calon penerima bantuan yang layak.

Di sisi lain, algoritma Naive Bayes memperoleh akurasi 92.97%, dengan 22 record yang diprediksi salah, menghasilkan tingkat kesalahan sebesar 7.03%. Meskipun akurasi Naive Bayes masih di atas 90%, performanya tidak sebanding dengan C4.5. Nilai presisi Naive Bayes adalah 61.86%, recall 89.55%, specificity 99.11%, dan F1 scoree 73.17%. Hasil ini menunjukkan bahwa meskipun Naive Bayes dapat digunakan untuk klasifikasi, C4.5 lebih unggul dalam hal akurasi dan kemampuan untuk meminimalkan kesalahan.

Penelitian ini juga membandingkan hasil dengan penelitian sebelumnya yang menggunakan algoritma serupa. Beberapa studi menunjukkan bahwa algoritma C4.5 dan Naive Bayes memiliki performa yang bervariasi tergantung pada konteks dan jenis data yang digunakan. Misalnya, penelitian oleh Sulihati (2022) menunjukkan bahwa Naive Bayes memiliki akurasi 91.08% dalam konteks seleksi penerimaan perguruan tinggi, sementara penelitian lain menunjukkan C4.5

dapat mencapai akurasi hingga 98.30% dalam memprediksi kriteria calon penerima bantuan sosial.

Dari hasil penelitian ini, dapat disimpulkan bahwa kedua algoritma memiliki potensi yang baik dalam memprediksi kelayakan penerima bantuan jaminan kesehatan. Namun, C4.5 terbukti lebih efektif dan akurat dalam konteks ini. Temuan ini diharapkan dapat memberikan kontribusi bagi pengembangan sistem pendukung keputusan yang lebih baik dalam sektor kesehatan, serta membantu petugas dalam proses identifikasi calon penerima bantuan secara lebih efisien dan akurat. Dengan demikian, penelitian ini tidak hanya memberikan wawasan tentang algoritma yang digunakan, tetapi juga berpotensi meningkatkan aksesibilitas dan keadilan dalam program jaminan kesehatan bagi masyarakat yang membutuhkan [6].

Klasifikasi Kesiapan Anak Masuk Sekolah Dasar Menggunakan Algoritma Naïve Bayes dan Algoritma C4.5

Penelitian yang berjudul "Klasifikasi Kesiapan Anak Masuk Sekolah Dasar Menggunakan Algoritma Naïve Bayes dan Algoritma C4.5" oleh M. Rudi Fanani dan Dona Siska Sintia bertujuan untuk menganalisis dan memprediksi kesiapan anak-anak dari TK Pratiwi untuk memasuki sekolah dasar. Sekolah dasar merupakan tahap penting dalam pendidikan anak, yang menuntut kesiapan yang lebih serius dibandingkan dengan saat mereka berada di taman kanak-kanak. Oleh karena itu, penelitian ini berfokus pada pengembangan metode yang dapat membantu dalam menentukan apakah seorang anak siap atau belum siap untuk memasuki sekolah dasar.

Proses penelitian dimulai dengan tahap pemahaman bisnis, di mana peneliti melakukan survei lapangan untuk mengidentifikasi kondisi dan permasalahan yang dihadapi anak-anak di TK Pratiwi. Peneliti juga mengumpulkan data mengenai faktor-faktor yang mempengaruhi kesiapan anak untuk masuk sekolah dasar. Data yang dikumpulkan kemudian dianalisis untuk mendapatkan pemahaman yang lebih dalam tentang karakteristik siswa yang lulus ke sekolah dasar.

Tahap berikutnya adalah persiapan data, di mana data mentah yang telah dikumpulkan dipersiapkan untuk pemodelan. Proses ini melibatkan pembersihan dan pengolahan data agar siap untuk digunakan dalam analisis lebih lanjut. Setelah data siap, peneliti menggunakan dua algoritma, yaitu Naïve Bayes dan C4.5, untuk melakukan klasifikasi. Algoritma Naïve Bayes bekerja berdasarkan probabilitas, di mana peneliti menghitung probabilitas hipotesis untuk masing-masing kelas, yaitu siswa yang siap dan belum siap masuk sekolah dasar. Sementara itu, algoritma C4.5 menggunakan pendekatan pohon keputusan yang memerlukan perhitungan nilai entropy dan gain untuk menentukan atribut yang paling relevan.

Evaluasi model dilakukan dengan menggunakan confusion matrix, yang mencakup metrik seperti akurasi, recall, dan presisi. Hasil evaluasi menunjukkan bahwa algoritma Naïve Bayes memiliki akurasi sebesar 96,30%, dengan nilai presisi untuk kelas "belum siap" mencapai 100% dan untuk kelas "siap" sebesar 95%. Sementara itu, algoritma C4.5 juga menunjukkan hasil yang baik, dengan nilai recall untuk kelas "belum siap" sebesar 87,50% dan untuk kelas "siap" mencapai 100%. Hasil ini menunjukkan bahwa kedua algoritma memiliki kemampuan yang baik dalam memprediksi kesiapan anak untuk masuk sekolah dasar.

Penelitian ini menekankan pentingnya penggunaan metode data mining dalam menganalisis data pendidikan. Dengan memanfaatkan algoritma Naïve Bayes dan C4.5, peneliti dapat memberikan informasi yang berharga bagi pendidik dan orang tua dalam memahami kesiapan anak untuk memasuki sekolah dasar. Hasil dari penelitian ini diharapkan dapat digunakan sebagai acuan dalam pengambilan keputusan terkait pendidikan anak, serta sebagai dasar untuk penelitian lebih lanjut di bidang yang sama [7].

Perbandingan Klasifikasi Naive Bayes dan C4.5 untuk Diagnosa Penyakit Stroke

Penelitian ini berfokus pada perbandingan antara dua algoritma klasifikasi, yaitu Naive Bayes dan C4.5, dalam mendiagnosis penyakit stroke. Stroke merupakan salah satu penyakit dengan angka kematian tinggi di Indonesia, dan pemahaman yang lebih baik tentang faktor-faktor yang mempengaruhi penyakit ini sangat penting untuk meningkatkan diagnosis dan pengobatan. Penelitian ini dilakukan oleh Fitri et al. dan dipresentasikan dalam seminar SENTIMAS yang diselenggarakan oleh Institut Riset dan Publikasi Indonesia (IRPI).

Dataset yang digunakan dalam penelitian ini diambil dari Kaggle, yang terdiri dari 5110 rekaman data dengan 12 atribut. Proses pra-pemrosesan data dilakukan untuk memastikan kualitas data yang tinggi, termasuk penghilangan outlier, noise, dan data yang tidak konsisten. Dari total 5110 rekaman, 1767 rekaman dihapus karena tidak memenuhi kriteria, sehingga menyisakan 3343 rekaman yang digunakan untuk analisis lebih lanjut.

Algoritma C4.5 digunakan untuk membangun pohon keputusan yang memuat aturan klasifikasi. Proses ini melibatkan pemilihan atribut sebagai akar pohon, pembuatan cabang untuk setiap nilai atribut, dan pemisahan kasus di setiap cabang hingga semua kasus dalam cabang tersebut memiliki kelas yang sama. Di sisi lain, algoritma Naive Bayes merupakan metode klasifikasi statistik yang memperkirakan kemungkinan pengungkapan kelas berdasarkan Teorema Bayes. Algoritma ini mengasumsikan independensi antar atribut, yang membuatnya efisien dalam menangani dataset besar.

Hasil dari penelitian menunjukkan bahwa kedua algoritma memiliki kelebihan dan kekurangan masing-masing dalam hal akurasi dan kecepatan. Naive Bayes cenderung lebih cepat dalam proses klasifikasi, tetapi mungkin kurang akurat dibandingkan C4.5 dalam beberapa kasus. Sebaliknya, C4.5 dapat memberikan hasil yang lebih akurat tetapi memerlukan waktu pemrosesan yang lebih lama. Penelitian ini menekankan pentingnya pemilihan algoritma yang tepat berdasarkan karakteristik dataset dan tujuan analisis [8].

2.2 Landasan Teori

Agar penelitian ini memiliki landasan yang kuat, maka perlu dipahami terlebih dahulu definisi-definisi kunci yang akan menjadi fokus analisis. Definisi-definisi tersebut adalah sebagai berikut :

2.2.1 Diabetes

Menurut *American Diabetes Association*, definisi Diabetes mellitus adalah sekelompok penyakit metabolik yang ditandai oleh hiperglikemia, yaitu kondisi di mana kadar glukosa dalam darah meningkat secara abnormal. Hiperglikemia ini

terjadi akibat gangguan dalam sekresi insulin, aksi insulin, atau kombinasi keduanya. Insulin adalah hormon yang diproduksi oleh sel-sel beta pankreas dan berfungsi untuk mengatur kadar glukosa dalam darah dengan memfasilitasi penyerapan glukosa oleh sel-sel tubuh. Dalam diabetes, baik sekresi insulin yang tidak memadai maupun resistensi terhadap aksi insulin dapat menyebabkan peningkatan kadar glukosa.

Penyebab diabetes sangat bervariasi dan dapat mencakup faktor genetik, lingkungan, dan gaya hidup. Diabetes dibagi menjadi beberapa tipe, dengan tipe 1 dan tipe 2 sebagai yang paling umum. Tipe 1 diabetes biasanya disebabkan oleh penghancuran autoimun sel-sel beta pankreas, yang mengakibatkan defisiensi insulin absolut. Sementara itu, tipe 2 diabetes lebih umum terjadi dan sering kali terkait dengan resistensi insulin, di mana sel-sel tubuh tidak merespons insulin dengan baik, meskipun kadang-kadang sekresi insulin masih ada [1].

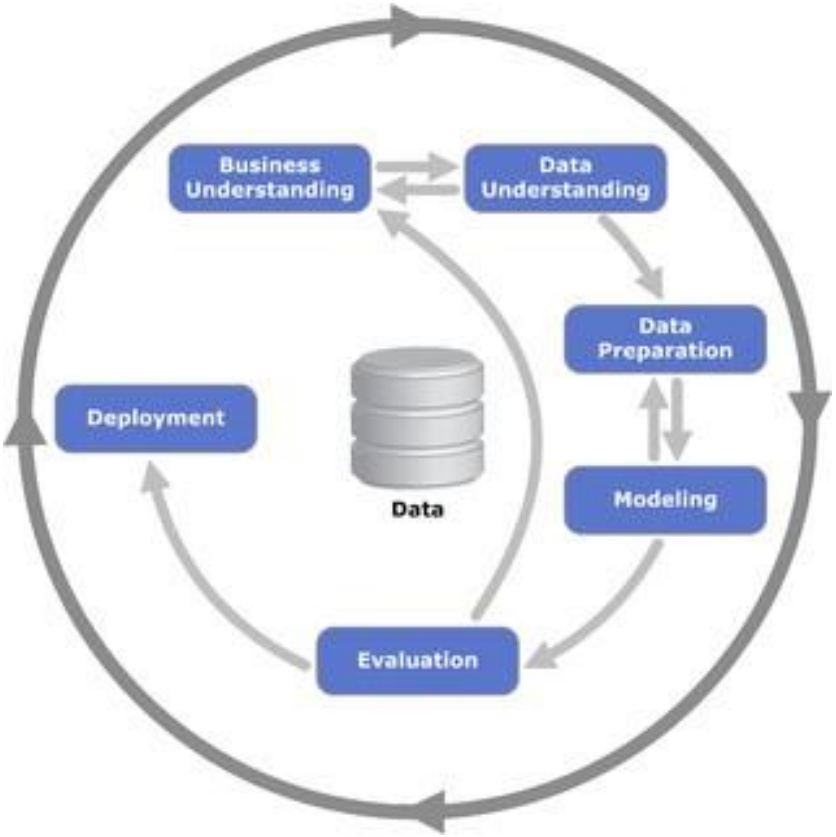
2.2.2 Data Mining

Data mining adalah proses yang kompleks dan terstruktur untuk menemukan pola, tren, dan informasi berharga dari kumpulan data besar dengan menggunakan teknik statistik, algoritma, dan pembelajaran mesin. Proses ini mencakup beberapa langkah, termasuk pembersihan data, integrasi data, dan transformasi data, yang bertujuan untuk mempersiapkan data sebelum analisis. Dengan menerapkan metode ini, organisasi dapat mengidentifikasi hubungan yang tidak terlihat sebelumnya dalam data, yang dapat digunakan untuk pengambilan keputusan yang lebih baik di berbagai bidang, seperti bisnis, kesehatan, dan ilmu sosial. Data mining memungkinkan transformasi data mentah menjadi wawasan

yang dapat ditindaklanjuti, meningkatkan efisiensi, efektivitas, dan daya saing organisasi [9].

2.2.3 CRISP-DM

CRISP-DM, yang merupakan singkatan dari Cross-Industry Standard Process for Data Mining, adalah sebuah metodologi yang banyak digunakan dalam proyek data mining.



Gambar 2. 1 *CRISP-DM Methodology*

Metodologi ini terdiri dari enam tahap yang bersifat iteratif, di mana setiap tahap memiliki tugas dan langkah-langkah spesifik yang harus diikuti.

1. Pemahaman Bisnis (*Business Understanding*)

   Tahap pertama ini berfokus pada pemahaman yang mendalam mengenai tujuan bisnis yang ingin dicapai serta bagaimana analisis data dapat

berkontribusi untuk mencapainya. Langkah-langkah dalam tahap ini meliputi penetapan tujuan proyek, identifikasi masalah yang ada, dan perumusan pertanyaan bisnis yang jelas.

2. Pemahaman Data (*Data Understanding*)

   Pada tahap ini, data yang relevan dikumpulkan, dieksplorasi, dan dievaluasi. Proses ini bertujuan untuk mendapatkan pemahaman yang lebih baik tentang data yang tersedia, kualitasnya, serta bagaimana data tersebut berhubungan dengan pertanyaan bisnis yang telah dirumuskan sebelumnya.

3. Persiapan Data (*Data Preparation*)

   Tahap ini berfokus pada persiapan data yang diperlukan untuk analisis lebih lanjut. Proses ini mencakup pembersihan data, pemilihan atribut yang relevan, transformasi data, dan pengorganisasian data agar siap digunakan dalam model analisis.

4. Pemodelan (*Modeling*)

   Dalam tahap ini, berbagai teknik pemodelan diterapkan untuk menganalisis data. Pemilihan model yang tepat dan penerapan algoritma yang sesuai sangat penting untuk mendapatkan hasil yang akurat.

5. Evaluasi (*Evaluation*)

   Setelah model dibangun, tahap evaluasi dilakukan untuk menilai seberapa baik model tersebut memenuhi tujuan bisnis yang telah ditetapkan. Jika hasilnya tidak memuaskan, peneliti dapat kembali ke tahap sebelumnya untuk melakukan perbaikan.

6. Penerapan (*Deployment*)

Tahap terakhir ini melibatkan penerapan model yang telah dikembangkan ke dalam lingkungan nyata. Ini termasuk pengembangan rencana untuk implementasi dan pemantauan hasil dari model yang diterapkan.

Metodologi CRISP-DM memberikan kerangka kerja yang sistematis dan terstruktur untuk membantu peneliti dan praktisi dalam menjalankan proyek data mining secara efektif.

2.2.4 Klasifikasi

Klasifikasi adalah salah satu metode dalam data mining yang bertujuan untuk mengelompokkan data ke dalam kategori atau kelas tertentu berdasarkan karakteristik yang dimilikinya. Dalam proses ini, sebuah model atau algoritma pembelajaran mesin dibangun dari dataset yang telah diklasifikasikan sebelumnya (disebut juga dataset pelatihan). Model ini kemudian digunakan untuk mengklasifikasikan data baru yang belum dikenal. Klasifikasi sering digunakan dalam berbagai aplikasi seperti deteksi penipuan, diagnosis medis, dan pengenalan pola. Teknik ini melibatkan beberapa langkah penting termasuk pemilihan fitur, pelatihan model, evaluasi kinerja, dan penerapan model pada data baru. Ian Witten dalam bukunya menekankan pentingnya memahami algoritma yang digunakan, seperti pohon keputusan, naive Bayes, dan SVM, serta teknik-teknik untuk meningkatkan akurasi model seperti pemilihan fitur yang tepat dan validasi silang[10].

2.2.5 Algoritma C4.5

Berdasarkan buku "Machine Learning: A Probabilistic Perspective" karya Kevin P. Murphy. Algoritma C4.5 adalah salah satu algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan. Algoritma ini merupakan pengembangan dari algoritma ID3 dan digunakan untuk klasifikasi. C4.5 bekerja dengan membagi dataset menjadi subset berdasarkan atribut yang memberikan keuntungan informasi tertinggi, yaitu atribut yang paling efektif dalam memisahkan data ke dalam kelas yang berbeda [11].

Prosesnya dimulai dengan memilih atribut yang terbaik sebagai node root, kemudian dataset dibagi berdasarkan nilai-nilai dari atribut tersebut. Langkah ini diulangi secara rekursif untuk setiap cabang yang dihasilkan sampai semua data dalam cabang tersebut termasuk dalam satu kelas, atau sampai atribut yang tersisa tidak memberikan keuntungan informasi yang signifikan.

C4.5 juga memiliki beberapa fitur penting seperti penanganan nilai yang hilang, kemampuan untuk menangani atribut numerik dan kategorikal, serta pruning pohon untuk mencegah overfitting dengan memangkas cabang-cabang yang tidak memberikan kontribusi signifikan terhadap akurasi klasifikasi.

Dalam buku tersebut, Kevin P. Murphy menjelaskan bahwa algoritma C4.5 adalah salah satu algoritma pohon keputusan yang paling populer dan banyak digunakan dalam aplikasi pembelajaran mesin karena kemampuannya yang efektif dan efisien dalam menangani data dengan berbagai tipe atribut.

2.2.6 Algoritma Naive Bayes

Naive Bayes adalah salah satu metode klasifikasi probabilistik yang sederhana namun sangat efektif, dan ia berakar pada prinsip dasar dari Teorema Bayes. Metode ini mengkomparasikan nilai posterior dengan nilai posterior lainnya. Kelas dengan nilai peluang posterior tertinggi diklasifikasikan sebagai kelas positif atau negative [12]. Meskipun nama "Naive" mungkin terdengar merendahkan, algoritma ini sering kali memberikan hasil yang sangat baik dalam berbagai aplikasi, terutama ketika data memiliki fitur-fitur yang saling bergantung.

Prinsip Dasar Teorema Bayes, Naive Bayes didasarkan pada Teorema Bayes, yang merupakan prinsip dasar dalam probabilitas. Teorema ini menghubungkan probabilitas dari suatu kategori (kelas) dengan probabilitas fitur-fitur yang diamati, menggunakan formula:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

dengan :

B : data dengan class yang belum diketahui

A : Hipotesis data B

P(A|B) :Probabilitas A berdasarkan B

P(B|A) : Probabilitas B berdasarkan A

P(A) : Probabilitas dari A

P(B) : Probabilitas dari B

Dengan menggunakan teorema ini, Naive Bayes bertujuan untuk menghitung probabilitas setiap kategori berdasarkan data fitur yang ada, dan

2.3 Perangkat Lunak yang Digunakan

Berikut adalah beberapa perangkat lunak atau software yang akan dipakai dalam pembuatan aplikasi dan identifikasi data sehingga dapat terealisasi dengan baik dan mendapatkan hasil yang optimal.

2.3.1 RapidMiner Studio

RapidMiner pertama kali dikembangkan dengan nama YALE (Yet Another Learning Environment) pada tahun 2001 oleh Ralf Klinkenberg, Ingo Mierswa, dan Simon Fischer di Unit Kecerdasan Buatan dari Universitas Dortmund. RapidMiner adalah perangkat lunak open source yang memungkinkan pengembangannya oleh siapa saja tanpa perlu membayar royalti kepada pembuatnya. Perangkat lunak ini memiliki lebih dari 500 operator untuk preprocessing dan visualisasi data. RapidMiner dikembangkan menggunakan bahasa pemrograman Java, sehingga dapat digunakan di berbagai sistem operasi. Selain itu, RapidMiner menawarkan antarmuka grafis (GUI) yang memungkinkan pengguna merancang alur analitik. Antarmuka ini menghasilkan file XML yang mendefinisikan proses analitik yang diinginkan untuk diterapkan pada data. File XML ini kemudian diproses dan dianalisis secara otomatis oleh RapidMiner. Pada tahun 2010-2011, RapidMiner dinyatakan sebagai perangkat lunak data mining terbaik dalam survei oleh KDnuggets, sebuah portal data mining

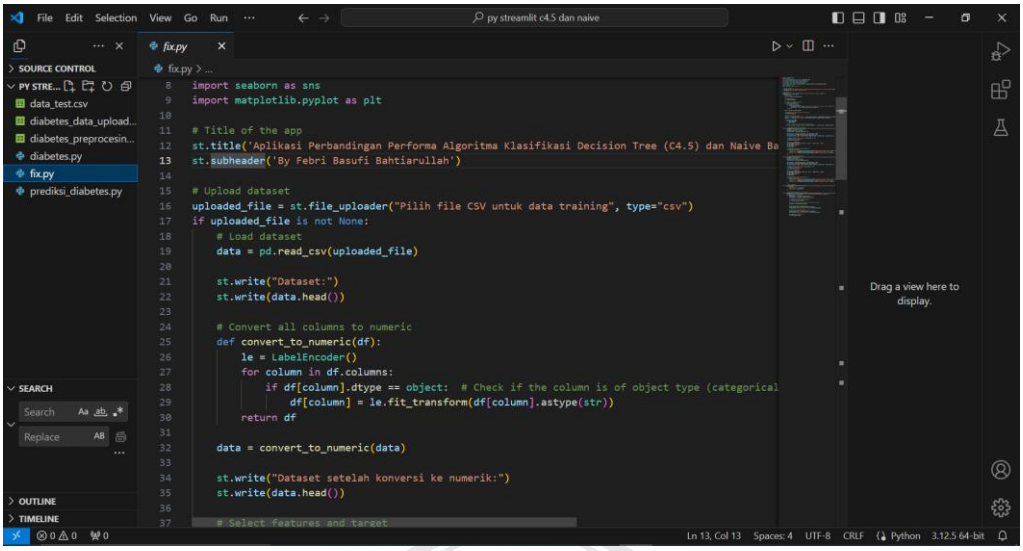
2.3.2 Microsoft Excel 2019

Microsoft Excel adalah salah satu aplikasi dalam paket Microsoft Office yang juga mencakup Microsoft Word, Microsoft PowerPoint, dan lainnya. Versi terbaru dari Microsoft Office adalah Microsoft Office 365. Dalam proses data

mining, aplikasi pertama yang digunakan untuk pencatatan data adalah Microsoft Excel, khususnya versi 2019. Excel berfungsi sebagai alat pengolah angka berbasis spreadsheet, yang memungkinkan pembuatan dan pengolahan data dalam format baris dan kolom sesuai kebutuhan pengguna. Excel sangat populer di seluruh dunia untuk pembuatan data karena kemampuannya dalam mengolah angka dan perhitungan dengan bantuan berbagai rumus, mempermudah pengolahan data, terutama dalam konteks keuangan perusahaan.

2.3.3 Visual Studio Code

Visual Studio Code (VS Code) merupakan editor kode sumber yang dikembangkan oleh Microsoft, yang dirancang untuk memberikan pengalaman pengembangan yang efisien dan produktif. VS Code dikenal karena kemampuannya yang luas dalam menangani berbagai bahasa pemrograman berkat dukungan ekstensi yang kaya. Jurnal tersebut menjelaskan bahwa fitur utama VS Code meliputi sistem debugging yang terintegrasi, terminal built-in, serta kemampuan untuk mengelola proyek secara efektif melalui sistem kontrol versi seperti Git. Keunggulan lain yang diungkapkan adalah antarmuka pengguna yang responsif dan kustomisasi yang fleksibel, yang memungkinkan pengembang untuk menyesuaikan lingkungan pengembangan sesuai dengan kebutuhan spesifik mereka. Nurul H. juga menekankan bahwa VS Code adalah pilihan populer di kalangan pengembang modern karena kemampuannya untuk meningkatkan produktivitas dan menyediakan alat yang mendukung berbagai tahap siklus pengembangan perangkat lunak [13].



Gambar 2. 2 Tampilan Text Editor VSCode Studio

2.3.4 Python

Dalam jurnal Pengertian Python diuraikan sebagai bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Python dikenal karena sintaksisnya yang sederhana dan mudah dibaca, yang memudahkan pengembang untuk menulis dan memahami. Python memiliki berbagai fungsi, termasuk pemrograman web, analisis data, kecerdasan buatan, dan pengembangan perangkat lunak umum. Keunggulan utama Python terletak pada ekosistem library yang luas, dukungan komunitas yang kuat, serta kemampuan untuk beradaptasi dengan berbagai paradigma pemrograman seperti pemrograman berorientasi objek dan fungsional. Namun, python juga memiliki kekurangan, seperti performa eksekusi yang lebih lambat dibandingkan bahasa pemrograman lain seperti C++, serta penggunaan memori yang relatif tinggi [14]. Meskipun demikian, Python tetap menjadi salah satu bahasa pemrograman yang paling populer dan banyak digunakan dalam berbagai aplikasi dan industri.

Gambar 2. 3 Logo Python

a) Streamlit

Streamlit adalah framework open-source yang dibuat untuk membantu orang membuat aplikasi web interaktif yang menggunakan bahasa pemrograman Python. Diluncurkan pada tahun 2018, Streamlit dimaksudkan untuk pengembang dan data scientist yang ingin membuat aplikasi web dengan cepat dan mudah tanpa harus menulis kode HTML, CSS, atau JavaScript. Menggunakan elemen seperti slider, tombol, dan grafik yang mudah diintegrasikan dengan pustaka data analitik seperti Pandas dan Matplotlib, Streamlit memungkinkan pengguna membuat antarmuka pengguna dengan menulis kode Python murni. Setiap kali kode diubah, Streamlit dapat memperbarui aplikasi secara real-time, yang memudahkan debugging dan iterasi pengembangan. Ini adalah keunggulan utama Streamlit. Selain itu, Streamlit mendukung integrasi dengan visualisasi data dan berbagai pustaka pembelajaran mesin, menjadikannya alat yang sangat bermanfaat untuk prototyping dan presentasi hasil analisis data.

b) Pandas

Pandas merupakan Library yang sangat berharga bagi para ilmuwan data. Sebagai Library open-source yang mendukung pembelajaran mesin, Pandas menawarkan struktur data yang fleksibel serta berbagai alat analisis tingkat tinggi. Dengan menggunakan Pandas, proses analisis, manipulasi, dan pembersihan data menjadi lebih efisien. Perpustakaan ini mendukung berbagai operasi, termasuk penyortiran, pengindeksan ulang, iterasi, penggabungan, konversi data, visualisasi, agregasi, dan banyak lagi.

c) Numpy

Numpy yang dikenal sebagai Python Numerical, adalah perpustakaan yang berfokus pada komputasi ilmiah. NumPy menyediakan berbagai fungsi yang siap digunakan untuk mendukung berbagai kebutuhan komputasi dalam penelitian, memungkinkan pengguna untuk melakukan operasi matematis dan analisis data dengan lebih mudah dan cepat.

d) Scikit-learn

Scikit-learn adalah salah satu library Python yang paling populer untuk machine learning dan analisis data. Dikembangkan dengan tujuan untuk menyediakan alat yang sederhana dan efisien untuk data mining dan analisis data, scikit-learn mencakup berbagai algoritma untuk klasifikasi, regresi, clustering, dan pengolahan data [15]. Salah satu keunggulan utama scikit-learn adalah kemudahan penggunaannya, berkat antarmuka yang konsisten dan dokumentasi yang sangat baik, yang memungkinkan pengguna untuk menerapkan algoritma machine learning dengan relatif cepat dan mudah [16].

Library ini juga dapat terintegrasi dengan baik dengan pustaka lain seperti NumPy, SciPy, dan Pandas, mempercepat proses pengembangan model dan analisis data [17]. Namun, scikit-learn juga memiliki beberapa kekurangan, termasuk keterbatasan dalam menangani dataset besar dan kompleks, karena pustaka ini tidak dirancang untuk komputasi skala besar seperti TensorFlow atau PyTorch [18].

Selain itu, beberapa algoritma yang lebih canggih atau terbaru mungkin tidak tersedia di scikit-learn, yang dapat membatasi kemampuan analisis bagi pengguna yang memerlukan teknik-teknik machine learning terkini

e) Seaborn

Sebuah library visualisasi data yang dibangun di atas matplotlib. Pustaka ini menyediakan antarmuka yang lebih tinggi untuk menciptakan grafik statistik yang tidak hanya menarik tetapi juga informatif. Salah satu keunggulan Seaborn memiliki kemudahan bagi penggunaannya, yang memungkinkan pengguna untuk membuat plot yang kompleks dengan kode yang minimal. Selain itu, Seaborn dilengkapi dengan berbagai fungsi standar yang mendukung visualisasi data statistik yang umum dilakukan, serta menawarkan gaya bawaan yang estetis, sehingga plot yang dihasilkan terlihat lebih menarik. Library ini juga menyediakan fungsi untuk mengubah tema, palet warna, dan gaya plot lainnya, memberikan fleksibilitas dalam presentasi data.

f) Matplotlib

Salah satu pustaka Python yang paling komprehensif dan banyak digunakan untuk membuat visualisasi data. Pustaka ini memiliki kemampuan

untuk menghasilkan grafik dalam bentuk statis maupun interaktif, serta mendukung pembuatan visualisasi dalam format dua dimensi (2D) dan tiga dimensi (3D). Matplotlib pertama kali diperkenalkan pada tahun 2003 oleh John D. Hunter, seorang ahli saraf dari Amerika Serikat, dengan tujuan awal untuk memungkinkan pengguna Python mereplikasi kemampuan pembuatan plot yang tersedia di aplikasi MATLAB, yang sudah dikenal luas di kalangan ilmuwan dan insinyur.

Salah satu keunggulan utama Matplotlib adalah fleksibilitasnya dalam memvisualisasikan berbagai jenis data. Pustaka ini memungkinkan pengguna untuk membuat berbagai jenis grafik, termasuk tetapi tidak terbatas pada plot garis, plot batang, histogram, scatter plot, dan box plot. Dengan fitur-fitur ini, Matplotlib menjadi alat yang sangat berguna dalam analisis data, karena dapat membantu peneliti dan ilmuwan data untuk menyajikan informasi dengan cara yang lebih jelas dan mudah dipahami.

Matplotlib juga menawarkan berbagai opsi kustomisasi, yang memungkinkan pengguna untuk mengubah elemen-elemen visual dari grafik, seperti warna, ukuran, dan gaya garis, serta menambahkan label, judul, dan anotasi. Hal ini memberikan kebebasan kepada pengguna untuk menyesuaikan visualisasi sesuai dengan kebutuhan presentasi mereka. Selain itu, Matplotlib dapat diintegrasikan dengan pustaka lain seperti NumPy dan Pandas, yang memudahkan pengguna dalam mengelola dan memproses data sebelum divisualisasikan.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Metode penelitian adalah rangkaian prosedur sistematis yang digunakan oleh peneliti untuk mengumpulkan, menganalisis, dan menginterpretasi data guna menjawab pertanyaan penelitian atau menguji hipotesis. Pada penelitian ini, metode yang digunakan adalah data mining. Data mining adalah proses eksplorasi data dalam jumlah besar untuk menemukan pola atau informasi tersembunyi dengan menggunakan teknik statistik, matematika, dan algoritma komputer.

Keberhasilan dan kegagalan proyek skripsi sangat dipengaruhi oleh metode penelitian yang digunakan. Metode penelitian yang sesuai akan memastikan data yang diperoleh relevan dan valid, serta analisis yang dilakukan dapat secara akurat menjawab pertanyaan penelitian. Sebaliknya, metode penelitian yang kurang tepat dapat menghasilkan data yang tidak akurat, analisis yang bias, dan kesimpulan yang tidak valid.

3.2 Alur Penelitian

Alur penelitian adalah proses sistematis yang melibatkan serangkaian langkah yang harus dilakukan secara berurutan untuk memastikan bahwa penelitian dilaksanakan dengan metodologi yang tepat dan menghasilkan temuan yang valid dan dapat diandalkan. Proses ini dimulai dari identifikasi masalah hingga penyusunan laporan akhir, dan setiap tahapan memiliki perannya masing-masing dalam membentuk keseluruhan penelitian. Berikut adalah penjelasan rinci mengenai tahapan alur penelitian :

1. Identifikasi Masalah

   Pada tahap ini, peneliti melakukan pencarian untuk menemukan masalah yang akan diteliti. Setelah masalah diidentifikasi, peneliti akan mencari solusi untuk permasalahan tersebut. Fokus utama dari tahap ini adalah bagaimana mengimplementasikan algoritma K-Medoids untuk mengelompokkan daerah berdasarkan tingkat pendapatan pajak.

2. Studi Literatur

   Peneliti melakukan pencarian terhadap literatur yang relevan dengan tujuan penelitian untuk menentukan kontribusi yang dapat diberikan oleh penelitian ini. Peneliti juga mencari informasi yang berguna dari berbagai sumber, termasuk buku, artikel akademis, dan sumber online.

3. Rancangan Penelitian

   Pada tahap ini, peneliti menyusun rencana penelitian yang komprehensif, mencakup langkah-langkah yang akan diambil selama penelitian. Rancangan ini mencakup aspek-aspek seperti jadwal penelitian, teknik pengumpulan data, dan metode pengolahan data.

4. Pengumpulan Data

   Di tahap ini, peneliti mengumpulkan data yang berkaitan dengan topik penelitian. Data yang dikumpulkan berasal dari observasi langsung dan penelusuran literatur yang relevan.

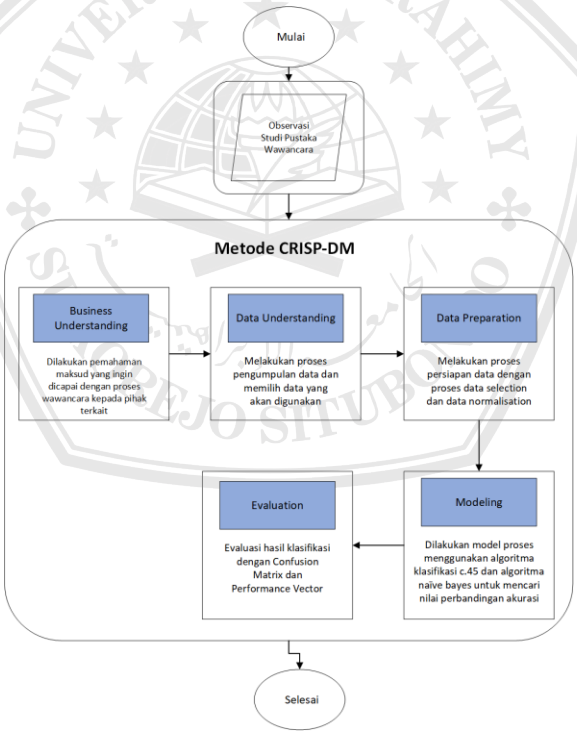
5. Pengolahan Data

   Setelah data terkumpul, peneliti melakukan pengolahan data menggunakan algoritma C4.5 dan Naïve Bayes. Proses ini melibatkan analisis

dan perhitungan untuk mendapatkan hasil yang valid akurat dan sesuai yang diinginkan.

6. Hasil dan Kesimpulan

Setelah menyelesaikan berbagai tahapan penelitian, peneliti merangkum hasil yang diperoleh dan mengevaluasi kekuatan serta kelemahan dari penelitian yang dilakukan. Kesimpulan yang dihasilkan harus sesuai dengan tujuan penelitian dan menjawab pertanyaan yang diajukan di bab pendahuluan. Selain itu, peneliti juga memberikan saran untuk penelitian selanjutnya agar dapat dilakukan dengan lebih baik..



Gambar 3. 1 Alur Penelitian

3.2.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang diperlukan dalam penelitian. Berdasarkan kebutuhan terkait masalah yang diteliti, berikut

adalah teknik-teknik pengumpulan data yang digunakan seperti yang tersaji pada pada penelitian ini :

a. Studi Pustaka (Library Research)

Peneliti mencari referensi melalui penelitian literatur atau buku-buku yang relevan dengan topik penelitian sebagai dasar pemikiran atau teori, yang dilakukan di perpustakaan dan sumber media lainnya yang meliputi studi pustaka seperti pada tinjauan penelitian sebelumnya di Bab 2 serta daftar pustaka dan sitasi yang telah tersaji pada penelitian ini.

b. Metode Penelitian Lapangan (Field Research)

Dalam konteks ini, penulis memanfaatkan Field Research yang telah dilakukan oleh peneliti lain dan disediakan sebagai dataset publik di situs *UCI Machine Learning Repository* dengan judul *"Early Stage Diabetes Risk Prediction"*. Data tersebut dikumpulkan melalui kuesioner yang diisi langsung oleh pasien di Rumah Sakit Diabetes Sylhet, Sylhet, Bangladesh, dan telah diverifikasi oleh seorang dokter. Dapat diakses pada link berikut https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset.

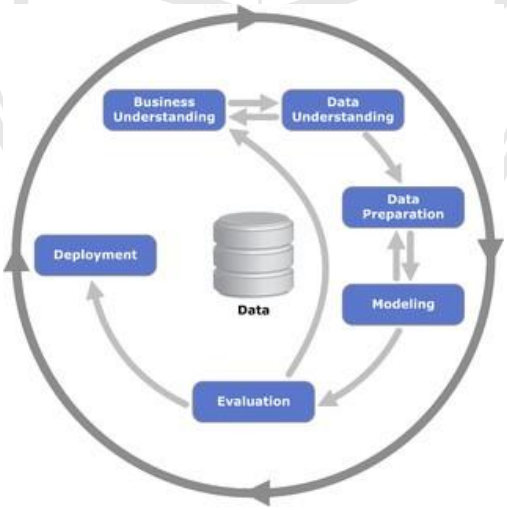
c. Metode Penelitian Exsperimen

Eksperimen dalam penelitian ini dirancang untuk membandingkan kinerja dua algoritma klasifikasi, yaitu C4.5 dan Naive Bayes, dalam memprediksi risiko diabetes berdasarkan dataset medis. Algoritma C4.5, yang merupakan pengembangan dari algoritma ID3, menggunakan pohon keputusan untuk membagi data berdasarkan fitur-fitur yang paling informatif, sedangkan

Naive Bayes merupakan metode probabilistik yang menerapkan prinsip teorema Bayes dengan asumsi independensi antar fitur. Proses eksperimen melibatkan pembagian dataset menjadi subset pelatihan dan pengujian, di mana algoritma C4.5 dan Naive Bayes dilatih pada subset pelatihan dan dievaluasi pada subset pengujian. Kinerja kedua algoritma diukur berdasarkan metrik akurasi, presisi, recall, dan F1-scoree untuk menilai efektivitas mereka dalam klasifikasi diabetes. Selain itu, analisis sensitivitas dilakukan untuk memahami bagaimana perubahan parameter mempengaruhi hasil model.

3.3 Metode Penelitian (CRISP-DM)

Penelitian ini dilaksanakan dengan menerapkan metode *Cross Industry Standard Process Model for Data Mining* (CRISP-DM). Berikut adalah penjelasan mengenai tahapan yang terjadi dalam CRISP-DM:



Gambar 3. 2 Implementasi CRISP-DM Methodology

1. Pemahaman Bisnis *(Business Understanding)*

   Pada tahap ini, fokus utama adalah memahami tujuan bisnis yang ingin dicapai dan bagaimana analisis data dapat membantu dalam mencapainya.

Langkah-langkah yang dilakukan termasuk menetapkan tujuan proyek, mengidentifikasi masalah yang ada, dan merumuskan pertanyaan bisnis yang spesifik.

2. Pemahaman Data *(Data Understanding)*

Di tahap ini, data yang relevan dikumpulkan dan dieksplorasi untuk mendapatkan pemahaman yang lebih baik tentang data tersebut, kualitasnya, serta hubungan data dengan pertanyaan bisnis yang telah ditetapkan.

3. Persiapan Data *(Data Preparation)*

Tahap ini berfokus pada persiapan data yang diperlukan untuk analisis lebih lanjut. Proses ini melibatkan pembersihan data, pemilihan atribut yang relevan, transformasi data, dan pengorganisasian data agar siap digunakan dalam analisis.

4. Pemodelan *(Modeling)*

Dalam fase ini, hasil dari proses persiapan data digunakan untuk menjalankan tahap pemodelan. Ini melibatkan pemilihan model yang tepat dan penerapan teknik data mining, termasuk algoritma dan alat yang akan digunakan.

a. Algoritma C4.5

1) Preprocessing Data

a) Pembersihan Data: Menangani nilai yang hilang dan outlier.

b) Normalisasi: Menstandarisasi fitur numerik jika diperlukan.

2) Pembagian Dataset

Pembagian Data: Membagi dataset menjadi subset pelatihan dan pengujian (misalnya, 80:20).

3) Pembangunan Pohon Keputusan

Pemilihan Fitur: Menggunakan Gain Ratio untuk memilih fitur yang membagi data secara optimal.

4) Pembentukan Node:

Membuat pohon keputusan berdasarkan pembagian fitur hingga semua data dalam node homogen atau tidak ada fitur lagi untuk dibagi.

5) Pruning: Mengurangi ukuran pohon untuk menghindari overfitting.

b. Algoritma Naive Bayes

1) Preprocessing Data

a) Pembersihan Data: Menangani nilai yang hilang dan outlier.

b) Normalisasi: Normalisasi fitur numerik jika diperlukan.

2) Pembagian Dataset

Pembagian Data: Membagi dataset menjadi subset pelatihan dan pengujian (misalnya, 80:20).

3) Perhitungan Probabilitas

Estimasi Probabilitas: Menghitung probabilitas kondisional dan prior untuk setiap fitur dan kelas berdasarkan data pelatihan.

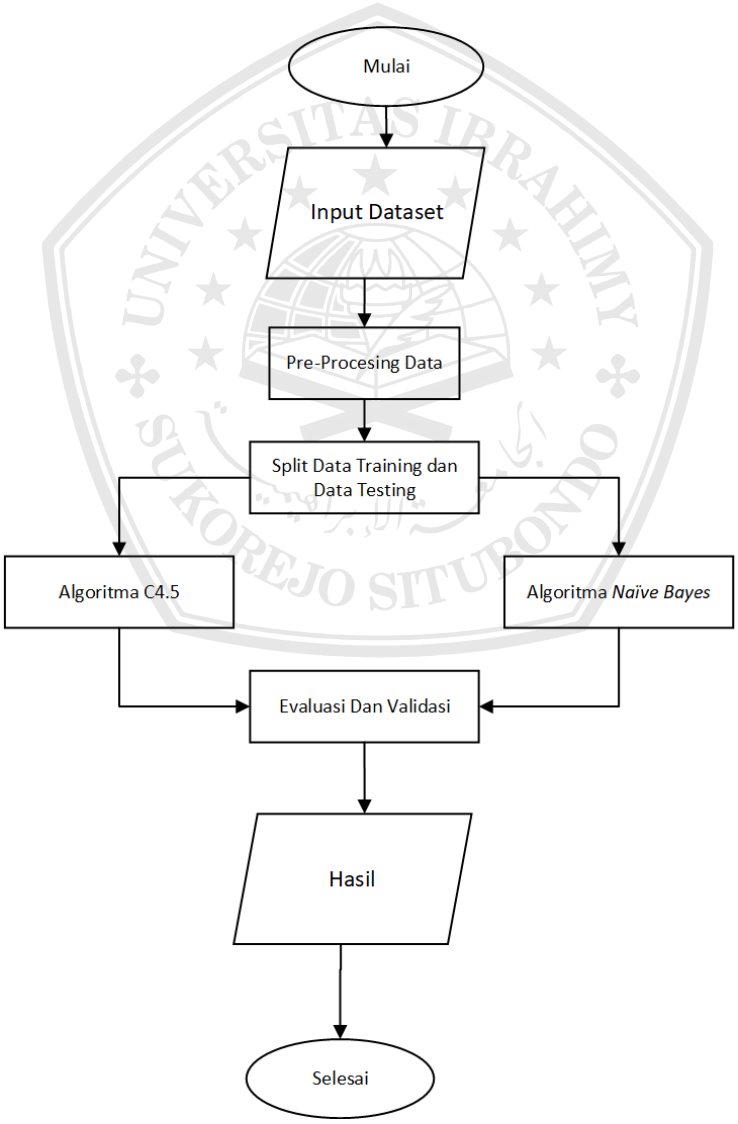
4) Prediksi:

Menggunakan Teorema Bayes untuk menghitung probabilitas posterior dan mengklasifikasikan instance ke kelas dengan probabilitas tertinggi.

5) Evaluasi (Evaluation)

Setelah model dibangun, tahap evaluasi dilakukan untuk menilai seberapa baik model tersebut memenuhi tujuan bisnis yang telah ditetapkan. Seperti mengukur kinerja model C4.5 dan Naive Bayes pada subset pengujian dengan menghitung akurasi, presisi, recall, dan F1-score.

3.3.1 Implementasi Metode



Gambar 3. 3 Implementasi Metode

3.4 Data Penelitian

Penentuan atribut pada data merupakan langkah penting dalam proses penelitian yang berkaitan dengan pemilihan variabel-variabel yang akan digunakan dalam analisis data. Atribut, atau variabel, merupakan elemen yang mendeskripsikan karakteristik data dan mempengaruhi hasil analisis. Data terdiri dari 520 baris dengan 17 variabel, yang mencakup 16 atribut dan 1 kelas.

Atribut yang digunakan dalam memprediksi tingkat Kesehatan mental siswa di pelajaran matematika adalah sebagai berikut:

1. ***Age***

   Atribut age adalah variabel yang mewakili usia individu dalam dataset penelitian ini. Atribut ini biasanya digunakan untuk menunjukkan usia seseorang dalam satuan tahun dan dapat mempengaruhi analisis serta hasil penelitian.

Tabel 3. 1 Atribut *Age*

No	Age	Jumlah
1.	*16*	1
2.	*25*	2
3.	*26*	1
4.	*27*	6
5.	*28*	9
6.	*29*	1
7.	*30*	25
8.	*31*	3
9.	*32*	5
10.	*33*	4
11.	*34*	6

Tabel 3.1 Lanjutan

No	Age	Jumlah
12.	*35*	30
13.	*36*	8
14.	*37*	7
15.	*38*	20
16.	*39*	16
17.	*40*	24
18.	*41*	4
19.	*42*	9
20.	*43*	25
21.	*44*	7
22.	*45*	18
23.	*46*	8
24.	*47*	21
25.	*48*	28
26.	*49*	7
27.	*50*	18
28.	*51*	5
29.	*52*	4
30.	*53*	20
31.	*54*	16
32.	*55*	22
33.	*56*	8
34.	*57*	15
35.	*58*	18
36.	*59*	4
37.	*60*	15
38.	*61*	8
39.	*62*	7

Tabel 3.1 Lanjutan

No.	Age	Jumlah
40.	*63*	3
41.	*64*	5
42.	*65*	6
43.	*66*	9
44.	*67*	8
45.	*68*	10
46.	*69*	5
47.	*70*	5
48.	*72*	9
49.	*79*	1
50.	*85*	2
51.	*90*	2
TOTAL		520

2. *Gender*

Atribut Gender merupakan salah satu variabel dalam dataset yang menggambarkan jenis kelamin Laki-laki dan wanita seperti pada tabel berikut :

Tabel 3. 2 Atribut *Gender*

No	Status	Jumlah
1	Male	328
2	Female	192

3. *Polyuria*

Atribut Polyuria merupakan salah satu variabel penting dalam dataset yang menggambarkan secara sederhana, poliuria adalah sering buang air kecil dengan volume urine yang tinggi seperti pada tabel berikut :

Tabel 3. 3 Atribut *Polyuria*

No	Status	Jumlah
1	Yes	258
2	No	262

4. *Polydipsia*

Polidipsia adalah istilah medis yang merujuk pada rasa haus yang berlebihan dan terus-menerus. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 4 Atribut *Polydipsia*

No	Status	Jumlah
1	Yes	233
2	No	287

5. *Sudden Weight Loss*

Sudden Weight Loss adalah kondisi penurunan berat badan secara tiba-tiba. Kondisi ini terjadi ketika seseorang mengalami penurunan berat badan yang signifikan dalam waktu yang relatif singkat tanpa adanya upaya diet atau olahraga yang disengaja. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 5 Atribut *Sudden Weight Loss*

No	Status	Jumlah
1	Yes	217
2	No	303

6. *Weakness*

Weakness adalah kondisi di mana seseorang merasa tidak bertenaga, lelah, dan sulit melakukan aktivitas sehari-hari. Kondisi ini bisa disebabkan oleh berbagai factor. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 6 Atribut *Weakness*

No	Status	Jumlah
1	Yes	305
2	No	215

7. *Polyphagia*

Polyphagia adalah istilah medis yang merujuk pada nafsu makan yang berlebihan atau rasa lapar yang terus-menerus. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut .

Tabel 3. 7 Atribut *Polyphagia*

No	Status	Jumlah
1	Yes	237
2	No	283

8. *Genital thrush*

Atribut ini menunjukkan kondisi di mana seorang mengalami infeksi jamur yang terjadi di area genital, baik pada pria maupun wanita. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut

Tabel 3. 8 Atribut *Genital Thrush*

No	Status	Jumlah
1	Yes	116
2	No	404

9. *Visual Blurring*

Visual blurring adalah kondisi di mana penglihatan menjadi kabur atau tidak jelas. Ini bisa terjadi pada satu atau kedua mata. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 9 Atribut *Visual Blurring*

No	Status	Jumlah
1	Yes	233
2	No	287

10. *Itching*

Atribut ini menjelaskan kondisi di mana orang mengalami sensasi tidak menyenangkan pada kulit yang menimbulkan keinginan untuk menggaruk. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 10 Atribut *Itching*

No	Status	Jumlah
1	Yes	253
2	No	267

11. *Irritability*

Atribut ini menjelaskan bahwa, Irritability adalah kondisi di mana seseorang cenderung mudah marah, jengkel, atau tersinggung. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 11 Atribut *Irritability*

No	Status	Jumlah
1	Yes	126
2	No	394

12. *Delayed Healing*

Pengertian Atribut Delayed Healing Delayed healing atau penyembuhan yang tertunda adalah kondisi di mana luka atau jaringan yang rusak membutuhkan waktu lebih lama dari biasanya untuk sembuh sempurna. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 12 Atribut *Delayed Healing*

No	Status	Jumlah
1	Yes	239
2	No	281

13. *Partial Peresis*

Atribut ini menjelaskan bahwa, Partial paresis adalah kondisi di mana kekuatan otot mengalami penurunan sebagian. Ini berarti seseorang masih dapat menggerakkan otot-otot yang terkena, tetapi kekuatannya berkurang dibandingkan dengan kondisi normal. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 13 Atribut *Partial Peresis*

No	Status	Jumlah
1	Yes	224
2	No	296

14. *Muscle Stiffness*

Atribut ini menjelaskan bahwa, Muscle stiffness adalah kondisi di mana otot terasa kaku, tegang, dan sulit digerakkan. Ini sering kali disertai dengan rasa nyeri atau ketidaknyamanan. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 14 Atribut *Muscle Stiffness*

No	Status	Jumlah
1	Yes	195
2	No	325

15. *Alopecia*

Atribut ini menjelaskan bahwa, Alopecia adalah istilah medis untuk kebotakan atau kerontokan rambut. Ini terjadi ketika jumlah rambut yang rontok lebih banyak dari pada rambut yang tumbuh.. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 15 Atribut *Alopecia*

No	Status	Jumlah
1	Yes	179
2	No	341

16. *Obesity*

Atribut ini menjelaskan bahwa, Obesity adalah kondisi medis di mana seseorang memiliki kelebihan lemak tubuh yang berlebihan sehingga dapat mengganggu kesehatan. Terdapat 2 kondisi Yes atau No seperti pada tabel berikut :

Tabel 3. 16 Atribut *Obesity*

No	Status	Jumlah
1	Yes	88
2	No	432

17. *Class*

Class dalam konteks dataset ini merujuk pada kategori atau label yang diberikan untuk hasil diagnosis terkait gejala diabetes. Kategori ini sangat penting dalam analisis data, karena membantu dalam mengklasifikasikan

individu berdasarkan kondisi kesehatan mereka. Dalam dataset ini, terdapat dua status yang mungkin, yaitu Positive, yang menunjukkan adanya diabetes, dan Negative, yang menunjukkan tidak adanya diabetes, seperti pada table berikut :

Tabel 3. 17 Atribut *Class*

No	Status	Jumlah
1	Positive	320
2	Negative	200

3.5 Spesifikasi Perangkat Penelitian

Spesifikasi perangkat penelitian dalam skripsi ini mencakup:

a. Kebutuhan Perangkat Keras (Hardware)
   1. Laptop: HP Pavilion X360 Convertible
   2. Processor: Intel® Core™ i3- 8145U
   3. Memory: 8GB SO-DIMM DDR4-2666
   4. Storage: 256 GB SSD, 500 GB HDD
   5. OS: Windows 10

b. Kebutuhan Perangkat Lunak (Software)
   1. Visual Studio Code
   2. Python
   3. Microsoft Excel
   4. Rapidminer Studio

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pemahaman Bisnis *(Business Understanding)*

Dataset yang digunakan dalam penelitian ini diambil dari *UCI Machine Learning Repository*, mencakup informasi yang berisi data tanda dan gejala pasien baru atau calon pasien diabetes.

4.2 Pemahaman Data *(Data Understanding)*

Dataset diabetes yang dianalisis adalah data dari *UCI Machine Learning Repository* tahun 2019, yang mencakup berbagai jenis *variable* seperti pada *list* berikut:

1. Umur (Age) 1.20-65
2. Jenis Kelamis (Sex) 1. Male, 2.Female
3. Sering buang air kecil (Polyuria) 1.Yes, 2.No.
4. Haus berlebihan (Polydipsia) 1.Yes, 2.No.
5. Penurunan berat badan tiba-tiba (sudden weight loss) 1.Yes, 2.No.
6. Lemas/Lelah (weakness) 1.Yes, 2.No.
7. Nafsu makan berlebih (Polyphagia) 1.Yes, 2.No.
8. Sariawan Genital (Genital thrush) 1.Yes, 2.No.
9. Penglihatan Kabur (visual blurring) 1.Yes, 2.No.
10. Gatal (Itching) 1.Yes, 2.No.
11. Mudah marah (Irritability) 1.Yes, 2.No.
12. Penyembuhan yang tertunda (delayed healing) 1.Yes, 2.No.
13. Penurunan kekuatan otot (partial paresis) 1.Yes, 2.No.

14. Kekakuan pada otot (muscle stifness) 1.Yes, 2.No.
15. Kerontokan Rambut (Alopecia) 1.Yes, 2.No.
16. Kelebihan berat badan (Obesity) 1.Yes, 2.No.
17. Class 1.Positive, 2.Negative.

4.3 Persiapan Data *(Data Preparation)*

Pada tahap ini, penulis melaksanakan serangkaian langkah untuk menyiapkan data agar siap digunakan dalam analisis. Persiapan data adalah langkah krusial yang mencakup:

1. Pemilihan Data

   Memilih data yang relevan dari berbagai sumber untuk analisis. Data diabetes yang dipilih merupakan dataset publik yang sudah sesuai dengan kriteria yang telah ditetapkan sebelumnya.

2. Transformasi Data

   Mengubah format variable parameter pada atribut data ke dalam numerik agar sesuai dengan kebutuhan analisis.

3. Pengkodean:

   Dalam penelitian ini, penulis menggunakan perangkat lunak Visual Studio Code sebagai alat utama untuk melakukan analisis dan perhitungan data. Visual Studio Code dipilih karena kemampuannya yang fleksibel dan dukungan yang luas terhadap berbagai bahasa pemrograman, termasuk Python, yang digunakan dalam penelitian ini. Penulis mengimpor berbagai library yang diperlukan, seperti Pandas untuk pengolahan data, NumPy untuk perhitungan numerik, dan Matplotlib untuk visualisasi data dan library lainnya.

Segmen Program 4. 1 *Import Library Python*

```
1.  : import streamlit as st
2.  : import pandas as pd
3.  : from sklearn.model_selection
      import train_test_split
5.  : from sklearn.preprocessing import
      LabelEncoder
6.  : from sklearn.tree import
      DecisionTreeClassifier,plot_tree
7.  : from sklearn.naive_bayes import GaussianNB
8.  : from sklearn.metrics import
      accuracy_score, confusion_matrix,
      classification_report
9.  : import seaborn as sns
10. : import matplotlib.pyplot as plt
```

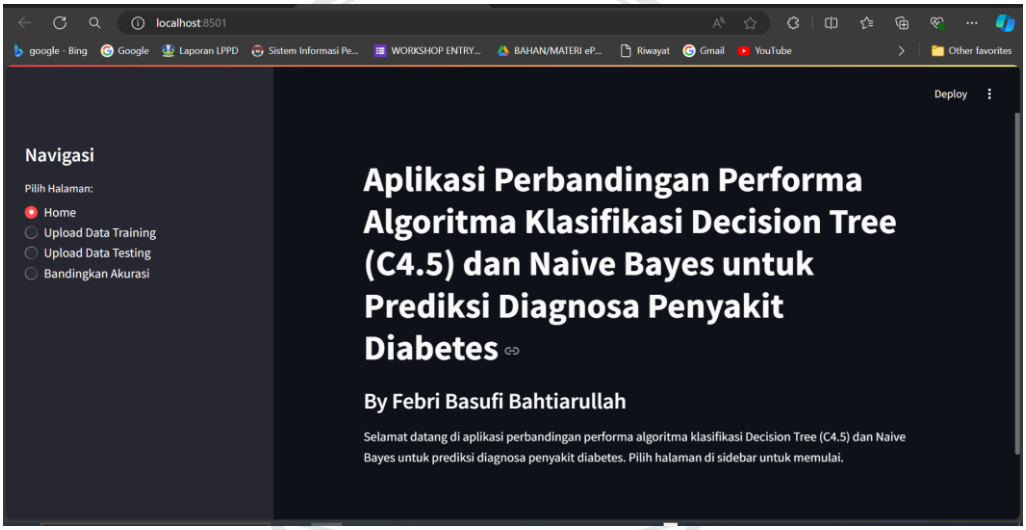
Segmen Program 4.1 mencakup proses penting dalam analisis data dengan memanfaatkan berbagai library yang diperlukan untuk manipulasi, pemrosesan, dan visualisasi data. Library yang diimpor meliputi Pandas, yang digunakan untuk manipulasi data dan pembersihan dataset; Scikit-learn (sklearn) untuk pemrosesan data, pengembangan model pembelajaran mesin, serta evaluasi model; dan Seaborn serta Matplotlib untuk visualisasi data yang lebih menarik, informatif, dan mudah dipahami. Dengan menggunakan library ini, data dapat diproses dengan efektif, mulai dari pembagian dataset menjadi set pelatihan dan pengujian, hingga pengkodean label untuk mempersiapkan data sebelum proses pelatihan model. Selanjutnya, model klasifikasi seperti pohon keputusan (Decision Tree) dan Naive Bayes dibangun dan dilatih. Kinerja model dievaluasi melalui berbagai metrik, termasuk akurasi, Confusion Matrix, precision, dan recall.

Setelah dilakukan import library yang dibutuhkan maka selanjutnya membuat title dan sidebar seperti pada segmen Program dibawah ini:

Segmen Program 4. 2 Navigasi *Sidebar* dan *Title*

```
1.  : # Title of the app
2.  : st.title('Aplikasi Perbandingan Performa
      Algoritma Klasifikasi Decision Tree (C4.5) dan Naive
      Bayes untuk Prediksi Diagnosa Penyakit Diabetes')
3.  : st.subheader('By Febri Basufi Bahtiarullah')
4.  : # Sidebar navigation
5.  : st.sidebar.title('Navigasi')
6.  : page = st.sidebar.radio('Pilih Halaman:',
      ['Home', 'Upload Data', 'Model Training & Evaluation',
      'Bandingkan Akurasi', 'Prediksi'])
```

Dari hasil penulisan Program tersebut maka akan ditampilkan GUI pada *web library streamlit* seperti gambar berikut :



Gambar 4. 1 Tampilan Navigasi *Sidebar* dan *Title*

4.2.1 Transformasi Data

Dengan menerapkan metode ini, kita dapat mengubah data kategori atau fitur non-numerik menjadi format yang sesuai untuk model pembelajaran mesin. Ini secara langsung dapat meningkatkan kinerja serta akurasi dalam analisis data selama proses data mining. Perubahan data kategori menjadi format numerik memungkinkan algoritma pembelajaran mesin bekerja lebih optimal dalam mendeteksi pola dan menghasilkan prediksi yang lebih akurat. Oleh karena itu,

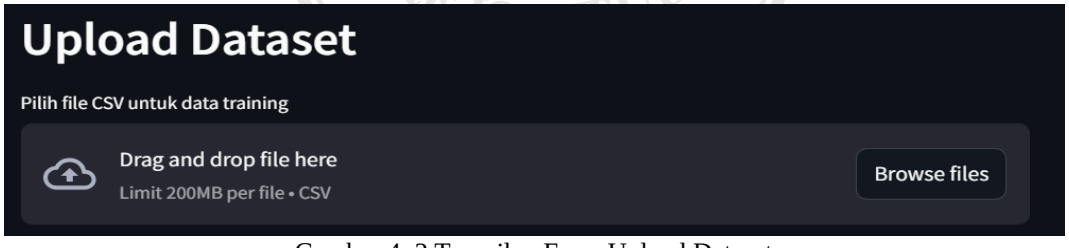
sebelum proses transformasi data ini dilakukan, sangat penting untuk mengunggah dataset ke dalam aplikasi.

Proses upload dataset ke aplikasi Streamlit menjadi langkah awal dalam pipeline data mining. Pengguna dapat mengunggah file dataset mereka, baik dalam format CSV maupun Excel, menggunakan antarmuka yang disediakan. Berikut adalah cuplikan segmen Program yang diperlukan untuk menangani proses ini:

Segmen Program 4. 3 Upload Data *Training*

```
1.  : elif page == 'Upload Data Training':
2.  :     st.header("Upload Dataset Training")
3.  :     uploaded_file = st.file_uploader("Pilih file
          CSV
          untuk data training", type="csv")
4.  :     if uploaded_file:
5.  :         try:
6.  :             # Read the dataset
7.  :             data_train =
                  pd.read_csv(uploaded_file)
8.  :             st.write("Dataset sebelum konversi ke
                  numerik:")
9.  :             st.write(data train.head())
```

Hasil dari penulisan Program tersebut akan menampilkan output tampilan pada aplikasi streamlit seperti berikut :



Gambar 4. 2 Tampilan Form Upload Dataset

Setelah proses upload dataset selesai, langkah berikutnya adalah mengonversi dataset ke tipe data numerik. Proses konversi ini dijelaskan dalam segmen program yang akan disajikan berikut ini:

Segmen Program 4. 4 Pendefinitrasian Fungsi *Convert Data Numeric*

```
1.  : # Function to convert categorical data to
      numeric
2.  : def convert_to_numeric(df):
3.  :    le = LabelEncoder()
4.  :    for column in df.columns:
5.  :        if df[column].dtype == object:
6.  :            df[column] =
                 le.fit_transform(df[column].astype(str))
7.  :    return df
```

1. Pendefinisian Fungsi:
    a. def convert_to_numeric(df): mendefinisikan sebuah fungsi dengan nama convert_to_numeric, yang menerima satu parameter, yaitu df, yang merupakan DataFrame.
2. Inisialisasi LabelEncoder:
    a. le = LabelEncoder() membuat objek LabelEncoder yang digunakan untuk mengonversi data kategorikal menjadi angka.
    b. Iterasi pada Kolom DataFrame:
    c. for column in df.columns: melakukan iterasi (perulangan) pada setiap kolom dalam DataFrame df.
3. Mengecek Tipe Data Kolom: if df[column].dtype == object: memeriksa apakah tipe data dari kolom tersebut adalah object (biasanya menunjukkan bahwa kolom tersebut berisi data kategorikal atau string).
4. Mengonversi Data Kategorikal ke Numerik:
    a. df[column] = le.fit_transform(df[column].astype(str)) mengonversi kolom tersebut menjadi tipe string terlebih dahulu (jika diperlukan) dan kemudian menerapkan fit_transform untuk menggantinya dengan nilai numerik yang sesuai.

5. Mengembalikan DataFrame yang Sudah Dikonversi:
    a. return df mengembalikan DataFrame yang sudah diubah sehingga semua kolom kategorikal di dalamnya telah dikonversi menjadi numerik.

Selanjutnya setelah fungsi dibuat maka akan dijalan dan dipanggil untuk ditampilkan ke dalam aplikasi streamlit:

Segmen Program 4. 5 Transformasi Format Data ke *Numeric*

```
1.  : # Convert categorical data to numeric
2.  : data_train_numeric = data_train.copy()
3.  : data_train_numeric =
      convert_to_numeric(data_train_numeric)
4.  : st.write("Dataset setelah konversi ke
      numerik:")
5.  : st.write(data_train_numeric.head())
6.  : # Store the training dataset in session state
7.  : st.session_state['data_train'] =
      data_train_numeric
```

Penjelasan pada segmen Program 4.4 adalah sebagai berikut :

1. Inisialisasi LabelEncoder: LabelEncoder digunakan untuk mengubah nilai kategori menjadi angka unik.
2. Iterasi melalui kolom DataFrame: Fungsi ini memeriksa setiap kolom di DataFrame. Jika kolom tersebut bertipe data object (yang biasanya menunjukkan data kategori), maka kolom tersebut diubah menjadi numerik.
3. Transformasi dan Penggantian: Kolom yang dikonversi diubah menjadi angka menggunakan LabelEncoder, dan hasilnya menggantikan kolom yang asli di DataFrame.
4. Mengembalikan DataFrame yang telah diubah: DataFrame dengan kolom kategori yang telah dikonversi ke format numerik dikembalikan sebagai output fungsi.

5. Integrasi dengan Streamlit: Penggunaan st.write: Setelah konversi, DataFrame yang telah dimodifikasi ditampilkan di aplikasi Streamlit menggunakan st.write. Ini memungkinkan pengguna untuk melihat pratinjau data setelah proses konversi

Output dari segmen program 4.5 dapat dilihat sebagai berikut :

Dataset setelah konversi ke numerik:

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	vi
0	40	1	0	1	0	1	0	0	
1	58	1	0	0	0	1	0	0	
2	41	1	1	0	0	1	1	0	
3	45	1	0	0	1	1	1	1	
4	60	1	1	1	1	1	1	0	

Gambar 4. 3 Tampilan data setelah konversi ke Numerik

4.4 Pemodelan (Modeling)

Pemodelan dalam data mining sangat penting untuk menguji seberapa efektif kinerja model algoritma berdasarkan rencana implementasi yang sudah dirancang sebelumnya. Pemodelan dilakukan menggunakan bahasa pemrograman python dan text editor VS Code yang nantinya akan menampilkan ke dalam Web menggunakan library streamlit python.

4.4.1 Pemilihan Fitur dan Penetapan Label Dataset

Setelah data dilakukan Transformasi maka tahapan selanjutnya adalah melakukan pemilihan penetapan fitur dan label pada dataset. Pada pemrogman python dapat dilihat sebagai berikut : Setelah data dilakukan Transformasi maka tahapan selanjutnya adalah melakukan pemilihan penetapan fitur dan label pada dataset. Pada pemrogman python dapat dilihat sebagai berikut Setelah data

dilakukan Transformasi maka tahapan selanjutnya adalah melakukan pemilihan penetapan fitur dan label pada dataset. Pada pemrogman python dapat dilihat sebagai berikut : Setelah data dilakukan Transformasi maka tahapan selanjutnya adalah melakukan pemilihan penetapan fitur dan label pada dataset. Pada pemrogman python dapat dilihat sebagai :

Segmen Program 4. 6 Pemilihan Fitur dan Label

```
1.  : # Select features and target
2.  : features = st.multiselect('Pilih fitur',
      data_train_numeric.columns.tolist(),
      default=data_train_numeric.columns[:-1].tolist())
3.  : target = st.selectbox('Pilih target',
      data_train_numeric.columns.tolist(),
      index=len(data_train_numeric.columns)-1)
4.  : if features and target:
5.  :     X_train = data_train_numeric[features]
6.  :     y_train = data_train_numeric[target]
```

Penjelasan program :

a) Pilih Fitur (Select Features):

1. st.multiselect adalah komponen Streamlit yang memungkinkan pengguna untuk memilih beberapa opsi dari daftar.
2. data_train_numeric.columns.tolist() mengonversi daftar nama kolom dari DataFrame data_train_numeric menjadi list.
3. default=data_train_numeric.columns[:-1].tolist() menetapkan fitur default yang dipilih adalah semua kolom kecuali kolom terakhir. Ini berguna jika kolom terakhir adalah target.

b) Pilih Target (Select Target):

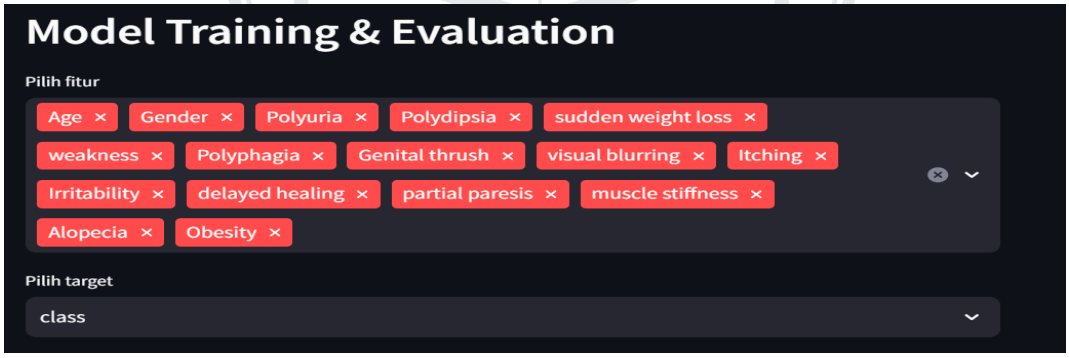
1. st.selectbox adalah komponen Streamlit yang memungkinkan pengguna untuk memilih satu opsi dari daftar.

2. data_train_numeric.columns.tolist() memberikan daftar nama kolom untuk dipilih.
3. index=len(data_train_numeric.columns)-1 menetapkan kolom terakhir sebagai default yang dipilih, biasanya kolom target.

c) Membuat Data Latih (Training Data):

1. Mengecek apakah fitur dan target telah dipilih. Jika iya, maka kode di dalam blok if akan dieksekusi.
2. X_train adalah DataFrame yang hanya mencakup kolom-kolom yang dipilih sebagai fitur.
3. y_train adalah Series yang hanya mencakup kolom yang dipilih sebagai target.

Dari penulisan hasil Program tersebut akan menghasilkan output :



Gambar 4. 4 Tampilan penentuan data fitur dan label

4.4.2 Membagi Data Training dan Data Testing

Pembagian data adalah langkah penting dalam proses pengembangan model prediksi. Dataset yang ada dipecah menjadi 2 file dengan data training 80% dan data testing 20% yang nantinya ouput dari data testing merupakan data testing tanpa label. Untuk segmen program pada fungsi ini dapat dilihat sebagai berikut :

Segmen Program 4. 7 Membagi Data *Training* dan Data *Testing*

```
1.  : import pandas as pd
2.  : from sklearn.model_selection import
      train_test_split
3.  : # Membaca dataset
4.  : df = pd.read_csv('diabetes_data_upload.csv')#
      ganti 'data.csv' dengan nama file dataset kamu
5.  : # Memisahkan fitur dan label
6.  : X = df.drop(columns='class')  # ganti 'label'
      dengan nama kolom target
7.  : y = df['class']  # ganti 'label' dengan nama
      kolom target
8.  : # Membagi data menjadi data training dan data
      testing : dengan rasio 80:20
9.  : X_train, X_test, y_train, y_test =
      train_test_split(X, y, test_size=0.2, random_state=42)
10. : # Menggabungkan kembali data training dan
      testing dengan labelnya
11. : train_data = pd.concat([X_train, y_train],
      axis=1)
12. : test_data = pd.concat([X_test, y_test], axis=1)
13. : # Menyimpan data ke file CSV
14. : train_data.to_csv('data_training.csv',
      index=False)
15. : test_data.to_csv('data_testing.csv',
      index=False)
16. : print("Data training dan data testing berhasil
      disimpan!")
```

Dalam konteks ini dataset yang sudah dilakukan Transformasi kemudian dilakukan pembagian data dengan rasio 80% untuk data training dan 20% untuk data testing yang kemudian data disimpan dengan file format csv.

4.4.3 Prediksi Klasifikasi Decision Tree C4.5

Dalam aplikasi prediksi penyakit diabetes ini, kita menggunakan algoritma dua algoritma salah satunya adalah pohon keputusan (Decision Tree) yang dikenal sebagai C4.5 untuk menganalisis data dan membuat prediksi. Berikut adalah cara mengevaluasi kinerja model pohon keputusan yang telah dilatih dan bagaimana hasilnya ditampilkan kepada pengguna menggunakan python dan library streamlit.

Segmen Program 4. 8 Melatih Model Decision Tree C4.5

```
1.  : # Decision Tree
2.  : st.subheader("Decision Tree (C4.5)")
3.  : dt_model = DecisionTreeClassifier()
4.  : dt_model.fit(X_train, y_train)
5.  : dt_predictions = dt_model.predict(X_test)
```

Penjelasan segmen Program 4.8 :

a. Subheader:

   st.subheader("Decision Tree (C4.5)") menampilkan sub judul "Decision Tree (C4.5)" di halaman aplikasi.

b. Inisialisasi dan Pelatihan Model:

   1. dt_model = DecisionTreeClassifier() membuat objek model Decision Tree dengan default parameter.
   2. dt_model.fit(X_train, y_train) melatih model dengan data training (X_train sebagai fitur dan y_train sebagai target).

c. Melakukan Prediksi:

   dt_predictions = dt_model.predict(X_test) menggunakan model yang sudah dilatih untuk melakukan prediksi pada data testing (X_test).

4.4.4 Klasifikasi Naïve Bayes

Setelah data dievaluasi menggunakan algoritma Decision Tree C4.5, langkah selanjutnya adalah melakukan pengujian pada Model Naive Bayes. Dalam proses ini, kita melatih model Naive Bayes dengan menggunakan data training. Model ini belajar untuk menghubungkan fitur-fitur yang ada dengan kelas target berdasarkan probabilitas, sehingga dapat meningkatkan akurasi dan efektivitas dalam melakukan klasifikasi data yang diberikan.

Segmen Program 4. 9 Melatih Model Pelatihan Naïve Bayes

```
1.  : # Naive Bayes
2.  : st.subheader("Naive Bayes")
3.  : nb_model = GaussianNB()
4.  : nb_model.fit(X_train, y_train)
5.  : nb_predictions = nb_model.predict(X_test)
```

a) nb_model = GaussianNB(): Membuat model Naive Bayes dengan distribusi Gaussian.

b) nb_model.fit(X_train, y_train): Melatih model menggunakan data pelatihan (X_train dan y_train).

c) nb_predictions = nb_model.predict(X_test): Memprediksi hasil dari data uji (X_test).

4.5 Evaluasi Model

Evaluasi model klasifikasi adalah langkah penting untuk memastikan performa model dalam membuat prediksi seperti mengukur akurasi, confusion matriks dan classification report seperti Precision, Recall dan F-1 Scoree.

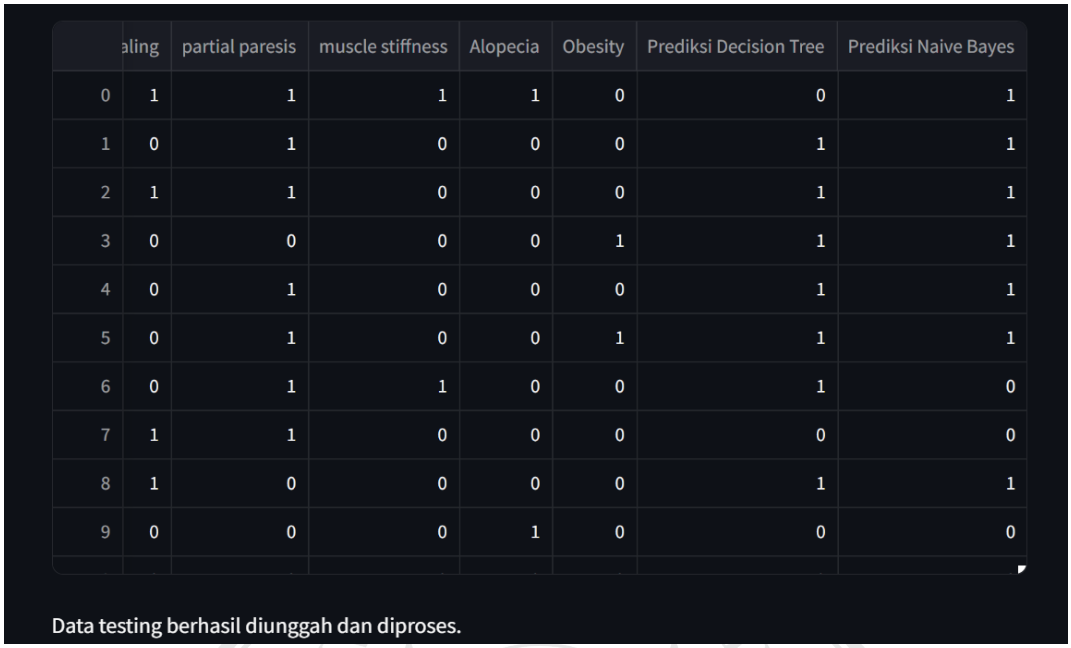
Segmen Program 4. 10 Prediksi Model Pelatihan dengan Data *Testing* yang di Unggah

```
1.  : # Upload Data Testing Page
2.  : elif page == 'Upload Data Testing':
3.  :       st.header("Upload Dataset Testing")
4.  :       if st.session_state['dt_model'] is not None
            and st.session_state['nb_model'] is not
            None:
5.  :        testing_file = st.file_uploader("Pilih
             file CSV untuk data testing", type="csv",
             key='test')
6.  : if testing_file:
7.  :             try:
8.  :                    # Read and preprocess testing
                         data
9.  :                    testing_data =
                         pd.read_csv(testing_file)
10. :                    testing_data =
                         convert_to_numeric(testing_data)
```

Segmen Program 4.10 Lanjutan

```
11. :                     # Get features for prediction
12. :                     features =
                          st.session_state['features']
13. :                     target =
                          st.session_state['target']
14. :                     # Check if the required features
                          are present in testing data
15. :                     if set(features).issubset
                          (testing_data.columns):
16. :                         # Adding columns for
                              predictions
17. :                         X_test = testing_data
                              [features]
18. :                         dt_predictions =
                              st.session_state['dt_model'].
                              predict(X_test)
19. :                         nb_predictions =
                              st.session_state['nb_model'].
                              predict(X_test)
20. :                         testing_data['Prediksi
                              Decision Tree'] =
                              dt_predictions
21. :                         testing_data['Prediksi Naive
                              Bayes'] = nb_predictions
22. :                         st.write("Hasil Prediksi:")
23. :                         st.write(testing_data)
24. :                         # Add target column for
                              comparison
25. :                         testing_data[target] =
                              testing_data['Prediksi
                              Decision Tree']
26. :                         # Store the testing dataset in
                              session state
27. # Store the testing dataset in
                              session state
28. :                         st.session_state['data_test']
                              = testing_data
29. :                         st.write("Data testing
                              berhasil diunggah dan
                              diproses.")
30. :                     else:
31. :                         st.error("Data testing tidak
                              memiliki fitur yang sesuai.")
32. :             except Exception as e:
33. :                 st.error(f"Error loading file:{e}")
34. : else:
35. :         st.error("Model belum dilatih. Harap
              unggah dan latih data training terlebih dahulu di
              halaman Upload Data Training.")
```

Output dari segmentasi Program 4.9 diatas adalah sebagai berikut :



	aling	partial paresis	muscle stiffness	Alopecia	Obesity	Prediksi Decision Tree	Prediksi Naive Bayes
0	1	1	1	1	0	0	1
1	0	1	0	0	0	1	1
2	1	1	0	0	0	1	1
3	0	0	0	0	1	1	1
4	0	1	0	0	0	1	1
5	0	1	0	0	1	1	1
6	0	1	1	0	0	1	0
7	1	1	0	0	0	0	0
8	1	0	0	0	0	1	1
9	0	0	0	1	0	0	0

Data testing berhasil diunggah dan diproses.

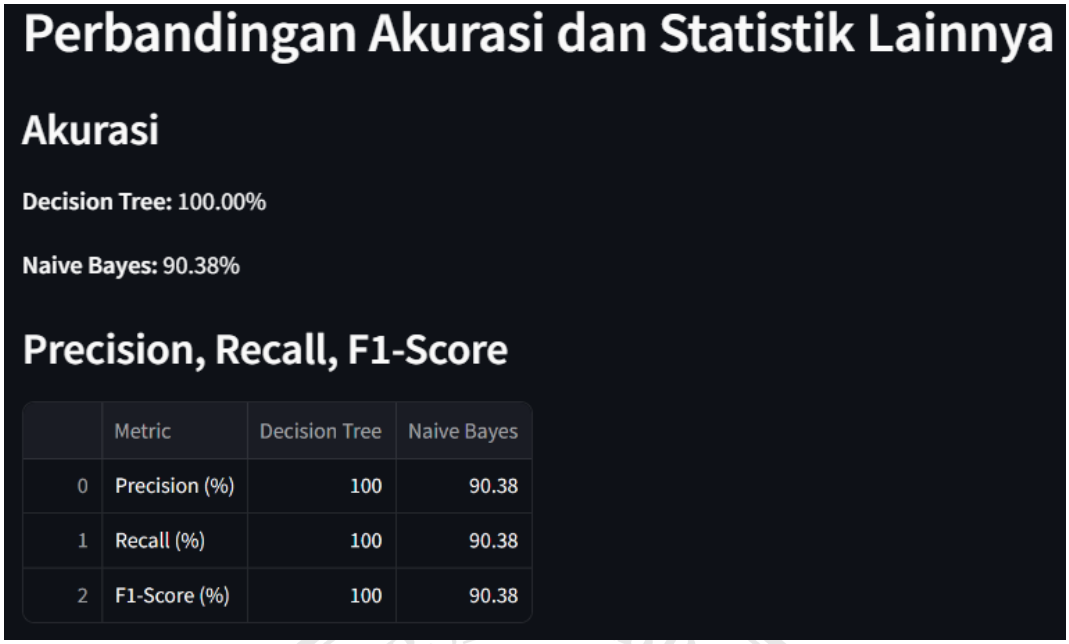
Gambar 4. 5 Hasil Prediksi kedua Model

Hasil prediksi kedua model algoritma Decision Tree C4.5 menunjukkan perbedaan dengan algoritma Naïve Bayes karena perbedaan metode klasifikasi yang digunakan. Algoritma C4.5 lebih mengutamakan pembentukan pohon keputusan berdasarkan entropy, sementara Naïve Bayes mengandalkan probabilitas bersyarat dengan asumsi independensi antar atribut.

4.5.1 Evaluasi Perbandingan Hasil Prediksi Model C4.5 dan Naïve Bayes

Pada penulisan Segmen Program 4.11, dilakukan perbandingan perhitungan performa kedua model algoritma, yaitu Decision Tree C4.5 dan Naïve Bayes, dengan tujuan untuk melihat perbedaan dalam akurasi, precision, recall, dan confusion matrix. Untuk penulisan segmen program 4.11 dapat dilihat pada **Lampiran A.1** Perbandingan Perhitungan Performa Kedua Model.

Kemudian Dari hasil penulisan segmen Program 4.11 maka akan dihasilkan tampilan output sebagai berikut :



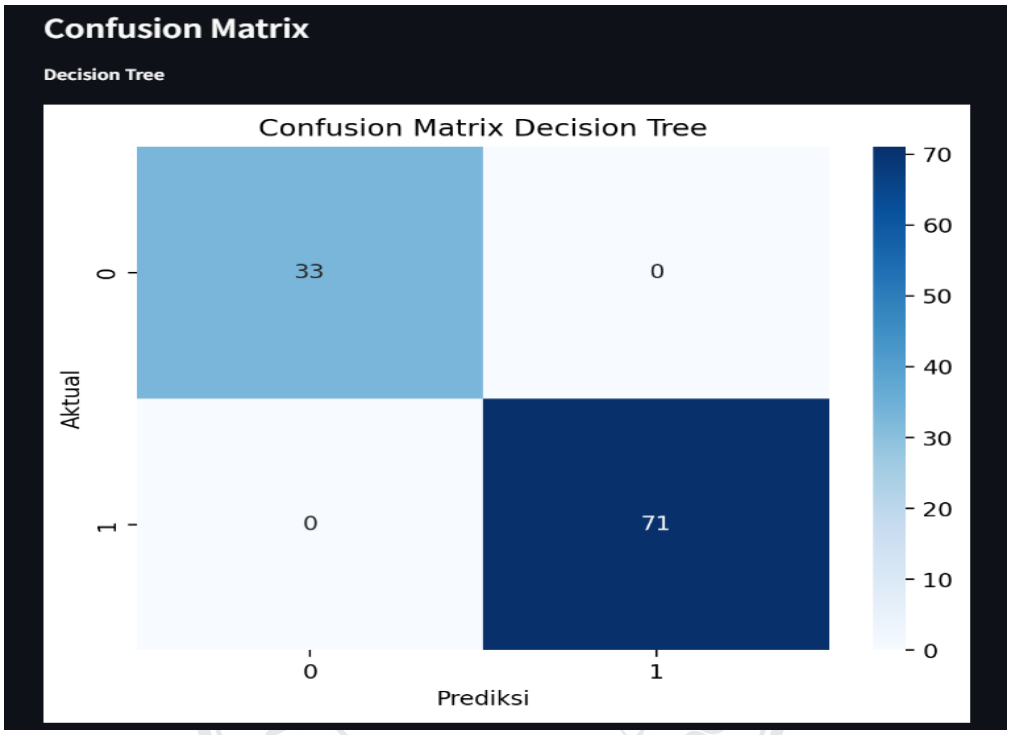
Gambar 4. 6 Perbandingan Akurasi C4.5 dan Naïve Bayes

Penjelasan gambar 4.6 dapat disimak sebagai berikut :

1. Precision (%)": Menghitung dan menampilkan precision (ketepatan) dari model dalam persentase. Precision adalah rasio prediksi benar positif terhadap total prediksi positif. Disini model membaca nilai Precision pada Decision Tree adalah sebesar 100 % sedangkan untuk Naïve Bayes 90,38%
2. "Recall (%)": Menghitung dan menampilkan recall (daya ingat) dari model dalam persentase. Recall adalah rasio prediksi benar positif terhadap total data positif sebenarnya. Disini model membaca nilai Recall pada Decision Tree adalah sebesar 100 %, namun untuk Naïve Bayes 90,38%
3. "F1-Scoree (%)": F1-Scoree adalah rata-rata harmonis dari precision dan recall, memberikan gambaran keseimbangan antara keduanya. Sedangkan untuk F-1Scoree model membaca nilai Recall model Decision Tree adalah sebesar 100 %, sedangkan untuk Naïve Bayes 90,38%

Kemudian untuk hasil ouput tampilan Confusion Matrix kedua model juga dapat dilihat sebagai berikut :

A. Decision Tree C4.5

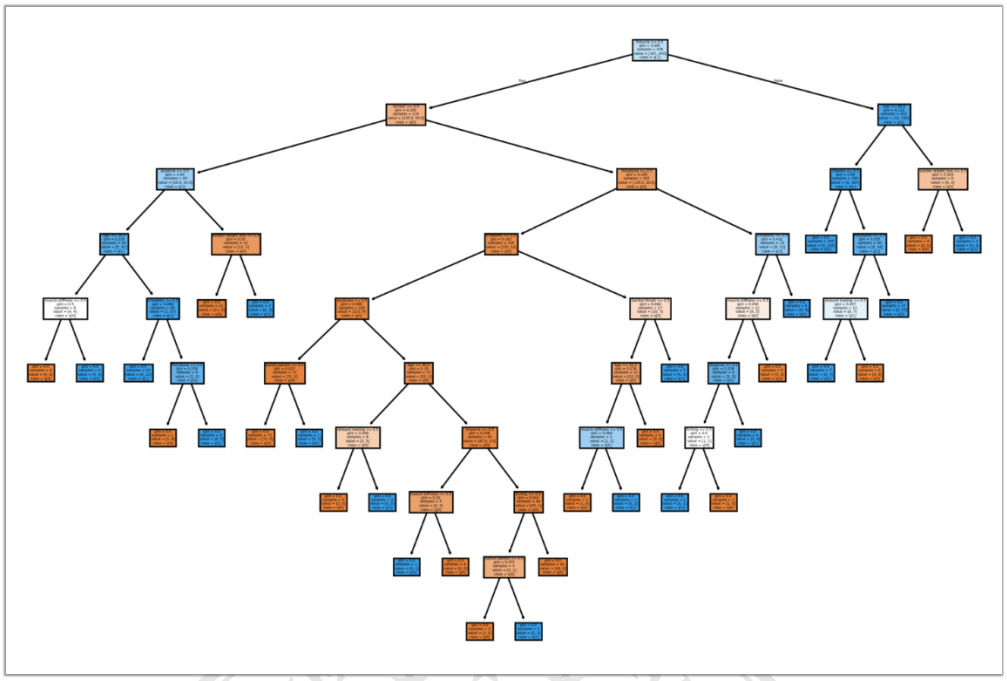


Gambar 4. 7 Menampilkan Hasil Akurasi dan Confusion Matriks Decision Tree C4.5

Pada pemodelan Decision Tree C4.5 yang sudah jalankan maka ditemukan akurasi Decision Tree C4.5 sebesar 100%. Sedangkan untuk nilai Confusion Matriks Decision Tree C4.5 dapat dilihat sebagai berikut :

1) True Positive (TP) : 71 Data positif yang diprediksi dengan benar.
2) True Negative (TN) : 33 Data negatif yang diprediksi dengan benar
3) False Positive (FP) : 0 Data negatif yang diprediksi sebagai positif.

Kemudian untuk visualisasi Pohon Keputusan node decision tree dapat dilihat sebagai berikut :



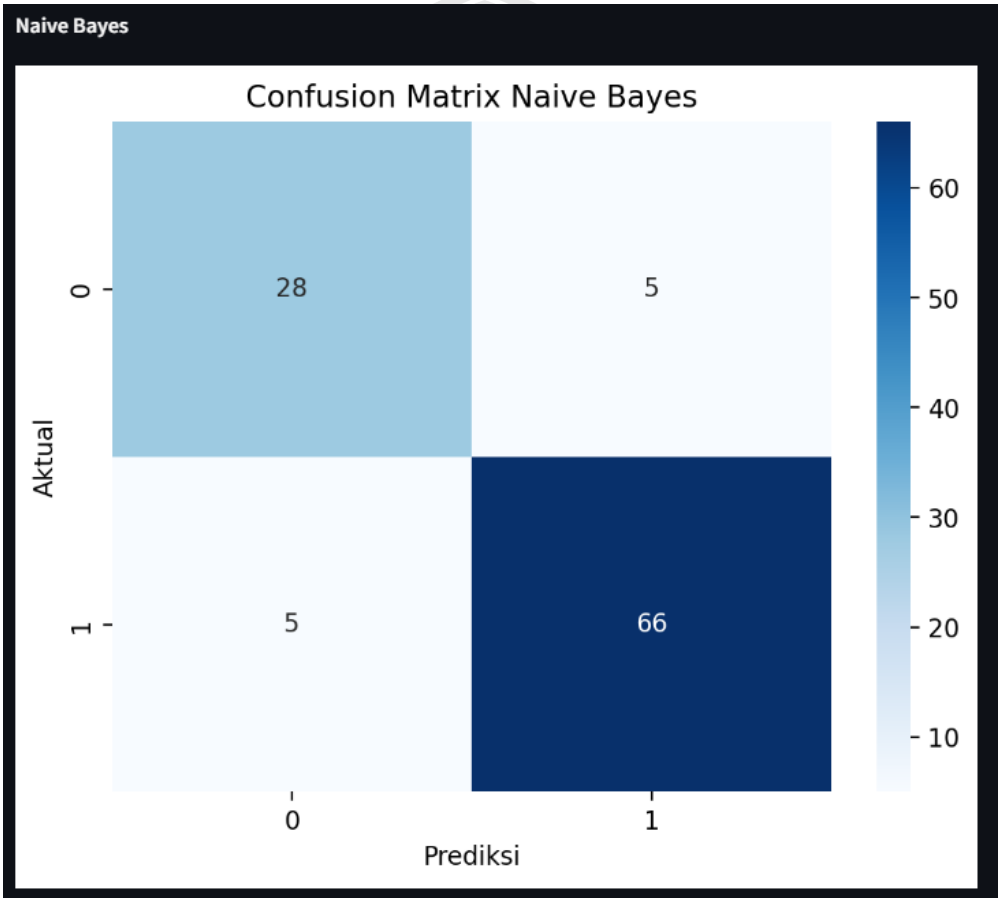
Gambar 4. 8 Menampilkan Hasil Visualisasi Pohon Keputusan Decision Tree C4.5 pada Aplikasi

Root node, atau akar pada pohon keputusan (decision tree), merupakan titik awal dari struktur pohon tersebut, di mana proses klasifikasi dimulai. Pada tahap ini, data pertama kali dibagi berdasarkan atribut tertentu yang dianggap paling relevan. Atribut yang dipilih sebagai root node biasanya adalah atribut yang memberikan informasi paling banyak atau yang paling efektif dalam membagi data untuk tujuan klasifikasi. Dalam konteks penelitian ini, atribut Polyuria terpilih sebagai root node. Pemilihan atribut ini menunjukkan bahwa Polyuria memiliki pengaruh paling signifikan dalam memisahkan data untuk prediksi, dibandingkan dengan atribut lainnya yang ada dalam dataset.

Keberadaan root node yang tepat sangat penting karena akan mempengaruhi seluruh struktur pohon keputusan yang dibangun. Dengan memilih atribut yang paling informatif, proses pembagian data selanjutnya menjadi lebih efisien, sehingga setiap cabang yang dihasilkan dapat mengarah

pada klasifikasi yang lebih akurat. Selain itu, pemilihan atribut yang baik juga membantu dalam mengurangi kompleksitas pohon, yang pada gilirannya dapat meningkatkan kecepatan dan efisiensi dalam proses pengambilan keputusan. Oleh karena itu, pemilihan atribut yang tepat, seperti Polyuria dalam hal ini, sangat krusial untuk mencapai hasil yang optimal dalam analisis data dan prediksi yang dilakukan.

B. Naïve Bayes



Gambar 4. 9 Menampilkan Hasil Akurasi dan Confusion Matriks pada Naïve Bayes

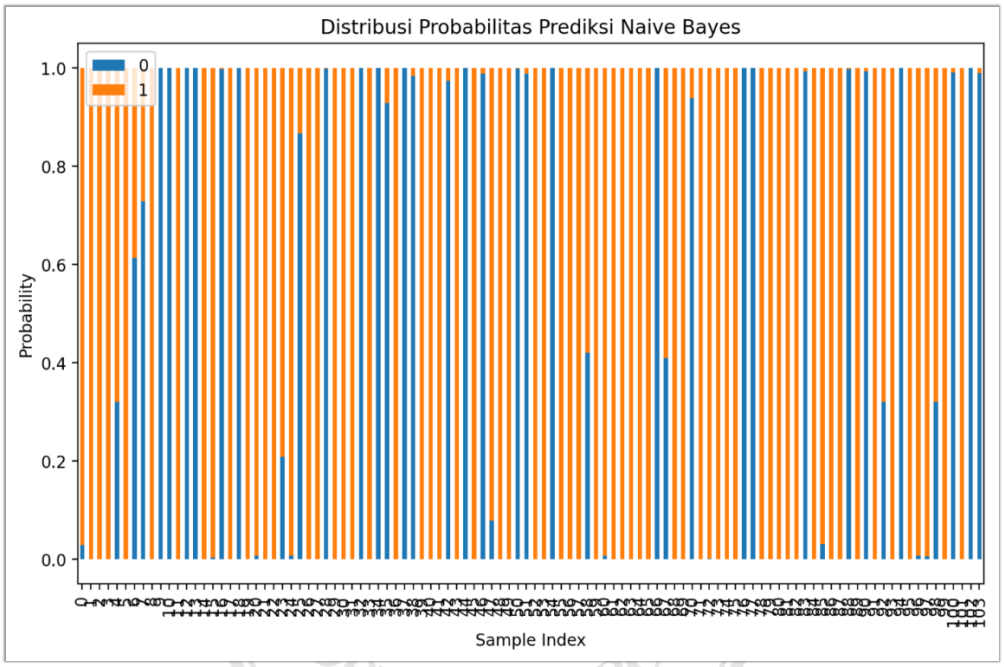
Berikut adalah nilai dari Confusion Matrix untuk Naïve Bayes:

1) True Positive (TP) : 66 Data positif yang diprediksi dengan benar.
2) True Negative (TN) : 28 Data negatif yang diprediksi dengan benar

3) False Positive (FP) : 5 Data negatif yang diprediksi sebagai positif.

4) False Negative (FN) : 5 Data positif yang diprediksi sebagai negatif.

Salah satu cara yang sederhana namun efektif adalah dengan memvisualisasikan distribusi probabilitas prediksi untuk setiap kelas yang dapat dilihat sebagai berikut :



Gambar 4. 10 Menampilkan Bar Distribusi Probabilitas Naïve Bayes pada Aplikasi

4.6 Ringkasan Analisis Model Menggunakan Python

Tabel 4. 1 Analisis Model Menggunakan Python

No.	Algoritma	Akurasi	Precision	Recall
1	Decision Tree C4.5	100 %	100%	100%
2	Naïve Bayes	90,38%	90,38%	90,38%

Pada analisis model menggunakan Python dapat dilihat nilai algoritma Decision Tree C4.5 memiliki performa lebih baik dari pada Naive Bayes dengan nilai akurasi 100%, Precision 100%, Recall 100%. Namun untuk algoritma Naïve Bayes memiliki nilai performa yang sedikit kurang baik dari pada algoritma C4.5 dengan nilai akurasi 90,38%, Precision 90,38% dan Recall 90,38%.

BAB V

PENUTUP

4.1 Kesimpulan

Berdasarkan hasil penelitian ini dapat disimpulkan bahwa dalam penelitian ini, penulis telah melakukan evaluasi yang komprehensif terhadap performa algoritma klasifikasi C4.5 dan Naïve Bayes dalam konteks prediksi diagnosis penyakit diabetes menggunakan bahasa pemrograman Python berbasis web. Hasil analisis menunjukkan bahwa algoritma C4.5 memiliki performa yang superior dibandingkan Naïve Bayes, dengan akurasi mencapai 100% pada platform Python. Temuan ini menegaskan bahwa C4.5 lebih efektif dalam menangani dataset yang digunakan, yang mencakup berbagai atribut kesehatan, dan memberikan wawasan penting bagi pengembangan sistem prediksi kesehatan yang lebih akurat dan efisien.

Dalam implementasi menggunakan Python, penulis memanfaatkan berbagai library seperti Pandas dan Scikit-learn untuk penerapan algoritma. Hal ini menunjukkan bahwa pemilihan alat dan teknik yang tepat dalam pengolahan data sangat berpengaruh terhadap hasil akhir dari analisis yang dilakukan.

Meskipun algoritma C4.5 menunjukkan hasil yang lebih baik dalam hal akurasi, precision, dan recall, Naïve Bayes tetap memiliki keunggulan yang signifikan dalam hal kecepatan dan efisiensi, terutama ketika dihadapkan pada dataset yang lebih besar. Naïve Bayes, yang berbasis pada prinsip probabilitas

bersyarat, memiliki keunggulan dalam hal waktu komputasi yang lebih cepat karena proses pelatihannya yang relatif sederhana.

Penelitian ini memiliki implikasi yang signifikan bagi praktik kesehatan, terutama dalam meningkatkan akurasi diagnosis penyakit diabetes. Dengan penerapan algoritma klasifikasi yang tepat, tenaga medis dapat lebih cepat dan akurat dalam mengidentifikasi pasien yang berisiko. Temuan ini menunjukkan potensi besar dari teknologi data dalam mendukung keputusan klinis dan meningkatkan efektivitas intervensi medis..

4.2 Saran

Untuk memperkaya hasil penelitian ini, disarankan agar penelitian selanjutnya mempertimbangkan eksplorasi algoritma klasifikasi tambahan, seperti Random Forest atau Support Vector Machine. Hal ini dapat memberikan perspektif yang lebih luas mengenai efektivitas berbagai metode dalam prediksi diagnosis diabetes.

Selain itu, melakukan pengujian dengan berbagai dataset yang berbeda dapat membantu dalam memahami performa algoritma dalam konteks yang beragam. Penelitian selanjutnya juga dapat mempertimbangkan optimasi parameter menggunakan teknik seperti Grid Search untuk meningkatkan akurasi model.

Akhirnya, integrasi hasil dari berbagai algoritma dalam satu sistem dapat menciptakan model yang lebih komprehensif dan akurat, yang pada gilirannya dapat berkontribusi pada peningkatan kualitas layanan kesehatan dan deteksi dini penyakit diabetes. Dengan langkah-langkah ini, penelitian selanjutnya dapat lebih mendalam dan aplikatif dalam konteks kesehatan

DAFTAR PUSTAKA

[1] American Diabetes Association, "Diagnosis and classification of diabetes mellitus," *Diabetes Care*, vol. 37, no. Issue Supplement_1, pp. S81–S90, Jan. 2014, doi: 10.2337/dc14-S081.

[2] International Diabetes Federation (IDF), "Diabetes in Indonesia (2021)." Accessed: Jul. 19, 2024. [Online]. Available: https://idf.org/our-network/regions-and-members/western-pacific/members/indonesia/

[3] Kementerian Kesehatan RI, "Survei Kesehatan Indonesia Tahun 2023," 2023.

[4] S. Hafni Sahir, *Metodologi Penelitian*, Cetakan I., vol. I. Medan: KBM INDONESIA, 2021.

[5] W. Ghofur, I. K. Rahman, and A. H. Al Kattani, "Pendidikan Pornografi di Kalangan Mahasiswa," *JIIP - Jurnal Ilmiah Ilmu Pendidikan*, vol. 6, no. 3, pp. 1499–1506, 2023, doi: 10.54371/jiip.v6i3.1697.

[6] N. Attamami, A. Triayudi, and R. T. Aldisa, "Analisis Performa Algoritma Klasifikasi Naive Bayes dan C4.5 Teknologi Komunukasi dan Indormatika," *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, p. 2, Feb. 2023, doi: 10.35870/jti.

[7] M. R. Fanani and D. S. Sintia, "Klasifikasi Kesiapan Anak Masuk Sekolah Dasar MenggunakanAlgoritma Naïve Bayes dan Algoritma C4.5," *INNOVATIVE: Journal Of Social Science Research*, vol. 4 Nomor 3 Tahun 2024, no., pp. 10547–10555, 2024.

[8] N. Gusrialni Fitri, S. Adilya, and F. Azizi, "Perbandingan Klasifikasi Naive Bayes dan C4.5 untuk Diagnosa Penyakit Stroke," vol., no., pp. 49–55, Aug. 2023.

[9] J. Han, M. Kamber, and J. Pei, *DATA MINING Concepts and Techniques*, 3rd ed., vol. 3. 2011.

[10] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, vol. 3. 2011.

[11] K. P. Murphy, *Machine Learning - A Probabilistic Perspective*, vol. 1. 2012.

[12] R. A. Santoso, D. Syauqy, M. Hannats, and H. Ichsan, "Pengembangan Sistem Prediksi Hama Wereng Berdasarkan Data Cuaca Sensor Dan Cuaca Online Menggunakan Metode Naive Bayes," *Jurnal Pengembangan*

Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 10, pp. 4002–4010, 2018, [Online]. Available: http://j-ptiik.ub.ac.id

[13] H. Nurul, "Visual Studio Code: Pengertian, Fitur, Keunggulan dan Jenisnya," dewaweb. Accessed: Sep. 24, 2024. [Online]. Available: https://www.dewaweb.com/blog/mengenal-visual-studio-code/

[14] R. W. Fenia, "Bahasa Pemrograman Python: Pengertian, Fungsi, Kelebihan, dan Contoh Program," 16 Februari 2023. Accessed: Aug. 07, 2024. [Online]. Available: https://www.kmtech.id/post/bahasa-pemrograman-python-pengertian-fungsi-kelebihan-dan-contoh-program

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 9, pp. 2825–2830, 2011.

[16] J. Brownlee, *Machine Learning Mastery with Python: Understand You Data, Create Accurate Models and Work Projects End-to-End*, vol. 1.4. 2016.

[17] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual; CreateSpace*, vol. 2. CreateSpace Independent Publishing Platform, 2009.

[18] F. Chollet, *Deep Learning with Python*, 2nd ed., vol. 2, no. or ML. Shelter Island: Manning Publications Co, 2018. Accessed: Aug. 18, 2024. [Online]. Available: https://sourestdeeds.github.io/pdf/Deep%20Learning%20with%20Python.pdf

LAMPIRAN

Lampiran A.1 Segmen Program 4. 11 Perbandingan Perhitungan Performa Kedua Model

```
1.  : # Comparison Page
2.  : elif page == 'Bandingkan Akurasi':
3.  :       st.header("Perbandingan Akurasi dan
            Statistik Lainnya")
4.  :       if st.session_state['data_test'] is not
            None and st.session_state['dt_model'] is not None
            and st.session_state['nb_model'] is not None:
5.  :           testing_data = st.session_state
                ['data_test']
6.  :           features = st.session_state['features']
7.  :           target = st.session_state['target']
8.  :           # Ensure the target column is correctly
                named in the test data
9.  :           if set(features).issubset(testing_data.
                columns) and target in testing_data.
                columns:
10. :               X_test = testing_data[features]
11. :               y_test = testing_data[target]
12. :               # Decision Tree Metrics
13. :               dt_predictions =
                    st.session_state['dt_model'].predict
                    (X_test)
14. :               dt_accuracy = accuracy_score(y_test,
                    dt_predictions) * 100
15. :               dt_conf_matrix = confusion_matrix
                    (y_test, dt_predictions)
16. :               dt_classification_report =
                    classification_report(y_test,
                    dt_predictions, output_dict=True)
17.                 # Naive Bayes Metrics
18. :               nb_predictions = st.session_state
                    ['nb_model'].predict(X_test)
19. :               nb_accuracy = accuracy_score(y_test,
                    nb_predictions) * 100
20. :               nb_conf_matrix = confusion_matrix
                    (y_test, nb_predictions)
21. :               nb_classification_report =
                    classification_report(y_test,
                    nb_predictions, output_dict=True)
22. :               # Display Results
23. :               st.write("### Akurasi")
24. :               st.write(f"**Decision Tree:**
                    {dt_accuracy:.2f}%")
25. :               st.write(f"**Naive Bayes:**
                    {nb_accuracy:.2f}%")
26. :               st.write("### Precision, Recall, F1-
                    Score")
                    comparison_df = pd.DataFrame({
```

Lampiran A.1 Lanjutan Segmen Program 4.11

```
27. :                   'Metric': ['Precision (%)','Recall
                        (%)', 'F1-Score (%)'],
28. :                   'Decision Tree': [
29. :                           round(dt_classification_
                                report['weighted avg']
                                ['precision'] * 100, 2),
30. :                           round(dt_classification_
                                report['weighted avg']
                                ['recall'] * 100, 2),
31. :                           round(dt_classification_
                                report['weighted avg']
                                ['f1-score'] * 100, 2)
32. :                   ],
33. :                   'Naive Bayes': [
34. :                           round(nb_classification_
                                report['weighted avg']
                                ['precision'] * 100, 2),
35. :                           round(nb_classification_
                                report['weighted avg']
                                ['recall'] * 100, 2),
36. :                           round(nb_classification_
                                report['weighted avg']
                                ['f1-score'] * 100, 2)
37. :                   ]
38. :               })
39. :               st.write(comparison_df)
40. :               # Display Confusion Matrices
41. :               st.write("### Confusion Matrix")
42. :               st.write("**Decision Tree**")
43. :               plot_confusion_matrix(dt_conf_matrix,
                    "Confusion Matrix Decision Tree")
44. :               st.write("**Naive Bayes**")
45. :               plot_confusion_matrix(nb_conf_matrix,
                    "Confusion Matrix Naive Bayes")
47. :           else:
48. :                 st.error("Data testing tidak memiliki
                      fitur atau kolom target yang
                      sesuai.")
49. : else:
50. :       st.error("Data testing atau model belum
            tersedia. Harap unggah data testing.”)
```