

**ANALISIS PERBANDINGAN ALGORITMA DECISION TREE, RANDOM
FOREST, DAN NAIVE BAYES DALAM KLASIFIKASI GANGGUAN
TIDUR**

SKRIPSI



Oleh:

INDAH NOVITA SARI

2021502067

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO**

2025

**ANALISIS PERBANDINGAN ALGORITMA DECISION TREE, RANDOM
FOREST, DAN NAIVE BAYES DALAM KLASIFIKASI GANGGUAN
TIDUR**

SKRIPSI

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

Oleh:

INDAH NOVITA SARI

2021502067

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO**

2025

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : **Indah Novita Sari**

NPM/NIRM : 2021502067

Program Studi : S-1 Sistem Informasi

Fakultas : Fakultas Sains dan Teknologi

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

Situbondo, 10 Juni 2025

Saya yang menyatakan,



Indah Novita Sari

PERSETUJUAN PEMBIMBING

Nama : **Indah Novita Sari**

NPM/NIRM : **2021502067**

Judul : **Analisis Perbandingan Algoritma Decision Tree,
Random Forest, dan Naive Bayes dalam Klasifikasi
Gangguan Tidur**

Telah disetujui oleh:

Pembimbing I

Pembimbing II



Lukman Fakhri Lidimilah, M.Kom

NIDN : 0715099001



Ahmad Lutfi, M.Kom

NIDN : 0714108803

PENGESAHAN
SKRIPSI
ANALISIS PERBANDINGAN ALGORITMA DECISION TREE, RANDOM
FOREST, DAN NAIVE BAYES DALAM KLASIFIKASI GANGGUAN
TIDUR

INDAH NOVITA SARI

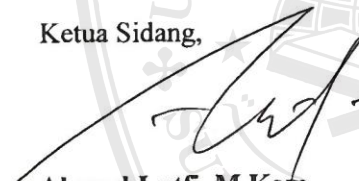
2021502067

Telah dipertahankan di depan dewan penguji Sidang/Munaqosah Skripsi pada hari Sabtu, tanggal 16 Agustus 2025 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom) pada Fakultas Sains dan Teknologi Universitas Ibrahimy.

Tim Penguji,

Ketua Sidang,

Sekretaris Sidang,


Ahmad Lutfi, M.Kom

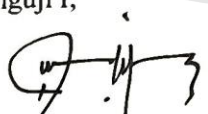
NIDN. 0714108803


Aisyatin Masruroh, S.Kom

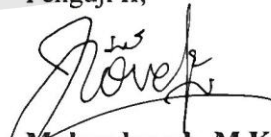
NIDN.

Penguji I,

Penguji II,


Irma Yunita, M.Kom

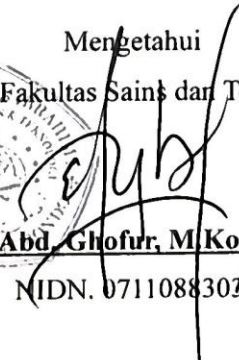
NIDN. 0719118404


Muhasshanah, M.Kom

NIDN. 0721058704

Mengetahui

Dekan Fakultas Sains dan Teknologi,


Abd. Ghofur, M.Kom

NIDN. 0711088303

MOTTO

“Orang yang berhasil adalah dia yang mau mencoba.

So, always try till die”.

بِبَرَكَهٖ الْقُرْآنُ



PERSEMBAHAN

Dengan penuh rasa syukur, skripsi ini saya persembahkan kepada:

1. Allah SWT yang telah memberi petunjuk dan rahmat-Nya sehingga skripsi ini dapat terselesaikan dengan baik.
2. Sang kekasih hati pemberi syafa'at didunia dan akhirat Baginda Nabi Muhammad SAW.
3. Murabbi ruhina KHR. Ach. Azaim Ibrahimy, S.Sy., M.H. dan seluruh keluarga besar Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo.
4. Orang tua tercinta, bapak Budiono dan ibu Halifah atas segala dukungan, cinta, pengorbanan, dan kekuatan hingga saya bisa berada pada titik ini.
5. Saudara – saudara tersayangku, Nanda Hairul Iqbal, Fathul Huda Afandi, Yulika Devi Cindiana, dan segenap keluarga besar yang selalu menjadi penyemangat dalam setiap langkah saya. Dan kepada Muhammad Idkhun Holiday, Siti Maysaroh, dan Muhammad Ridhoul Jinan yang telah hadir menjadi teman, saudara, motivator, dan *support* sistem ditengah – tengah perjuangan ini.
6. Para dosen pembimbing bapak Lukman Fakhid Lidimillah, M.Kom. dan bapak Ahmad Lutfi, M.Kom. yang sedari awal telah membimbing, mengarahkan dan mendampingi dalam penyusunan skripsi ini.
7. Ketua Program Studi Sistem Informasi bapak Achmad Baijuri, M.Kom serta segenap dosen dan keluarga besar Fakultas Sains dan Teknologi yang telah mendidik dan memberikan ilmu untuk meraih masa depan.
8. Teman seperjuangan mahasiswi Sistem Informasi angkatan 2021 yang telah menjadi partner dan guru terbaik selama 4 tahun ini.
9. Teman – teman diranah pengabdian tanpa batas yang telah menemani disetiap langkah pengabdian segenap Pengurus Rayon IKSASS Bondowoso khususnya Banom Q-blait.
10. Teman – teman seperjuangan di Badan Eksekutif Mahasiswa (BEM) Fakultas Sains dan Teknologi khususnya teman - teman di Media Saintek Nabila Khansa Noer Maudy, Angeli Dwiyantri Nur Azizah, Riska Aprilia, Nur Aini, dan Aisyah Putri Sabrina.
11. Teman seperjuangan di Jam'iyah Khottotin Pusat Putri yang selalu memberi semangat dan menemani dalam perjuangan ini.
12. Segenap keluarga besar Imaratut Takhshis No.01 yang telah menjadi teman, keluarga, serta tempat ternyaman untuk pulang ketika lelah dalam perjuangan.
13. Seseorang yang tidak bisa disebut namanya, yang menjadi sosok dibalik layar.

KATA PENGANTAR

Segala puji syukur peneliti panjatkan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya, skripsi berjudul “**Analisis Perbandingan Algoritma Decision Tree, Random Forest, dan Naive Bayes dalam Klasifikasi Gangguan Tidur**” dapat terselesaikan dengan baik dan lancar sebagai salah satu syarat menempuh program sarjana S-1.

Penyusunan skripsi ini semaksimal mungkin peneliti lakukan dan tentunya dengan dukungan dan bimbingan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, peneliti menyampaikan banyak terimakasih kepada :

1. Murobbi ruhina KHR. Ach. Azaim Ibrahimy, S.Sy., M.H. selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo
2. KH. Ach. Fadlail, S.H., M.H. selaku Rektor Universitas Ibrahimy
3. Bapak Abd. Ghofur, M.Kom. selaku Dekan Fakultas Sains dan Teknologi
4. Bapak Achmad Baijuri, M.Kom. selaku Ketua Program Studi Sistem Informasi
5. Bapak Lukman Fakhri Lidimillah, M.Kom. selaku pembimbing 1
6. Bapak Ahmad Lutfi, M.Kom. selaku pembimbing 2

Semoga segala amal baik yang telah diberikan mendapat balasan sebaik mungkin dari Allah SWT, Aamiin.

Situbondo, 10 Juni 2025

Peneliti

ABSTRAK

Indah Novita Sari 2025. **Analisis Perbandingan Algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* dalam Klasifikasi Gangguan Tidur.** Skripsi, Program Studi Sistem Informasi, Universitas Ibrahimy. Pembimbing: (I) Lukman Fakhri Lidimilah, M.Kom., Pembimbing (II) Ahmad Lutfi, M.Kom.

Analisis data memiliki banyak metode salah satunya adalah data mining dengan teknik klasifikasi berbasis *machine learning*. Penelitian ini membandingkan tiga algoritma data mining yaitu algoritma *Decision Tree*, algoritma *Random Forest*, dan algoritma *Naive Bayes*. Ketiga algoritma ini dipilih untuk menentukan algoritma mana yang efektif dan memiliki performa terbaik dalam proses klasifikasi. Klasifikasi merupakan proses menganalisis data dengan mengelompokkan sampel data berdasarkan kategori atau label yang sudah ditentukan sebelumnya menggunakan teknik *supervised learning*. Penelitian ini menggunakan kasus gangguan tidur sebagai sample datanya. Gangguan tidur adalah kondisi seseorang yang mengalami masalah terkait durasi, kualitas atau pola tidur. Kebanyakan orang tidak menghiraukan tanda – tanda gangguan tidur yang mereka alami sehingga mereka tidak menyadari bahwa bisa saja mereka mengalami gangguan tidur. Deteksi dini gangguan tidur dapat membantu mencegah berbagai masalah kesehatan yang serius. Penelitian ini dilakukan untuk menganalisis performa dari masing – masing algoritma dan memberikan rekomendasi algoritma terbaik untuk klasifikasi gangguan tidur serta pengembangan sistem deteksi gangguan tidur. Penelitian ini membandingkan hasil performa dari algoritma *Decision Tree*, algoritma *Random Forest*, dan algoritma *Naive Bayes*. Data yang digunakan yaitu *Sleep Health and Lifestyle Dataset* yang diperoleh secara online melalui *Open Data Source* yakni dari *Platform Kaggle* yang terdiri dari 374 data dan 13 atribut. Berdasarkan hasil pengujian *Decision Tree*, *Random Forest*, dan *Naive Bayes*, diperoleh hasil rata – rata akurasi *Decision Tree* 91.49%, *Random Forest* 87.23%, dan *Naive Bayes* 76.59%. Dapat disimpulkan bahwa algoritma *Decision Tree* memberikan performa terbaik dalam klasifikasi gangguan tidur.

Kata Kunci: Klasifikasi, Gangguan Tidur, *Decision Tree*, *Random Forest*, *Naive Bayes*

ABSTRACT

Indah Novita Sari 2025. *Comparative Analysis of Decision Tree, Random Forest, and Naive Bayes Algorithm in Sleep Disorder Classification*. Thesis, Information Systems Study Program, Ibrahmy University. Supervisor (I) Lukman Fakhri Lidimilah, M.Kom., Supervisor (II) Ahmad Lutfi, M.Kom.

Data analysis has many methods, one of which is data mining with machine learning-based classification techniques. This study compares three data mining algorithms, namely the Decision Tree algorithm, the Random Forest algorithm, and the Naïve Bayes algorithm. These three algorithms were chosen to determine which algorithm is effective and has the best performance in the classification process. Classification is the process of analyzing data by grouping data samples based on predetermined categories or labels using supervised learning techniques. This study uses sleep disorder cases as its data samples. Sleep disorders are conditions in which a person experiences problems related to the duration, quality or pattern of sleep. Most people ignore the signs of sleep disorders they experience so they do not realize that they could be experiencing sleep disorders. Early detection of sleep disorders can help prevent various serious health problems. This study was conducted to analyze the performance of each algorithm and provide recommendations for the best algorithm for classifying sleep disorders and developing a sleep disorder detection system. This study compares the performance results of the Decision Tree algorithm, the Random Forest algorithm, and the Naïve Bayes algorithm. The data used is the Sleep Health and Lifestyle Dataset obtained online through Open Data Source, namely from the Kaggle Platform, which consists of 374 data and 13 attributes. Based on the test results of Decision Tree, Random Forest, and Naïve Bayes, the average accuracy of Decision Tree was 91.49%, Random Forest 87.23%, and Naïve Bayes 76.59%. It can be concluded that the Decision Tree algorithm provides the best performance in classifying sleep disorders.

Keyword: Classification, Sleep Disorders, Decision Trees, Random Forest, Naïve Bayes

DAFTAR ISI

PERNYATAAN KEASLIAN TULISAN	II
PERSETUJUAN PEMBIMBING	III
PENGESAHAN	IV
MOTTO	V
PERSEMBAHAN.....	VI
KATA PENGANTAR.....	VII
ABSTRAK	VIII
<i>ABSTRACT</i>	IX
DAFTAR ISI.....	X
DAFTAR GAMBAR	XII
DAFTAR TABEL	XIV
DAFTAR SEGMENT PROGRAM.....	XV
DAFTAR RUMUS.....	XVII
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	5
1.3 Rumusan Masalah.....	6
1.4 Batasan Masalah	6
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian	7
1.7 Metode Penelitian.....	7
1.7.1 Jenis Penelitian	7
1.7.2 Teknik Pengumpulan Data.....	7
1.8 Sistematika Pembahasan	8
BAB II TINJAUAN PUSTAKA.....	10
2.1 Penelitian Terdahulu.....	10
2.2 Landasan Teori.....	13
2.3 Pemodelan.....	21
2.4 Perangkat Lunak yang Digunakan	23
BAB III ANALISIS DAN PERANCANGAN SISTEM	26
3.1 Gambaran Umum Objek Penelitian.....	26
3.2 Alur Proses	26
3.2.1 Identifikasi dan Analisis Proses Bisnis	27
3.2.2 Identifikasi dan Analisis Kebutuhan	65
3.3 Desain Sistem	67
3.3.1 Desain Output.....	68
3.3.2 Desain Input.....	69
3.3.3 Desain Proses	70
3.3.4 Identifikasi dan Desain <i>User Interface</i>	73
BAB IV IMPLEMENTASI SISTEM.....	76
4.1 Konstruksi Sistem	76
4.1.1 Kebutuhan Sistem	76
4.1.2 Instalasi Sistem	77
4.1.3 Segment Program	82

4.2	Skenario Pengujian	114
4.2.1	Pengujian Terhadap Login	114
4.2.2	Pengujian Terhadap Transaksi	116
4.3	Pengujian	118
4.3.1	Cara Kerja Sistem.....	119
4.3.2	Hasil Pengujian	127
4.4	Maintenance	129
BAB V PENUTUP		130
5.1	Kesimpulan	130
5.2	Saran	131
DAFTAR PUSTAKA.....		132



DAFTAR GAMBAR

Gambar 3.1 <i>Sleep Health and Lifestyle Dataset</i>	28
Gambar 3.2 Hasil Penghapusan Nilai Kosong	31
Gambar 3.3 <i>Confusion Matrix Heatmap Decision Tree</i>	43
Gambar 3.4 <i>Decision Tree Result</i>	45
Gambar 3.5 <i>Confusion Matrix Heatmap Random Forest</i>	51
Gambar 3.6 Fitur Teratas <i>Random Forest</i>	53
Gambar 3.7 Salah Satu Pohon <i>Random Forest</i>	55
Gambar 3.8 <i>Confusion Matrix Heatmap Naive Bayes</i>	59
Gambar 3.9 Output Evaluasi Hasil dan Perbandingan	64
Gambar 3.10 Desain Output <i>Preview</i>	68
Gambar 3.11 Desain Output Hasil Klasifikasi	68
Gambar 3.12 Desain Data Klasifikasi	69
Gambar 3.13 Desain Output Grafik Hasil Perbandingan Model	69
Gambar 3.14 Desain Input Dataset	70
Gambar 3.15 Desain Input Data Klasifikasi	70
Gambar 3.16 Arsitektur Aplikasi	72
Gambar 3.17 <i>Context Diagram</i>	72
Gambar 3.18 Desain <i>User Interface Home</i>	74
Gambar 3.19 Desain <i>Dashboard Admin</i>	74
Gambar 3.20 Desain <i>User Interface Login User</i>	75
Gambar 3.21 Desain <i>User Interface Login Admin</i>	75
Gambar 4.1 Instalasi <i>Python</i>	78
Gambar 4.2 Instalasi <i>Visual Studio Code</i>	78

Gambar 4.3 Instalasi <i>Web Framework Flask</i>	79
Gambar 4.4 Instalasi <i>Scikit-learn, Pandas, Numpy, Matplotlib, Imblearn</i>	80
Gambar 4.5 Folder Sistem	80
Gambar 4.6 <i>Terminal</i> Sistem	81
Gambar 4. 7 URL pada <i>Browser</i>	81
Gambar 4.8 Halaman Awal Sistem	82
Gambar 4.9 <i>Home</i>	119
Gambar 4.10 <i>Home</i> Tentang	120
Gambar 4.11 <i>Home</i> Fitur Aplikasi	120
Gambar 4.12 <i>Home</i> Kontak & Informasi	120
Gambar 4.13 <i>Login User</i>	121
Gambar 4.14 <i>Login Admin</i>	121
Gambar 4.15 <i>Register</i>	122
Gambar 4.16 Klasifikasi	122
Gambar 4.17 Hasil Klasifikasi.....	123
Gambar 4.18 <i>Dashboard Admin</i>	124
Gambar 4.19 <i>Kelola User</i>	124
Gambar 4.20 <i>Upload Dataset</i>	125
Gambar 4.21 <i>Preview Data</i>	125
Gambar 4.22 Hasil <i>Training</i> dan <i>Testing</i>	126
Gambar 4.23 Data Klasifikasi	126
Gambar 4.24 Grafik Perbandingan Model	127

DAFTAR TABEL

Tabel 2.1 Simbol – Simbol <i>Context Diagram</i>	22
Tabel 3.1 Hasil Mengisi <i>Missing Value</i>	32
Tabel 3.2 Hasil <i>Encoding</i> Kolom Kategorikal.....	34
Tabel 3.3 Hasil Seleksi dan Transformasi Data	36
Tabel 3.4 Hasil Pengujian <i>Decision Tree</i>	39
Tabel 3.5 <i>Confusion Matrix Decision Tree</i>	41
Tabel 3.6 Hasil Pengujian <i>Random Forest</i>	47
Tabel 3.7 <i>Confusion Matrix Random Forest</i>	49
Tabel 3.8 Hasil Pengujian <i>Naive Bayes</i>	56
Tabel 3.9 <i>Confusion Matrix Naive Bayes</i>	57
Tabel 3.10 <i>Class Prior & Posterior Probability</i>	61
Tabel 3.11 Ringkasan Perbandingan.....	65
Tabel 3.12 Identifikasi Desain Proses.....	71
Tabel 4.1 Pengujian <i>Login User</i>	114
Tabel 4.2 Skenario <i>Login Admin</i>	115
Tabel 4.3 Skenario Klasifikasi.....	116
Tabel 4.4 Skenario Hasil Klasifikasi	117
Tabel 4.5 Skenario Kelola <i>User</i>	117
Tabel 4.6 Skenario <i>Upload Dataset</i>	117
Tabel 4.7 Skenario Data Klasifikasi	118
Tabel 4.8 Skenario Grafik.....	118
Tabel 4.9 Hasil Pengujian.....	127

DAFTAR SEGMENT PROGRAM

Segmen Program 3.1 Pengumpulan Data.....	29
Segmen Program 3.2 Menghapus Nilai Kosong	30
Segmen Program 3.3 Mengisi <i>Missing Value</i>	31
Segmen Program 3.4 <i>Encoding</i> Kolom Kategorikal.....	33
Segmen Program 3.5 Seleksi dan Transformasi Data	35
Segmen Program 3.6 Pembagian Data.....	37
Segmen Program 3.7 <i>Decision Tree</i>	38
Segmen Program 3.8 <i>Confusion Matrix Heatmap Decision Tree</i>	42
Segmen Program 3.9 <i>Decision Tree Result</i>	44
Segmen Program 3.10 <i>Random Forest</i>	46
Segmen Program 3.11 <i>Confusion Matrix Heatmap Random Forest</i>	50
Segmen Program 3.12 Fitur Teratas <i>Random Forest</i>	52
Segmen Program 3.13 Salah Satu Pohon <i>Random Forest</i>	54
Segmen Program 3.14 <i>Naive Bayes</i>	55
Segmen Program 3.15 <i>Confusion Matrix Heatmap Naive Bayes</i>	58
Segmen Program 3.16 <i>Class Prior & Posterior Probability</i>	60
Segmen Program 3.17 Visualisasi Hasil dan Perbandingan.....	62
Segmen Program 4.1 Home	83
Segmen Program 4.2 <i>Login User</i>	92
Segmen Program 4.3 <i>Login Admin</i>	95
Segmen Program 4.4 Register.....	98
Segmen Program 4.5 Klasifikasi.....	99
Segmen Program 4.6 Hasil Klasifikasi	101
Segmen Program 4.7 <i>Dashboard Admin</i>	104
Segmen Program 4.8 <i>Kelola User</i>	105
Segmen Program 4.9 <i>Upload Dataset</i>	106
Segmen Program 4.10 Data Klasifikasi	107

Segmen Program 4.11 Grafik	109
----------------------------------	-----



DAFTAR RUMUS

Rumus 2.1 <i>Entropi</i>	15
Rumus 2.2 <i>Information Gain</i>	16
Rumus 2.3 <i>Bootstrap Sampling</i>	17
Rumus 2.4 Pemilihan Fitur Secara Acak.....	17
Rumus 2.5 Kriteria <i>Split</i>	17
Rumus 2.6 Fitur Terbaik.....	17
Rumus 2.7 Kriteria <i>Split</i>	18
Rumus 2.8 Voting Mayoritas.....	18
Rumus 2.9 Rata – Rata Prediksi.....	18
Rumus 2.10 Rumus Dasar <i>Naive Bayes</i>	18
Rumus 2.11 Menghitung <i>Prior Probabilitas</i>	19
Rumus 2.12 Menghitung <i>Likelihood</i>	19
Rumus 2.13 Diskrit Frekuensi Relatif.....	19
Rumus 2.14 Kontinu Distribusi Probabilitas.....	20
Rumus 2.15 Rumus Utama <i>Posterior Probabilitas</i>	20
Rumus 2.16 $P(X)$ Konstran Semua Kelas	20
Rumus 2.17 Menentukan Kelas	20

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di masa digitalisasi teknologi seperti saat ini, data menjadi elemen kunci dalam berbagai bidang, termasuk di sektor kesehatan. Dalam analisis data, terdapat sejumlah metode yang dapat diterapkan, diantaranya adalah *data mining* menggunakan teknik klasifikasi berbasis *machine learning*. Analisis data menggunakan teknik *data mining* sangat umum digunakan [1], algoritma yang diterapkan pada penelitian ini yaitu perbandingan antara tiga algoritma *machine learning* yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes*.

Decision Tree merupakan metode klasifikasi berbentuk pohon keputusan yang digunakan dengan memanfaatkan aturan berbasis atribut data untuk klasifikasi dan regresi dengan memanfaatkan aturan berbasis fitur dalam data. Keunggulan utama dari algoritma *Decision Tree* adalah kemudahan interpretasi, efisiensi dalam perhitungan, serta kemampuannya dalam menangani data dengan atribut *numerik* dan *kategorikal*. Algoritma *Random Forest*, merupakan pengembangan atau perluasan dari algoritma *Decision Tree*, dengan membangun beberapa pohon keputusan secara acak yang menggabungkan hasil prediksi untuk meningkatkan akurasi dan meminimalkan kemungkinan terjadinya *overfitting*. Algoritma ini membuat beberapa pohon keputusan yang dipilih secara acak, kemudian mengombinasikan hasil prediksi dari tiap pohon tersebut digabungkan untuk memperoleh hasil akhir yang optimal dan akurat. Sementara itu, algoritma *Naive Bayes* merupakan algoritma berbasis *probabilitas* dengan mengasumsikan

independensi antar fitur dalam data. Algoritma ini dikenal sangat efisien untuk klasifikasi teks dan aplikasi yang melibatkan dataset besar karena memiliki waktu komputasi yang cepat dan tidak terlalu kompleks dalam implementasi [2]. Ketiga algoritma ini digunakan untuk mengetahui algoritma mana yang efektif dan memiliki performa terbaik dalam mengklasifikasikan gangguan tidur.

Banyak penelitian telah menggunakan teknik klasifikasi [3]. Klasifikasi adalah cara menganalisis data dengan mengelompokkan sampel ke dalam kategori atau label yang sudah ditentukan sebelumnya. Metode ini menggunakan teknik *supervised learning* untuk mencari pola atau hubungan antara data masukan dan target yang ingin diprediksi [4]. Dalam penerapannya, dibutuhkan model klasifikasi yang mampu memberikan hasil akurat sekaligus efisien untuk mendukung sistem deteksi gangguan tidur, sehingga dapat diimplementasikan secara praktis di lingkungan operasional, seperti fasilitas kesehatan atau aplikasi berbasis teknologi.

Beberapa penelitian sebelumnya telah membandingkan performa ketiga algoritma tersebut dalam berbagai konteks. Penelitian yang dilakukan oleh Yulrio Brianorman dan Sucipto berfokus pada masalah sulitnya memprediksi potensi penerimaan calon mahasiswa baru di sebuah universitas swasta. Penelitian ini membandingkan algoritma *Decision Tree*, *Random Forest* dan *Naive Bayes* untuk mengetahui algoritma terbaik dalam memprediksi calon mahasiswa baru. Hasilnya menunjukkan algoritma *Random Forest* berhasil menghasilkan akurasi tertinggi yaitu sebesar 59,2%, diikuti oleh algoritma *Decision Tree* dengan akurasi 59,1%, serta algoritma *Naive Bayes* dengan akurasi 58,1%. *Random Forest*

terbukti lebih efektif, dengan potensi pendaftaran calon mahasiswa diperkirakan antara 72% hingga 78% dari total [5].

Penelitian yang dilakukan oleh Muhammad Bagas Arya Darmawan, Favian Dewanta, dan Sri Astuti membahas permasalahan prediksi banjir di Desa Dayeuhkolot yang disebabkan oleh tingginya curah hujan dan naiknya air sungai. Untuk meminimalkan risiko bencana, penelitian ini membandingkan tiga algoritma klasifikasi. Dari hasil analisis terlihat bahwa algoritma *Random Forest* menunjukkan performa terbaik dengan tingkat akurasi rata-rata akurasi mencapai 99,05%, presisi 97,91%, recall 99,18%, *F1-score* 98%, dan waktu komputasi 0,2561 detik, menjadikannya lebih unggul dibandingkan *Decision Tree* dan *Naive Bayes* [6]. Penelitian oleh Muhammad Yasir dan Robertus Suraji yang membahas tentang masalah analisis sentimen terhadap kebijakan kenaikan biaya haji tahun 2023 melalui komentar di YouTube. Untuk mengetahui sentimen masyarakat, digunakan metode *CRISP-DM* dengan tiga algoritma klasifikasi. Hasilnya menunjukkan akurasi sebesar 90% untuk algoritma *Naive Bayes*, 83% untuk algoritma *Decision Tree*, dan 87% untuk algoritma *Random Forest* [7].

Penelitian – penelitian tersebut menunjukkan bahwa setiap algoritma yang digunakan memiliki keunggulan tersendiri berdasarkan konteks yang diteliti. Namun, sebagian besar penelitian masih berfokus pada bidang pendidikan, prediksi bencana, dan analisis sentimen. Sementara kajian khusus yang membahas masalah kesehatan, terutama dalam hal klasifikasi gangguan tidur masih sangat terbatas. Padahal, gangguan tidur memiliki dampak yang cukup serius bagi kesehatan fisik dan mental. Oleh karena itu, terdapat celah penelitian (*Research*

Gap) yang penting untuk diatasi, yaitu dengan membandingkan performa algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* secara khusus pada klasifikasi gangguan tidur menggunakan dataset kesehatan tidur.

Pada penelitian ini sampel data yang akan dianalisis adalah kasus gangguan tidur. Berdasarkan data *National Institute of Neurological Disorders and Stroke*, tidur merupakan kondisi di mana aktivitas mental dan fisik melambat, kesadaran berkurang, dan interaksi dengan lingkungan menjadi minim. Namun, otak tetap bekerja aktif untuk mendukung proses belajar, membentuk ingatan, serta membersihkan racun. Tidur sangat penting bagi kesehatan tubuh dan pikiran, berpengaruh pada berbagai sistem tubuh seperti metabolisme, imunitas, dan suasana hati [8]. Manusia menghabiskan sekitar 25% hingga 33% dari hidupnya untuk tidur [9].

Gangguan tidur adalah kondisi yang mencakup masalah pada durasi, kualitas, atau pola tidur seseorang [10]. Meskipun telah menunjukkan gejala gangguan tidur pada diri mereka, masih banyak individu yang belum memiliki kesadaran bahwa sebenarnya mereka mengalami gangguan tidur. Terdapat berbagai jenis gangguan tidur, yang paling umum adalah *insomnia*. Dalam data yang digunakan pada penelitian ini, gangguan tidur yang menjadi tolak ukur yaitu tanpa gangguan tidur, *sleep apnea*, dan *insomnia*. *Sleep apnea* merupakan kondisi di mana terjadi penghentian pernapasan secara berulang saat tidur, yang dapat menimbulkan kurangnya pasokan oksigen ke dalam tubuh dan meningkatkan risiko penyakit *kardiovaskular*. Sedangkan *insomnia* merupakan keadaan ketika seseorang mengalami kesulitan untuk tertidur atau tidak mampu mempertahankan

tidur yang lama, yang dapat menyebabkan kelelahan, masalah konsentrasi, dan gangguan suasana hati [11].

Deteksi dini terhadap gangguan tidur sangat penting untuk membantu mencegah berbagai masalah kesehatan serius di kemudian hari, dengan penanganan yang diberikan pada waktu yang tepat, juga dapat membantu penderita menikmati kualitas hidup yang lebih baik dan melakukan aktivitas harian dengan lebih nyaman.

Berdasarkan kondisi yang telah disebutkan diatas, ada beberapa masalah yang harus diperhatikan. Banyak orang belum menyadari bahwa mereka mengalami gangguan tidur. Hasil penelitian terdahulu menunjukkan kinerja algoritma yang berbeda – beda sehingga belum diketahui algoritma mana yang paling efektif untuk digunakan. Dibutuhkan model klasifikasi yang tidak hanya akurat tetapi juga efisien agar bisa diterapkan di layanan kesehatan maupun aplikasi berbasis teknologi.

1.2 Identifikasi Masalah

Dari penjelasan latar belakang permasalahan yang telah disampaikan sebelumnya, masalah-masalah yang ada dapat diidentifikasi yaitu :

- a. Masih banyak orang yang belum menyadari bahwa mereka sebenarnya mengalami gangguan tidur.
- b. Banyaknya algoritma untuk menentukan performa terbaik dalam mengklasifikasi gangguan tidur berdasarkan kriteria evaluasi tertentu membuat bingung dengan algoritma yang harus digunakan.

- c. Algoritma *machine learning* yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes* perlu dibandingkan untuk penggunaan klasifikasi terbaik.
- d. Tantangan yang ditemukan yakni merancang model klasifikasi yang dapat memenuhi kebutuhan praktis, seperti penerapannya di lingkungan layanan kesehatan atau dalam aplikasi berbasis teknologi.

1.3 Rumusan Masalah

Sebagaimana penjelasan dari latar belakang dan identifikasi masalah sebelumnya, rumusan masalah pada penelitian ini adalah : “Bagaimana implementasi perbandingan algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* dalam mengklasifikasikan gangguan tidur?”.

1.4 Batasan Masalah

Penjelasan di atas dapat menjadi dasar untuk merumuskan batasan masalah pada penelitian ini. Adapun batasan masalahnya yaitu :

- a. Penelitian ini memanfaatkan dataset yang diperoleh dari situs web *Kaggle* sebagai sumber data yang digunakan.
- b. Algoritma yang dibandingkan yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes*.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini yaitu :

- a. Menganalisis performa algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* dalam mengklasifikasikan gangguan tidur.
- b. Memberikan rekomendasi algoritma terbaik untuk klasifikasi gangguan tidur dan untuk pembuatan sistem deteksi gangguan tidur.

1.6 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini ialah : Mendukung penyediaan model terbaik untuk mendiagnosis gangguan tidur dan pengembangan aplikasi berbasis teknologi untuk memantau gangguan tidur di rumah sakit atau lembaga kesehatan.

1.7 Metode Penelitian

1.7.1 Jenis Penelitian

Tahapan ini menjadi acuan dalam mengumpulkan data. Dalam penelitian ini menerapkan metode kuantitatif untuk mengumpulkan data, dan menggunakan *Library Research* atau penelitian kepustakaan untuk mengumpulkan informasi yang berkaitan dengan topik penelitian melalui kegiatan membaca, mempelajari, dan menganalisis materi untuk mengidentifikasi pokok permasalahan yang diperoleh dari berbagai sumber di perpustakaan, seperti buku, jurnal, artikel, dan dokumentasi. [12].

1.7.2 Teknik Pengumpulan Data

Data yang dipakai pada penelitian ini yaitu data sekunder yang diperoleh secara online melalui sumber data terbuka (*Open Data Source*) yakni *Platform Kaggle*. *Kaggle* merupakan *Platform Web Crowdsourced* di dunia untuk ilmuwan data dan praktisi *machine learning* [13]. *Kaggle* menawarkan berbagai dataset yang diunggah oleh komunitas *data science* dari seluruh dunia [14].

1.8 Sistematika Pembahasan

Sistematika pembahasan yang digunakan yaitu sebagai berikut :

Bab I : Pendahuluan

Bab ini menjelaskan tentang latar belakang, identifikasi masalah, perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, serta sistematika pembahasan terkait objek yang menjadi fokus penelitian.

Bab II : Tinjauan Pustaka

Bab ini membahas tinjauan pustaka yang menjadi dasar dan pendukung bagi penelitian ini. Tinjauan pustaka mencakup berbagai referensi, seperti buku, jurnal, skripsi, dan sumber lainnya yang berperan sebagai dasar dalam mendukung penelitian terhadap objek yang dikaji. Beberapa komponen utama yang dibahas dalam bab ini meliputi penelitian terdahulu, landasan teori, pemodelan, serta perangkat lunak yang dipakai.

Bab III : Analisis dan Perancangan Sistem

Bab ini memaparkan alur proses dan desain sistem dari objek penelitian. Gambaran umum memberikan penjelasan tentang kondisi sistem yang sedang berjalan, termasuk kelebihan dan kekurangannya. Alur proses mencakup identifikasi dan analisis proses bisnis, serta identifikasi dan analisis alternatif solusi. Sementara itu, desain sistem membahas perancangan sistem yang akan dikembangkan, meliputi aspek input dan output yang dihasilkan, hingga visualisasi desain antarmuka sistem yang akan dirancang.

Bab IV : Implementasi Sistem

Bab ini membahas konstruksi sistem yang mencakup kebutuhan sistem, proses instalasi, dan segmen program. Selain itu, bab ini juga menguraikan skenario pengujian, tahapan pengujian yang menjelaskan cara kerja dan hasil

pengujian dari sistem yang dikembangkan, serta aspek pemeliharaan (*maintenance*).

Bab V : Penutup

Bab ini menyajikan ringkasan hasil penelitian terkait objek yang diteliti, serta memberikan rekomendasi untuk pengembangan selanjutnya dari program yang telah dibuat.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Prediction Of Prospective New Students Using Decision Tree, Random Forest and Naive Bayes. Penelitian ini bertujuan untuk mengevaluasi efektivitas metode klasifikasi dalam memprediksi potensi penerimaan calon mahasiswa baru di sebuah universitas swasta di Pontianak. Data penerimaan mahasiswa baru tahun 2020 digunakan dalam prediksi pada penelitian ini. Penelitian ini melakukan perbandingan dari tiga metode klasifikasi yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes*, untuk menentukan model yang paling akurat dalam memprediksi pendaftaran mahasiswa. Analisis data pada penelitian ini menggunakan metode *CRISP-DM*, yang terdiri dari enam tahapan utama yaitu pemahaman bisnis (*Business Understanding*), pemahaman data (*Data Understanding*), persiapan data (*Data Preparation*), pemodelan (*Modelling*), evaluasi (*Evaluation*), serta penerapan (*Deployment*).

Hasil analisis memperlihatkan bahwa algoritma *Random Forest* menghasilkan tingkat akurasi tertinggi sebesar 59,2%, *Decision Tree* dengan akurasi 59,1%, dan *Naive Bayes* sebesar 58,1%. Algoritma *Random Forest* terbukti memiliki performa lebih unggul dan lebih efektif dalam memprediksi pendaftaran mahasiswa baru. Hasil analisis menunjukkan bahwa perkiraan potensi pendaftaran calon mahasiswa berada pada rentang 72% hingga 78% dari total keseluruhan calon mahasiswa yang ada [5].

Analisis Perbandingan Algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* untuk Prediksi Banjir di Desa Dayeuhkolot. Penelitian ini bertujuan untuk memperkirakan kemungkinan adanya bencana banjir atau tidak di Desa Dayeuhkolot menggunakan algoritma *machine learning* yaitu *Decision Tree*, *Random Forest* dan *Naive Bayes*. Data untuk penelitian ini diperoleh dari Balai Besar Wilayah Sungai (BBWS) Citarum, dengan parameter berupa intensitas curah hujan dan tinggi muka air sungai selama periode 2015-2018. Penelitian ini mengevaluasi kinerja model dengan membagi dataset menjadi data pelatihan dan pengujian dalam tiga rasio berbeda (8:2, 7:3, dan 6:4), serta melakukan pengukuran terhadap *accuracy*, *precision*, *recall*, *F1-score*, dan waktu komputasi pada setiap model.

Analisis menunjukkan bahwa algoritma *Random Forest* memiliki kinerja paling unggul dengan rata-rata *accuracy* sebesar 99,05%, *precision* 97,91%, *recall* 99,18%, dan *F1-score* 98%. Algoritma ini juga mencatat waktu komputasi rata-rata 0,2561 detik. *Random Forest* terbukti lebih efisien dibandingkan algoritma *Decision Tree* dan *Naive Bayes*, terutama pada pengujian dengan variasi rasio data yang digunakan [6].

Perbandingan Metode Klasifikasi *Naive Bayes*, *Decision Tree*, *Random Forest* Terhadap Analisis Sentimen Kenaikan Biaya Haji 2023 Pada Media Sosial *Youtube*. Penelitian ini bertujuan untuk menganalisis sentimen dengan mengklasifikasikan komentar positif dan negatif di *platform* media sosial YouTube terkait kebijakan peningkatan biaya perjalanan haji tahun 1444 H/2023

M. Data yang digunakan dalam penelitian ini sebanyak 1.014 komentar yang didapat dari proses *crawling* data di salah satu channel berita nasional Metro TV.

Penelitian ini menggunakan metode *CRISP-DM* dalam proses analisis data, dan menghasilkan akurasi sebesar 90% untuk *Naive Bayes*, 83% untuk *Decision Tree*, dan 87% untuk *Random Forest* [7].

Perbandingan Algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes* Pada Prediksi Penilaian Kepuasan Penumpang Maskapai Pesawat Menggunakan Dataset *Kaggle*. Penelitian ini bertujuan untuk mengukur kinerja tiga algoritma *machine learning* yaitu *Decision Tree*, *Random Forest* dan *Naive Bayes* dalam memprediksi penilaian kepuasan penumpang maskapai pesawat. Dataset yang dianalisis dalam penelitian ini adalah *Airline Customer Satisfaction*, yang diambil dari *Platform Kaggle* dengan total baris data sebanyak 129.880 data dan 23 feature. Penelitian ini menerapkan tahapan *CRISP-DM* yang mencakup *Data Understanding*, *Data Preparation*, *Modelling* dan *Evaluation*.

Data tersebut diproses melalui tahap *data cleaning* untuk menghilangkan *noise* dan *outlier*, kemudian dilakukan pembagian data (*data splitting*) dengan rasio 70:30 untuk data pelatihan dan pengujian. Selanjutnya, ketiga algoritma diuji menggunakan *confusion matrix* dengan parameter akurasi sebagai tolak ukur. Hasil analisis menunjukkan bahwa algoritma *Random Forest* mencapai akurasi tertinggi sebesar 95,44%, *Decision Tree* dengan akurasi 93,41%, dan *Naive Bayes* dengan akurasi 82,51%. Berdasarkan hasil analisis tersebut, dapat disimpulkan bahwa algoritma *Random Forest* merupakan algoritma yang paling

efektif dan efektif untuk memprediksi kepuasan penumpang, berkat performa akurasi yang unggul dibandingkan algoritma lainnya [15].

2.2 Landasan Teori

Berikut adalah landasan yang mendukung adanya penelitian ini antara lain:

a. *Data Mining*

Data mining merupakan proses mencari informasi atau pola signifikan dari data yang jumlahnya besar yang tersimpan di dalam database. Dalam literatur ilmiah, data mining disebut juga dengan sebagai *Knowledge Discovery in Database* (KDD). Saat ini, data mining telah mencakup berbagai berbagai jenis penelitian seperti sistem database kompleks, seperti *object oriented database*, *image/spatial database*, *time-series data/temporal database*, *teks mining*, *web mining*, dan *multimedia database* [16].

Dalam penelitian ini, *data mining* digunakan untuk mengklasifikasi gangguan tidur dengan menggunakan algoritma *Decision Tree*, *Random Forest* dan *Naive Bayes*. Proses *data mining* dalam penelitian ini memungkinkan dalam memahami pola gangguan tidur dan dapat digunakan dalam menentukan adanya gangguan tidur atau tidak.

b. Analisis

Analisis merupakan upaya yang dilakukan menggunakan metode tertentu untuk mengkaji sesuatu secara mendalam. Analisis memiliki peran

penting dalam mengobservasi dan memahami suatu hal dengan tujuan memperoleh hasil akhir dari proses pengamatan yang telah dilakukan [17].

c. *Decision Tree*

Menurut buku *Data Mining: Concepts and Techniques*, *Decision Tree* merupakan metode pembelajaran berbasis struktur pohon yang digunakan untuk tugas klasifikasi dan prediksi. Teknik ini membagi dataset menjadi beberapa cabang berdasarkan aturan keputusan yang diperoleh dari atribut-atribut data, hingga mencapai node akhir yang merepresentasikan hasil atau keputusan [18].

Langkah - langkah perhitungan menggunakan algoritma *Decision Tree* khususnya dengan konsep *Entropy* dan *Information Gain*, sebagai berikut:

1. Menyiapkan Dataset:

Pastikan dataset yang akan digunakan sudah lengkap, mencakup atribut dan label kelas yang diperlukan untuk proses klasifikasi.

2. Menghitung Entropi Awal ($E(S)$):

Gunakan rumus entropi untuk menentukan tingkat ketidakpastian dalam dataset secara keseluruhan. Dan kelompokkan data berdasarkan jumlah item yang termasuk dalam setiap kelas.

3. Menganalisis Setiap Atribut (A):

Untuk setiap atribut A , bagi dataset menjadi beberapa subset berdasarkan nilai-nilai unik atribut tersebut. Hitung entropi untuk setiap subset ($E(S_v)$) menggunakan rumus yang sesuai.

4. Menghitung Nilai *Information Gain* (IG):

Berdasarkan entropi awal dan entropi subset, hitung nilai *Information Gain* pada setiap atribut. Ambil atribut dengan *Information Gain* tertinggi untuk dipakai sebagai simpul (node) utama.

5. Memisahkan Dataset:

Bagilah dataset menjadi beberapa bagian sesuai dengan nilai atribut yang telah dipilih sebagai simpul utama. Ulangi langkah ini untuk setiap subset hingga memenuhi salah satu kondisi berikut:

- a) Semua data dalam subset memiliki label kelas yang sama.
- b) Tidak ada atribut lain yang bisa digunakan untuk pemisahan.

6. Menyusun Pohon Keputusan:

Gunakan simpul utama sebagai akar pohon, lalu buat cabang-cabang berdasarkan nilai dari atribut. Proses dilanjutkan secara rekursif hingga pohon selesai terbentuk.

7. Menguji dan Memvalidasi Model:

Gunakan dataset pengujian untuk mengevaluasi model yang telah dibuat. Hitung metrik evaluasi seperti akurasi, presisi, atau recall untuk mengukur performa model.

Rumus dasar perhitungan algoritma *Decision Tree* sebagai berikut :

1. *Entropi* (E)

$$\text{Entropy}(S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (2.1)$$

Dimana :

S = dataset.

p_i = proporsi data dalam kelas i .

n = jumlah kelas.

2. *Information Gain* (IG)

$$\text{Info Gain}(S, A) = \text{Entropy}(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times \text{Entropy}(S_i)$$

(2.2)

Dimana :

A = atribut yang diperiksa.

S_i = subset data S untuk nilai i .

$|S|$ = total data.

$|S_i|/|S|$ = proporsi data pada subset S_i .

d. *Random Forest*

Random Forest merupakan salah satu metode dalam *machine learning* yang membuat keputusan berdasarkan pada sekumpulan pohon keputusan (*Decision Tree*). Algoritma ini terkenal karena dapat menghasilkan tingkat performa akurasi yang lebih optimal dibandingkan dengan banyak model pembelajaran mesin lainnya [19].

Langkah – langkah perhitungan menggunakan algoritma *Random Forest* sebagai berikut :

1. *Bootstrap Sampling*

Lakukan pengambilan beberapa sampel data secara acak dengan penggantian (*with replacement*) dari dataset asli. Sebagai contoh, jika dataset awal memiliki n sampel, maka dapat dibuat m dataset *bootstrap*, di mana m adalah jumlah pohon yang akan dibangun.

Rumusnya :

$$D_i = \{x_j, y_j\}, j \in \{1, 2, \dots, n\} \quad (2.3)$$

Dengan x_j adalah fitur dan y_j adalah label. Dataset *bootstrap* D_i dipilih secara acak dari dataset asli D .

2. Pemilihan Fitur secara Acak

Untuk setiap pohon keputusan, pada setiap node pohon, pilih subset fitur (k) secara acak dari total fitur (p).

Jumlah fitur yang dipilih (k) :

$$k = \sqrt{p} \text{ (untuk klasifikasi)}$$

$$k = \frac{p}{3} \text{ (untuk regresi, aturan empiris)} \quad (2.4)$$

3. Membangun *Decision Tree*

Setiap dataset *bootstrap* dimanfaatkan untuk membangun sebuah pohon keputusan. Pada setiap node pohon, dipilih subset fitur secara acak untuk menentukan fitur terbaik yang dapat mengoptimalkan kriteria pemisahan (*split*).

Kriteria Split (Gini Index - Contoh untuk Klasifikasi):

$$Gini = 1 - \sum_{i=1}^C p_i^2 \quad (2.5)$$

Dimana :

C adalah jumlah kelas.

p_i adalah proporsi sample di kelas ke- i .

Fitur terbaik adalah yang memberikan pengurangan Gini terbesar :

$$\Delta Gini = Gini_{parent} - \sum_j \left(\frac{n_j}{n} \cdot Gini_{child_j} \right) \quad (2.6)$$

Kriteria Split (MSE – Contoh untuk regresi) :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2 \quad (2.7)$$

Fitur terbaik adalah yang meminimalkan MSE.

4. Voting atau *Averaging*

Setelah seluruh pohon terbentuk, hasil prediksi dari setiap pohon keputusan digabungkan :

Untuk klasifikasi gunakan voting mayoritas :

$$\hat{y} = \text{mode}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m) \quad (2.8)$$

Untuk regresi gunakan rata – rata prediksi :

$$\hat{y} = \frac{1}{m} \sum_{i=1}^m \hat{y}_i \quad (2.9)$$

e. *Naive Bayes*

Naive Bayes merupakan metode klasifikasi yang diberlandaskan pada konsep teorema *Bayes*. Algoritma ini memiliki beberapa keunggulan dibandingkan teknik klasifikasi lainnya. Pertama, karakteristik utamanya adalah asumsi independensi yang sangat kuat (atau disebut "*naif*") antara setiap kondisi atau peristiwa. Kedua, model ini memiliki struktur yang sederhana dan mudah untuk dikembangkan. Ketiga, algoritma ini sangat efektif diterapkan pada dataset berukuran besar [20].

Rumus dasar *Naive Bayes* :

$$P(C_k|X) = \frac{P(X|C_k).P(C_k)}{P(X)} \quad (2.10)$$

Dimana :

$P(C_k|X)$: Probabilitas *posterior* bahwa sampel X termasuk ke dalam kelas C_k .

$P(X|C_k)$: *Likelihood*, yaitu probabilitas bahwa data X muncul jika diketahui bahwa kelasnya adalah C_k .

$P(C_k)$: Probabilitas awal (*prior*) bahwa kelasnya adalah C_k tanpa memperhatikan fitur X .

$P(X)$: Probabilitas keseluruhan data X , yaitu faktor normalisasi (konstan untuk semua kelas).

Langkah perhitungan dengan menggunakan metode *Naive Bayes* sebagai berikut :

1. Menghitung *Prior* Probabilitas

$$P(C_k) = \frac{\text{Jumlah sampel di kelas } C_k}{\text{Total jumlah sampel}} \quad (2.11)$$

2. Menghitung *Likelihood*

Jika X memiliki banyak fitur $(x_1, x_2, \dots, x_n)$, dengan asumsi independensi fitur, maka :

$$P(X|C_k) = P(x_1|C_k) \cdot P(x_2|C_k) \cdot \dots \cdot P(x_n|C_k) \quad (2.12)$$

Probabilitas $P(x_i|C_k)$ dihitung berdasarkan jenis data fitur

- a) Diskrit (kategori) menggunakan frekuensi relatif :

$$P(x_i|C_k) = \frac{\text{Jumlah data di kelas } C_k \text{ dengan nilai } x_i}{\text{Total jumlah data di kelas } C_k} \quad (2.13)$$

- b) Kontinu menggunakan distribusi probabilitas (biasanya distribusi *Gaussian/normal*) :

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}\right) \quad (2.14)$$

Dimana :

μ_k : Rata – rata fitur x_i dalam kelas C_k .

σ_k^2 : Varians fitur x_i dalam kelas C_k .

3. Menghitung *Posterior* Probabilitas

a) Gunakan rumus utama

$$P(C_k|X) = \frac{P(X|C_k).P(C_k)}{P(X)} \quad (2.15)$$

b) Karena $P(X)$ adalah konstan untuk semua kelas, seringkali hanya dihitung :

$$P(C_k|X) \propto P(X|C_k).P(C_k) \quad (2.16)$$

4. Menentukan kelas

Pilih kelas C_k dengan probabilitas *posterior* tertinggi :

$$\hat{C} = \underset{C_k}{\operatorname{argmax}} P(C_k|X) \quad (2.17)$$

f. Klasifikasi

Klasifikasi merupakan proses untuk mengelompokkan data ke dalam sejumlah kategori tertentu berdasarkan kesamaan pola dan karakteristik yang dimilikinya. Secara umum, proses ini dimulai dengan menyediakan sekumpulan data sebagai acuan untuk membentuk aturan klasifikasi. Data acuan tersebut dikenal sebagai *training sets* [21].

g. Gangguan Tidur

Gangguan tidur adalah sebuah kondisi yang kerap dialami banyak orang dan ditandai dengan adanya kesulitan dalam menjaga waktu,

kualitas, atau keteraturan tidur. Ini bisa berupa tidur yang terlalu sebentar, terlalu lama, tidak nyenyak, atau bahkan pola tidur yang kacau. Meskipun terdengar sepele, gangguan ini dapat berdampak besar pada keseharian, mulai dari menurunnya energi fisik, kestabilan emosi, hingga kejernihan pikiran. Dalam jangka panjang, tidur yang terganggu bisa benar-benar mengganggu cara seseorang menjalani hidup dan berinteraksi dengan dunia di sekitarnya [10].

h. Website

Website adalah sekumpulan halaman yang dikelola dalam satu domain dan memuat berbagai informasi. Biasanya, sebuah situs web terdiri dari sejumlah halaman yang saling berkaitan dan bisa diakses melalui internet. Koneksi antarhalaman dalam situs web dikenal sebagai *hyperlink*, yang memberikan kemampuan kepada pengguna untuk navigasi antar halaman ke halaman lainnya dengan mudah. Sementara itu, teks yang digunakan sebagai penghubung dalam navigasi antarhalaman disebut *hypertext* [22].

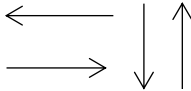
2.3 Pemodelan

a. *Context Diagram*

Diagram Context merupakan bentuk tertinggi dari *Data Flow Diagram* (DFD) yang menunjukkan aliran data masuk dan keluar dari sebuah sistem serta hubungannya dengan entitas luar. Diagram ini berbentuk satu lingkaran yang mewakili keseluruhan proses dalam sistem dan biasanya diberi nomor nol. *Context Diagram* tidak menampilkan

penyimpanan data, sehingga lebih sederhana. Informasi mengenai entitas eksternal dan aliran data diperoleh melalui wawancara dengan pengguna serta analisis dokumen [23].

Tabel 2.1 Simbol – Simbol *Context Diagram*

No	Gambar	Keterangan
1		Kesatuan Luar (Eksternal Entity) = adalah elemen diluar sistem yang bisa berupa individu, organisasi atau sistem lain yang berada di luar lingkungan sistem yang berfungsi memberikan masukan (input) atau menerima keluaran (output) dari system tersebut
2		Proses (Process) = kegiatan atau operasi yang dilakukan oleh perangkat mesin atau komputer, dimana data yang masuk ke dalam proses tersebut diolah untuk menghasilkan data yang akan keluar sebagai output.
3		Arus Data (Data Flow) = pergerakan data yang terjadi antara proses, penyimpanan data, dan entitas dalam sistem. Arus data ini menggambarkan aliran data yang masuk ke dalam proses sistem. menunjukkan arus data dari yang masuk kedalam proses sistem

2.4 Perangkat Lunak yang Digunakan

a. *Google Colab*

Google Colab atau *Google Colaboratory* atau bisa disingkat *Colab* merupakan salah satu produk dari *Google Research*. *Google Colab* merupakan sebuah platform penelitian yang dirancang untuk memudahkan pembuatan *prototype* model *machine learning*. Lingkungan *Jupyter Notebook* tersedia dalam platform ini yang dapat digunakan secara interaktif tanpa perlu menyetel server sendiri [24].

b. *Visual Studio Code*

Visual Studio Code merupakan editor kode yang dibuat dengan desain ringan tetapi tetap memiliki performa yang kuat. Dikembangkan oleh *Microsoft*, editor ini kompatibel dengan berbagai sistem operasi seperti *Windows*, *macOS*, dan *Linux*. Secara default, *Visual Studio Code* mendukung beberapa bahasa pemrograman populer seperti *JavaScript*, *TypeScript*, dan *Node.js*. Selain itu, editor ini juga dapat diperluas kemampuannya untuk bahasa pemrograman lain melalui pemasangan ekstensi yang tersedia di *marketplace* resmi. Beberapa fitur utama yang dimiliki *Visual Studio Code* meliputi *Intellisense* yang membantu melengkapi kode secara otomatis, integrasi langsung dengan sistem kontrol versi *Git*, kemampuan *debugging* yang canggih, serta berbagai ekstensi yang menambah fungsi editor. Update fitur dilakukan secara berkala setiap bulan, menjadikan *Visual Studio Code* unggul dan berbeda dibandingkan editor teks lainnya[25].

c. *Microsoft Excel*

Microsoft Excel adalah suatu aplikasi *spreadsheet* yang menawarkan berbagai fitur canggih untuk mengelola data guna mendukung pengambilan keputusan berbasis informasi. *Excel* mendukung penggunaan berbagai rumus dan fungsi bawaan untuk melakukan perhitungan dan analisis, seperti *SUM*, *IF AVERAGE*, dan lainnya. Karena fleksibilitas dan kemampuannya, *Excel* digunakan sebagai aplikasi yang banyak digunakan di berbagai sektor [26].

d. *Python*

Python adalah bahasa pemrograman tingkat lanjut yang dijalankan secara interpretatif dan mendukung konsep pemrograman berorientasi objek. Bahasa ini dirancang untuk memudahkan pengembangan perangkat lunak dengan menyediakan struktur data tingkat lanjut serta sistem pengetikan dan pengikatan variabel secara dinamis. Keunggulan tersebut menjadikan *Python* sangat sesuai untuk pengembangan aplikasi secara cepat (*rapid application development*) dan juga sebagai bahasa pemrograman skrip (*scripting language*).

Python merupakan bahasa pemrograman yang memiliki tingkat fleksibilitas tinggi karena bisa dijalankan pada berbagai platform sistem operasi, mulai dari *Linux*, *Windows*, dan *macOS*, hingga platform seperti *Java Virtual Machine (JVM)* serta sistem operasi lama seperti *OS/2* dan *Amiga*. Kemampuan ini menunjukkan bahwa *Python* bersifat lintas platform dan mampu beradaptasi dengan beragam lingkungan komputasi.

Selain itu, *Python* juga mendukung berbagai konsep pemrograman, seperti prosedural, berorientasi objek, dan fungsional. Keunggulan lainnya terletak pada fitur manajemen memori otomatis melalui mekanisme *garbage collection*, yang memudahkan pengembang dalam menulis kode tanpa harus menangani proses alokasi dan dealokasi memori secara manual, sehingga pengembangan program menjadi lebih efisien dan terfokus pada logika utama aplikasi [27].



BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Gambaran Umum Objek Penelitian

Tidur adalah kebutuhan yang sangat penting bagi siapapun. Setiap individu kebutuhan tidurnya berbeda – beda, namun tidak banyak dari mereka yang dapat memenuhinya dengan baik yang mengakibatkan terjadinya gangguan tidur. Gangguan tidur merupakan suatu kondisi yang terjadi pada seseorang dimana berkurangnya kualitas tidur yang baik. Kondisi ini mencakup pada durasi, kualitas dan pola tidur seseorang.

Gangguan tidur dapat di ukur melalui beberapa metrik, seperti durasi tidur, kualitas tidur, aktivitas fisik, tingkat stres, dan metrik lainnya. Analisis ini penting untuk mengklasifikasi jenis gangguan tidur yang di alami seseorang, seperti tanpa gangguan tidur, *sleep apnea*, *insomnia*. Pada skripsi ini, menerapkan perbandingan tiga algoritma *machine learning* yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes* untuk mengklasifikasikan gangguan tidur. Hasil klasifikasi akan menentukan algoritma mana yang efektif untuk mengklasifikasikan gangguan tidur, serta membantu dalam pembuatan sistem deteksi gangguan tidur.

3.2 Alur Proses

Alur proses merupakan langkah – langkah atau serangkaian tahapan untuk mencapai sebuah tujuan. Dalam penelitian ini, alur proses merujuk pada bagaimana serangkaian tahapan yang dilakukan dalam sebuah sistem yang sedang dirancang. Alur proses dapat membantu mempermudah dalam pembuatan sebuah

sistem dan mempermudah dalam penyelesaian sebuah masalah dengan memahami tahapan – tahapan apa yang nantinya harus dilakukan.

3.2.1 Identifikasi dan Analisis Proses Bisnis

Identifikasi dan analisis proses bisnis merupakan bagian yang memberikan gambaran tentang proses bisnis yang ada. Tahap ini merupakan bagian penting untuk memahami cara kerja suatu sistem.

a. Identifikasi Proses Bisnis

Identifikasi proses bisnis dilakukan agar objek penelitian teridentifikasi yaitu menjabarkan proses – proses terkait klasifikasi gangguan tidur. Untuk mengklasifikasikan gangguan tidur, beberapa proses dilakukan yaitu :

1. Pengumpulan data
2. *Preprocessing* data
3. Seleksi dan transformasi data
4. *Data Mining* dan evaluasi
5. Visualisasi hasil dan perbandingan

b. Analisis Proses Bisnis

1. Pengumpulan Data

Data sekunder yang diperoleh dari *Platform Kaggle* digunakan pada penelitian ini. *Kaggle* adalah platform web yang menyediakan berbagai dataset dari komunitas data *science* di seluruh dunia yang dapat diakses secara gratis. Dataset pada penelitian ini dapat diakses melalui link berikut: www.kaggle.com .

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
1	1	Male	27	Software Engineer	06.01	6	42	6	Overweight	126/83	77	4200	None
2	2	Male	28	Doctor	06.02	6	60	8	Normal	125/80	75	10000	None
3	3	Male	28	Doctor	06.02	6	60	8	Normal	125/80	75	10000	None
4	4	Male	28	Sales Representative	05.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
5	5	Male	28	Sales Representative	05.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
6	6	Male	28	Software Engineer	05.09	4	30	8	Obese	140/90	85	3000	Insomnia
7	7	Male	29	Teacher	06.03	6	40	7	Obese	140/90	82	3500	Insomnia
8	8	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
9	9	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
10	10	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
11	11	Male	29	Doctor	06.01	6	30	8	Normal	120/80	70	8000	None
12	12	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
13	13	Male	29	Doctor	06.01	6	30	8	Normal	120/80	70	8000	None
14	14	Male	29	Doctor	6	6	30	8	Normal	120/80	70	8000	None
15	15	Male	29	Doctor	6	6	30	8	Normal	120/80	70	8000	None
16	16	Male	29	Doctor	6	6	30	8	Normal	120/80	70	8000	None
17	17	Female	29	Nurse	06.05	5	40	7	Normal Weight	132/87	80	4000	Sleep Apnea
18	18	Male	29	Doctor	6	6	30	8	Normal	120/80	70	8000	Sleep Apnea
19	19	Female	29	Nurse	06.05	5	40	7	Normal Weight	132/87	80	4000	Insomnia
20	20	Male	30	Doctor	07.06	7	75	6	Normal	120/80	70	8000	None
21	21	Male	30	Doctor	07.07	7	75	6	Normal	120/80	70	8000	None
22	22	Male	30	Doctor	07.07	7	75	6	Normal	120/80	70	8000	None
23	23	Male	30	Doctor	07.07	7	75	6	Normal	120/80	70	8000	None

Gambar 3.1 Sleep Health and Lifestyle Dataset

Dataset *Sleep Health and Lifestyle* berisi 374 data tentang kesehatan tidur dan gaya hidup individu, dengan 13 atribut utama seperti *Gender*, *Age*, *Occupation*, *Sleep Duration* (durasi tidur), *Quality of Sleep* (skor kualitas tidur), *Physical Activity Level* (aktivitas fisik), *Stress Level*, *BMI Category*, *Blood Pressure* (tekanan darah), *Heart Rate*, *Daily Steps*, dan *Sleep Disorder* (gangguan tidur).

Pada langkah ini, data dikumpulkan dengan cara mengambil dataset dari platform *Kaggle* melalui *Google Colab*. Tahap awal dimulai dengan proses autentikasi menggunakan file *API Key* dari akun *Kaggle*, kemudian dilanjutkan dengan pemasangan *library* yang diperlukan serta pengunduhan dataset langsung ke dalam *Colab*. Proses tersebut dapat dilakukan dengan menjalankan kode berikut :

Segmen Program 3.1 Pengumpulan Data

```
!kaggle datasets download -d  
uom190346a/sleep-health-and-lifestyle-  
dataset  
!unzip sleep-health-and-lifestyle-  
dataset.zip
```

Kode ini digunakan untuk mengambil dataset yang ada di *Kaggle* yang dijalankan di platform *Google Colab*. Pada perintah *!kaggle datasets download* akan mengunduh dataset *Sleep Health and Lifestyle* pada akun uom190346a di *Kaggle*. Dataset tersebut terunduh dengan format ZIP kemudian di ekstrak dengan menggunakan kode *!unzip* sehingga data tersebut bisa diakses dan digunakan untuk di analisis menggunakan bahasa pemrograman *Python*.

2. *Preprocessing Data*

Tahap *preprocessing* dilakukan untuk melakukan proses pembersihan, pengubahan, dan mempersiapkan data agar layak digunakan dalam proses data mining. Langkah ini mencakup penanganan terhadap nilai yang hilang, menghapus data yang duplikat, normalisasi nilai-nilai data, serta konversi variabel kategorikal ke dalam format numerik agar dapat diproses oleh algoritma data mining [28].

- a) Membersihkan data kosong dan menghapus baris dengan nilai *Sleep Disorder* yang kosong dengan kode berikut :

Segmen Program 3.2 Menghapus Nilai Kosong

```
df =  
pd.read_csv("Sleep_health_and_lifestyle_da  
taset.csv")
```

```
df = df.dropna(subset=['Sleep Disorder'])
```

Membersihkan data yang kosong dan menghapus baris nilai *Sleep Disorder* dilakukan agar analisis dan pembuatan model dilakukan dengan tepat. Dan dikarenakan *Sleep Disorder* merupakan variabel target sehingga jika ada data yang tidak lengkap bisa menyebabkan kekeliruan dan tidak validnya proses analisis.

Kode ini digunakan untuk membersihkan data kosong dan menghapus baris nilai *Sleep Disorder* yang ada pada file CSV yang bernama *Sleep_health_and_lifestyle_dataset.csv*. Data tersebut dibaca kemudian disimpan pada variabel *df* sebagai *DataFrame* dengan bantuan *library pandas*. Bagian data yang kosong dibersihkan dengan menghapus baris dengan nilai kosong (NaN) di kolom *Sleep Disorder*. Sehingga hanya data dengan informasi lengkap yang akan digunakan untuk analisis selanjutnya.

Hasil dari penghapusan nilai kosong pada kolom target *Sleep Disorder* adalah sebagai berikut :

Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	NaN
1	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
3	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea

Gambar 3.2 Hasil Penghapusan Nilai Kosong

Hasil diatas menunjukkan tampilan awal dari dataset *Sleep Health and Lifestyle* yang berisi informasi tentang pola hidup serta kesehatan tidur dari sejumlah individu. Setiap baris mewakili satu individu dengan 13 kolom atribut sebagai tolak ukur dari data ini. Terdapat beberapa baris pada kolom *Sleep Disorder* yang masih menunjukkan nilai kosong (NaN) yang berarti data belum teridentifikasi. Oleh sebab itu, selama tahap pra-pemrosesan data, baris dengan nilai kosong harus dihapus agar analisis bisa dilakukan dengan baik.

- b) Mengisi *missing value* dengan kode sebagai berikut :

Segmen Program 3.3 Mengisi *Missing Value*

```
kolom_numerik =  
df.select_dtypes(include=['int64',  
'float64']).columns  
kolom_kategorikal =  
df.select_dtypes(include=['object']).columns  
for kolom in kolom_numerik:  
    if df[kolom].isnull().sum() > 0:  
        df[kolom] =  
df[kolom].fillna(df[kolom].mean())  
for kolom in kolom_kategorikal:  
    if df[kolom].isnull().sum() > 0:  
        df[kolom] =  
df[kolom].fillna(df[kolom].mode()[0])
```

Untuk mengisi data yang hilang pada dataset, digunakanlah kode ini. Kolom data dibagi menjadi dua, yaitu kolom numerik dan kolom kategorikal. Pada nilai numerik yang hilang akan diisi menggunakan nilai rata – rata dari keseluruhan data pada kolom tersebut. Sedangkan pada kolom kategorikal, akan diisi dengan nilai yang paling sering muncul pada kolom tersebut. Hal ini

dilakukan agar tidak ada data yang hilang yang dapat mengganggu proses analisis.

Hasil dari proses mengisi *missing value* adalah sebagai berikut :

Tabel 3.1 Hasil Mengisi *Missing Value*

<i>Person ID</i>	0
<i>Gemder</i>	0
<i>Age</i>	0
<i>Occupation</i>	0
<i>Sleep Duration</i>	0
<i>Quality of Sleep</i>	0
<i>Physical Activity Level</i>	0
<i>Stress Level</i>	0
<i>BMI Category</i>	0
<i>Blood Pressure</i>	0
<i>Heart Rate</i>	0
<i>Daily Steps</i>	0
<i>Sleep Disorder</i>	0
dtype: int64	

Analisis ini menunjukkan bahwa setiap kolom dalam dataset tidak memiliki data yang kosong atau hilang. Angka 0 pada setiap atribut menunjukkan bahwa setiap baris data telah terisi penuh. Dengan begitu, dataset ini bisa dibilang sudah bersih dan siap digunakan untuk analisis lebih lanjut.

c) *Encoding* kolom kategorikal dengan kode sebagai berikut :

Segmen Program 3.4 *Encoding* Kolom Kategorikal

```
from sklearn.preprocessing import
LabelEncoder
le = LabelEncoder()
for kolom in kolom_kategorikal:
    df[kolom] =
le.fit_transform(df[kolom])
```

Kode ini memanfaatkan *LabelEncoder* dari *sklearn.preprocessing* untuk mengubah data yang berupa kategorikal menjadi data dalam bentuk numerik. Pada prosesnya, *LabelEncoder()* digunakan untuk membuat objek *le*. Dengan menggunakan perulangan *for*, setiap kolom di dalam daftar kolom kategorikal akan diubah menjadi numerik dengan memanfaatkan fungsi *fit_transform()*. Lalu hasilnya akan disimpan didalam kolom yang sama pada *DataFrame* *df*. Dengan adanya proses ini, setiap nilai unik pada kolom akan diubah menjadi angka sehingga data tersebut bisa diproses oleh model.

Dari proses *encoding* pada setiap kolom kategorikal, hasil yang diperoleh adalah sebagai berikut :

Tabel 3.2 Hasil *Encoding* Kolom Kategorikal

	Gender	Occupation	BMI Category	Sleep Disorder
	1	5	2	1
	1	5	2	1
	1	8	2	0
	1	9	2	0
6	0	4	1	1

Tabel diatas menunjukkan data yang telah diproses menggunakan teknik *LabelEncoding*. Data kategorikal diubah menjadi numerik guna mempermudah proses analisis atau ketika melatih model *machine learning*, seperti pada data *Gender*, *Occupation*, *BMI Category*, dan *Sleep Disorder*. Seperti pada hasil dalam tabel diatas, pada kolom *Gender* angka 1 bisa jadi menunjukkan *Male* dan angka 0 menunjukkan *Female* atau bisa sebaliknya. Pada kolom *Occupation* angka 5, 8, 9, atau 4 menunjukkan jenis pekerjaan tertentu. Kolom *BMI Category* dengan angka 2 dan 1 menunjukkan tingkat berat badan. Dan pada kolom *Sleep Disorder* dengan angka 0 dan 1 menunjukkan adanya gangguan tidur atau tidak ada gangguan tidur. Angka – angka tersebut digunakan menyederhanakan data dari kategori aslinya sehingga mengefisiensikan dalam pengolahan datanya.

3. Seleksi dan Transformasi Data

Pada tahap pemrosesan data, khususnya saat seleksi dan transformasi, kolom *Blood Pressure* (tekanan darah) dipecah menjadi dua komponen terpisah, yaitu *Systolic* dan *Diastolic*. Pemisahan ini bertujuan untuk menyederhanakan interpretasi data, sehingga masing-masing jenis tekanan darah dapat dianalisis secara lebih mendalam. Dengan membedakan kedua nilai tersebut, proses analisis menjadi lebih terfokus dan akurat, serta mendukung pengambilan keputusan yang

lebih tepat pada tahap-tahap selanjutnya dalam pengolahan data atau penelitian.

Segmen Program 3.5 Seleksi dan Transformasi Data

```
df[['Systolic', 'Diastolic']] =  
df['Blood Pressure'].str.split('/',  
expand=True).astype(int)  
df.drop(['Blood Pressure', 'Person  
ID'], axis=1, inplace=True)
```

Kode ini bertujuan untuk menangani kolom *Blood Pressure* di *DataFrame* *df*. Fungsi `.str.split('/', expand=True)` digunakan untuk memisahkan nilai sebelum dan sesudah garis miring (“/”) menjadi dua kolom berbeda untuk kolom *Systolic* dan *Diastolic*. Fungsi `.astype(int)` digunakan untuk mengubah format menjadi *integer* pada dua kolom yang telah di pisah. Perintah `df.drop(['Blood Pressure', 'Person ID'], axis=1, inplace=True)` untuk menghapus kolom *Blood Pressure* dan *Person ID* yang ada pada *DataFrame* karena datanya sudah dipisahkan dan ID umumnya tidak relevan dalam analisis model. Melalui proses ini, data *Blood Pressure* lebih mudah untuk di proses karena sudah tersimpan dalam dua kolom angka.

Pada ini menunjukkan hasil sebagai berikut :

Tabel 3.3 Hasil Seleksi dan Transformasi Data

<i>Gender</i>	int64
<i>Age</i>	int64
<i>Occupation</i>	int64
<i>Sleep Duration</i>	float64

<i>Quality of Sleep</i>	int64
<i>Physical Activity Level</i>	int64
<i>Stress Level</i>	int64
<i>BMI Category</i>	int64
<i>Heart Rate</i>	int64
<i>Daily Step</i>	int64
<i>Systolic</i>	int64
<i>Diastolic</i>	int64
<i>dtype: object</i>	

Berdasarkan tabel, menunjukkan bahwa tipe data untuk semua kolom adalah int64 yang berarti semua kolom berisi bilangan bulat yang bisa digunakan pada proses analisis statistik atau perhitungan numerik. Namun pada kolom *Sleep Duration* memiliki tipe data float64 yang menunjukkan bahwa kolom tersebut memiliki bilangan pecahan. Nilai *dtype: object* menunjukkan informasi mengenai tipe data kolomnya bukan isi datanya. Dengan format ini, data bisa digunakan untuk analisis lebih lanjut menggunakan model *machine learning*. Proses pembagian data akan memisahkan dataset menjadi dua komponen utama yaitu data pelatihan (*training*) dan data pengujian (*testing*).

Segmen Program 3.6 Pembagian Data

```
x = df.drop('Sleep Disorder', axis=1)
y = df['Sleep Disorder']
from sklearn.model_selection import
train_test_split
```

```
X_train, X_test, y_train, y_test =  
train_test_split(X, y, test_size=0.3,  
random_state=42)
```

Langkah yang dapat dilakukan sebelum melatih model *machine learning* adalah menyiapkan datanya. Kode ini digunakan untuk memisahkan fitur – fitur yang akan dipakai pada label atau target yang akan diprediksi dan membagi data menjadi data *training* dan data *testing*. Perintah `x = df.drop('Sleep Disorder', axis=1)` akan memilih semua kolom yang ada pada *DataFrame* kecuali kolom *Sleep Disorder* yang menjadi label dari data yang akan diprediksi. Data tersebut kemudian disimpan kedalam variabel *y* menggunakan fungsi `y = df['Sleep Disorder']`. Data akan dibagi menjadi dua dengan menggunakan *X_train* dan *Y_train* untuk data *training* serta *X_test* dan *Y_test* untuk data *testing*. Dengan parameter `test_size=0.3` yaitu sebesar 70% dari total data sebagai data *training*, sedangkan untuk data *testing* sebesar 30%. Untuk menjaga konsistensi hasil pembagian data konsisten selama pengujian, maka digunakan perintah `random_state=42`. Dataset *Sleep Health and Lifestyle* terdiri dari 374 data. Setelah dilakukan *preprocessing*, penghapusan nilai kosong, dan *encoding*, diperoleh 155 data bersih yang siap digunakan. Data kemudian dibagi menjadi 108 data *training* (70%) dan 47 data *testing* (30%).

Hasil pembagian data menunjukkan bahwa ukuran *X_train* : (108,12) dimana data latih memiliki 108 baris data sampel dan 12 kolom

variabel, ukuran X_{test} : (47,12) menunjukkan bahwa data uji memiliki 47 baris dengan 12 kolom variabel, ukuran Y_{train} : (108) menunjukkan bahwa label yang menjadi target untuk data latih sebanyak 108 nilai, dan ukuran Y_{test} : (47) menunjukkan bahwa data uji pada label yang menjadi target sebanyak 47 nilai.

4. *Data Mining* dan Evaluasi

Penelitian ini menerapkan tiga algoritma klasifikasi, yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes*. Ketiga model tersebut dilatih menggunakan data pelatihan, lalu diuji dengan data pengujian. Tujuan dari tahap ini adalah untuk membandingkan seberapa akurat masing-masing model dalam memprediksi tipe gangguan tidur.

a) *Decision Tree*

Segmen Program 3.7 *Decision Tree*

```
from sklearn.tree import
DecisionTreeClassifier
from sklearn.metrics import
accuracy_score, classification_report
model =
DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Akurasi:",
accuracy_score(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test,
y_pred))
```

Model *Machine Learning* dengan algoritma *Decision Tree* diuji, dilatih dan dievaluasi menggunakan kode diatas. Model algoritma *Decision Tree* diimport dengan memanggil perintah *from*

sklearn.tree import *DecisionTreeClassifier*. Sementara alat ukur untuk mengevaluasi performa model diimport menggunakan perintah *from sklearn.metrics import accuracy_score, classification_report*. Selanjutnya, *model = DecisionTreeClassifier(random_state=42)* digunakan untuk membuat sebuah objek model dengan pengaturan acak agar hasil yang didapatkan konsisten. Model ini dilatih menggunakan data *training* menggunakan fungsi *model.fit(X_train, Y_train)* yang kemudian digunakan untuk memprediksi hasil pada data uji melalui *y_pred = model.predict(X_test)*. Untuk mengukur presentase prediksi akurasi pada model memakai fungsi *accuracy_score(Y_test, Y_pred)*. Performa model yang dihasilkan dapat dilihat pada nilai *precision, recall, f1-score, dan support* pada setiap kelas yang menunjukkan kinerja model dalam mengklasifikasi. Hasil dari pengujian algoritma *Decision Tree* adalah sebagai berikut :

Tabel 3.4 Hasil Pengujian *Decision Tree*

<i>Decision Tree</i>				
<i>Accuracy: 0.9148936170212766</i>				
	<i>pr ecision</i>	<i>r ecall</i>	<i>F 1-score</i>	<i>s upport</i>
0	0. 86	1 .00	0. 93	2 5

Tabel 3.4 Lanjutan

1	1. 00	0 .82	0. 90	2 2
---	----------	----------	----------	--------

<i>accuracy</i>			0.91	47
<i>macro avg</i>	0.93	0.91	0.91	47
<i>weighted avg</i>	0.93	0.91	0.91	47

Berdasarkan evaluasi model *Decision Tree* menghasilkan akurasi yang cukup tinggi sebesar 0.91 atau 91% yang menunjukkan bahwa model ini dapat melakukan prediksi dengan benar sekitar 91% dari keseluruhan data yang diuji. Hasil prediksi kelas 0 menunjukkan *precision* sebesar 0.86 atau 86% yang benar benar merupakan kelas 0, *recall* sebesar 1.00 atau 100% yang merupakan seluruh data yang termasuk dalam kelas 0 yang berhasil dikenali oleh model. Sementara untuk kelas 1, *precision* mencapai 1.00 atau 100% yang menunjukkan bahwa seluruh prediksi yang dilakukan untuk kelas 1 adalah benar, meskipun model hanya mampu menemukan *recall* sebesar 0.82 atau 82% dari data kelas 1 yang sesungguhnya. *Macro avg* yang merupakan gabungan dari nilai rata-rata metrik berkisar 0.91 sampai 0.93 dan *weighted avg* yang mempertimbangkan proporsi data setiap kelas berada di sekitar 0.91 sampai 0.93 yang menunjukkan performa model yang sangat baik dan seimbang pada kedua kelas.

Tabel 3.5 *Confusion Matrix Decision Tree*

	Prediksi = 0	Prediksi = 1
Aktual = 0 (negatif)	25 (TN)	0 (FP)
Aktual = 1 (positif)	4 (FN)	18 (TP)

Tabel *confusion matrix* pada *Decision Tree* ini menggambarkan performa model dalam memprediksi gangguan tidur. Hasil analisis menunjukkan model ini dapat mengenali 25 orang tanpa gangguan tidur (*True Negative*) dan 18 orang yang mengalami gangguan tidur (*True Positive*). Model tidak melakukan kesalahan dalam memprediksi orang yang sehat sebagai penderita (*False Positive* = 0), yang menandakan bahwa model memiliki kemampuan yang baik dalam mengidentifikasi kondisi normal. Namun model ini masih memiliki kelemahan yang perlu diperbaiki, yaitu ada 4 orang yang mengalami gangguan tidur yang diprediksi baik – baik saja (*False Negative*). Secara keseluruhan, model ini cukup efisien dalam mendeteksi kondisi normal, walaupun masih perlu ada peningkatan dalam mendeteksi semua kasus gangguan tidur dengan lebih akurat.

Hasil dari *confusion matrix* juga bisa di implementasikan secara visual dengan bentuk *heatmap* dengan menggunakan kode berikut:

Segmen Program 3.8 *Confusion Matrix Heatmap Decision*

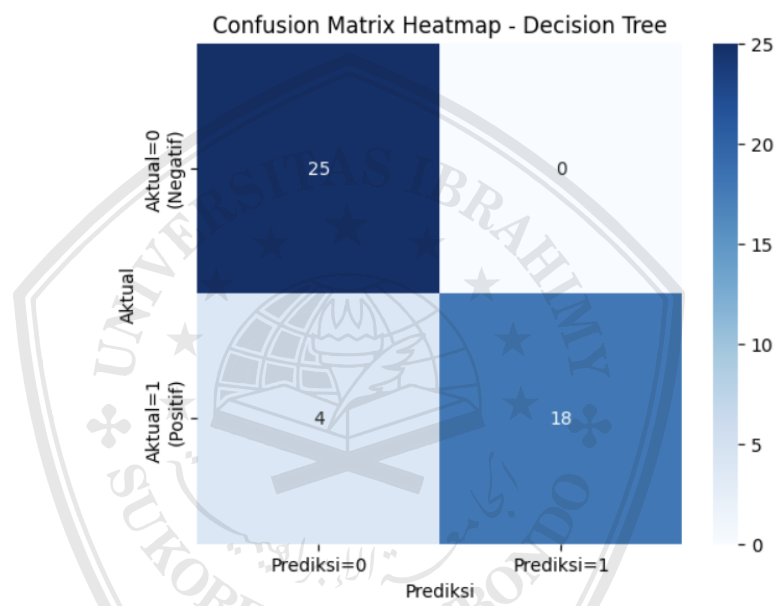
Tree

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import
confusion_matrix
from sklearn.tree import
DecisionTreeClassifier
from sklearn.model_selection import
train_test_split
X = np.random.rand(100, 5)
y = np.random.choice([0, 1], size=100)
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=42)
model =
DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
conf_matrix = np.array([[25, 0],
                        [4, 18]])
labels = ['Negatif (0)', 'Positif (1)']
plt.figure(figsize=(6,5))
sns.heatmap(conf_matrix, annot=True,
fmt='d', cmap='Blues',
xticklabels=['Prediksi=0', 'Prediksi=1'],
yticklabels=['Aktual=0\n(Negatif)',
'Aktual=1\n(Positif)'])
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix Heatmap -
Decision Tree')
plt.show()
```

Kode ini memuat beberapa *library* atau pustaka penting seperti *numpy* untuk pengolahan data, *matplotlib* dan *seaborn* untuk membuat tampilan visual serta modul dari *scikit-learn* untuk merancang dan menilai model *Decision Tree*. Data *dummy* atau buatan dihasilkan secara acak dengan fitur dan label biner, kemudian dibagi menjadi data *training* dan data *testing*. Model *Decision Tree* dilatih dengan menggunakan data latih dan

menghasilkan prediksi pada data uji. Selanjutnya, *confusion matrix* divisualisasikan dalam format *heatmap* menggunakan *seaborn* agar memudahkan dalam memahami kinerja model klasifikasi dengan label yang jelas pada sumbu prediksi dan aktual.

Hasil yang didapatkan melalui kode tersebut seperti pada gambar dibawah ini:



Gambar 3.3 Confusion Matrix Heatmap Decision Tree

Untuk perhitungan menggunakan rumus pada algoritma *Decision*

Tree adalah sebagai berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{18 + 25}{18 + 25 + 0 + 4} = \frac{43}{47} \approx 0.9149$$

$$Precision = \frac{TP}{TP + FP} = \frac{18}{18 + 0} = \frac{18}{18} \approx 1.00$$

$$Recall = \frac{TP}{TP + FN} = \frac{18}{18 + 4} = \frac{18}{22} \approx 0.8182$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{1.00 \cdot 0.8182}{1.00 + 0.8182}$$

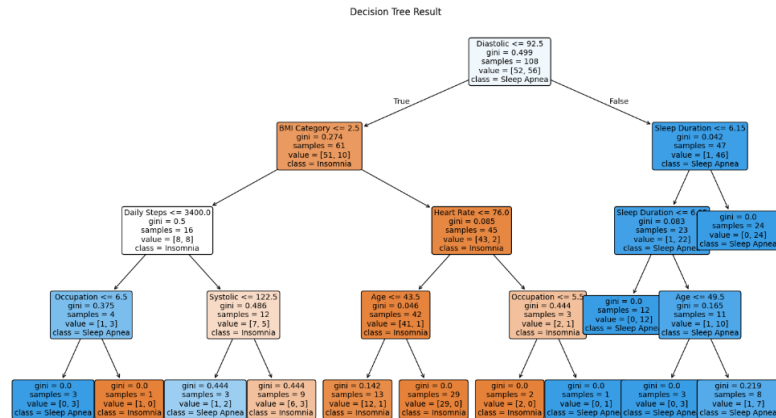
$$= \frac{1.6364}{1.8182} \approx 0.900$$

Hasil dari *Decision Tree* sebagai berikut :

Segmen Program 3.9 *Decision Tree Result*

```
plt.figure(figsize=(18,10))
plot_tree(
    dt_model,
    filled=True,
    feature_names=X_train.columns,
    class_names=[str(c) for c in
dt_model.classes_],
    rounded=True,
    fontsize=10
)
plt.title("Decision Tree Result")
plt.show()
```

Kode ini digunakan untuk memvisualisasikan hasil model *Decision Tree* dalam bentuk diagram pohon. Fungsi *plot_tree()* menampilkan variabel yang digunakan sebagai percabangan, label kelas pada setiap *node*, serta pewarnaan berdasarkan kelas dominan sehingga struktur keputusan lebih mudah dipahami. Dengan visualisasi ini, peneliti dapat melihat alur klasifikasi, variabel yang berpengaruh, serta distribusi data pada tiap *node* secara lebih jelas.



Gambar 3.4 Decision Tree Result

Hasil analisis menggunakan algoritma *Decision Tree* pada Gambar X memperlihatkan bahwa tekanan darah diastolik menjadi variabel paling dominan dalam memisahkan kategori gangguan tidur. Individu dengan nilai *diastolik* ≤ 92.5 lebih sering digolongkan mengalami insomnia, sedangkan mereka dengan nilai *diastolik* yang lebih tinggi cenderung masuk dalam kelompok *sleep apnea*. Selain itu, faktor lain seperti indeks massa tubuh (BMI), lama waktu tidur, detak jantung, serta usia juga berperan dalam memperjelas arah klasifikasi. Sebagai contoh, responden dengan durasi tidur ≤ 6 jam hampir seluruhnya diprediksi menderita *sleep apnea*, sementara responden dengan BMI rendah dan denyut jantung normal lebih sering diperkirakan mengalami *insomnia*. Temuan ini menegaskan bahwa pola hidup dan kondisi kesehatan fisik memiliki keterkaitan erat dengan jenis gangguan tidur yang dialami seseorang.

b) *Random Forest*

Segmen Program 3.10 *Random Forest*

```
from sklearn.ensemble import
RandomForestClassifier
rf_model =
RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
from sklearn.metrics import
classification_report
print(classification_report(y_test,
rf_pred))
```

Algoritma *Random Forest* diimplementasikan menggunakan pustaka *scikit-learn* dengan memanfaatkan kelas *RandomForestClassifier*. Parameter *random_state=42* digunakan agar hasil yang didapatkan konsisten. Selanjutnya untuk memahami pola dari data tersebut, model dilatih menggunakan data *training* (*X_train* dan *Y_train*). Setelah proses pelatihan selesai, model akan dilatih pada data uji (*X_test*) sehingga dihasilkan prediksi (*rf_pred*). Untuk mengevaluasi performa model digunakan fungsi *classification_report* dari *scikit-learn* yang melakukan perbandingan terhadap hasil prediksi dengan label sebenarnya (*Y_test*) pada data uji. Proses evaluasi ditunjukkan dengan metrik *accuracy*, *precision*, *recall*, dan *f1-score*, yang menunjukkan tingkat kemampuan model dalam melakukan klasifikasi.

Hasil yang didapat dari pengujian algoritma *Random Forest* seperti pada gambar berikut :

Tabel 3.6 Hasil Pengujian *Random Forest*

<i>Random Forest</i>				
<i>Accuracy: 0.8723404255319149</i>				
	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>support</i>
0	0.85	0.92	0.88	25
1	0.90	0.82	0.86	22
<i>accuracy</i>			0.87	47
<i>macro avg</i>	0.88	0.87	0.87	47
<i>weighted avg</i>	0.87	0.87	0.87	47

Berdasarkan evaluasi, model *Random Forest* menghasilkan akurasi sebesar 0.87 atau 87% yang menunjukkan bahwa model berhasil memprediksi sekitar 87% data yang diuji dengan tepat. Untuk kelas 0, menghasilkan *precision* sebesar 0.85 atau 85% yang memang benar kelas 0, *recall* sebesar 0.92 atau 92% dari semua data kelas 0, dan *f1-score* sebesar 0.88 atau 88%. Sementara untuk kelas 1, menghasilkan *precision* sebesar 0.90 atau 90% dari prediksi kelas 1 adalah benar, *recall* sebesar 0.82 atau 82% dari data kelas 1 yang sebenarnya. Nilai rata – rata makro yang mengukur rata-rata metrik berkisar antara 0.87 – 0.88 dan rata-rata *wighted* nya sekitar 0.87 yang menunjukkan bahwa model bekerja cukup seimbang di kedua kelas.

Berdasarkan output yang telah dihasilkan diatas, selanjutnya kita dapat menyusun *confusion matrix*.

1) Untuk kelas 0

- *Support* atau jumlah data sebenarnya untuk kelas 0 = 25
- *Recall* (0) = 0.92, yang artinya model berhasil mengenali 92% dari data kelas 0.

Maka :

$$\text{True Negative (TN)} = 0.92 \times 25 = 23$$

Ada 23 data pada kelas 0 yang diprediksi benar.

$$\text{False Negative (FN)} = 25 - 23 = 2$$

Pada kelas 0 ada 2 data diprediksi salah sebagai kelas 1.

- *Precision* kelas 0 = 0.85, dari semua prediksi kelas 0,85% adalah benar.

Maka :

$$\frac{TN}{TN + FP} = \frac{23}{23 + FP} = 0.85 \approx 4$$

Sekitar 4 data dalam kelas 1 diprediksi sebagai kelas 0.

2) Untuk kelas 1

- *Support* kelas 1 = 22
- *Recall* = 0.82

$$\text{True Positive (TP)} = 0.82 \times 22 = 18$$

$$\text{False Negative (FN)} = 22 - 18 = 4$$

- *Precision* = 0.90

$$\frac{TP}{TP + FP} = \frac{18}{18 + FP} = 0.90 \approx FP = 2$$

Terdapat 25 data yang termasuk kedalam kategori kelas 0 (tidak mengalami gangguan tidur). Model berhasil mengklasifikasikan sebanyak 23 data sebagai nilai *True Negative* (TN), dan terdapat kekeliruan dalam klasifikasi, dimana ada 2 data yang diprediksi mengalami gangguan tidur yang disebut sebagai *False Positive* (FP). Sementara untuk kelas 1, ada 22 data yang diklasifikasi dengan benar. Dimana terdapat 18 data sebagai *True Positive* (TP) dan 4 data yang salah diklasifikasikan sebagai tidak mengalami masalah tidur atau *False Negative* (FN).

Tabel 3.7 Confusion Matrix Random Forest

	Prediksi = 0	Prediksi = 1
Aktual = 0 (negatif)	23 (TN)	2 (FP)
Aktual = 1 (positif)	4 (FN)	18 (TP)

Hasil *confusion matrix* juga bisa diimplementasikan dalam bentuk *confusion matrix heatmap* sebagai berikut:

Segmen Program 3.11 Confusion Matrix Heatmap Random

Forest

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

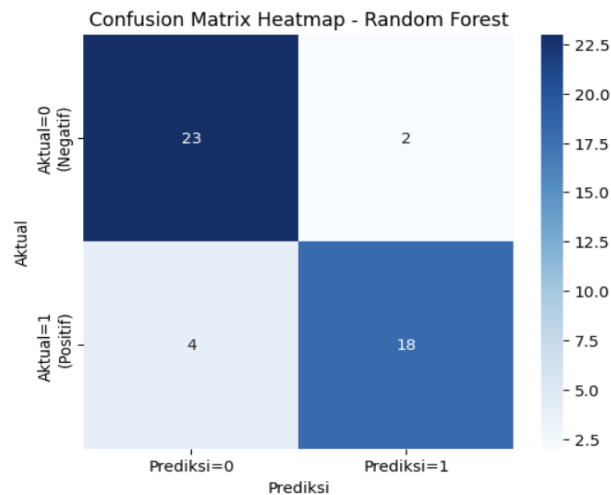
```

from sklearn.metrics import
confusion_matrix
from sklearn.ensemble import
RandomForestClassifier
from sklearn.model_selection import
train_test_split
X = np.random.rand(100, 5)
y = np.random.choice([0, 1], size=100)
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3,
random_state=42)
model =
RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
conf_matrix = np.array([[23, 2],
                        [4, 18]])

labels = ['Negatif (0)', 'Positif (1)']
plt.figure(figsize=(6,5))
sns.heatmap(conf_matrix, annot=True,
fmt='d', cmap='Blues',
xticklabels=['Prediksi=0', 'Prediksi=1'],
yticklabels=['Aktual=0\n(Negatif)',
'Aktual=1\n(Positif)'])
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix Heatmap -
Random Forest')
plt.show()

```

Hasilnya:



Gambar 3.5 Confusion Matrix Heatmap Random Forest

Adapun perhitungan algoritma *Random Forest* menggunakan rumus sebagai berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{18 + 23}{47} = \frac{41}{47} \approx 0.8723$$

$$Precision = \frac{TP}{TP + FP} = \frac{18}{18 + 2} = \frac{18}{20} \approx 0.90$$

$$Recall = \frac{TP}{TP + FN} = \frac{18}{18 + 4} = \frac{18}{22} \approx 0.8182$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0.90 \cdot 0.8182}{0.90 + 0.8182}$$

$$= \frac{1.4724}{1.718} \approx 0.857$$

Hasil yang didapat pada algoritma *Random Forest* berupa 10 fitur teratas dan salah satu pohon dari algoritma *Random Forest* sebagai berikut :

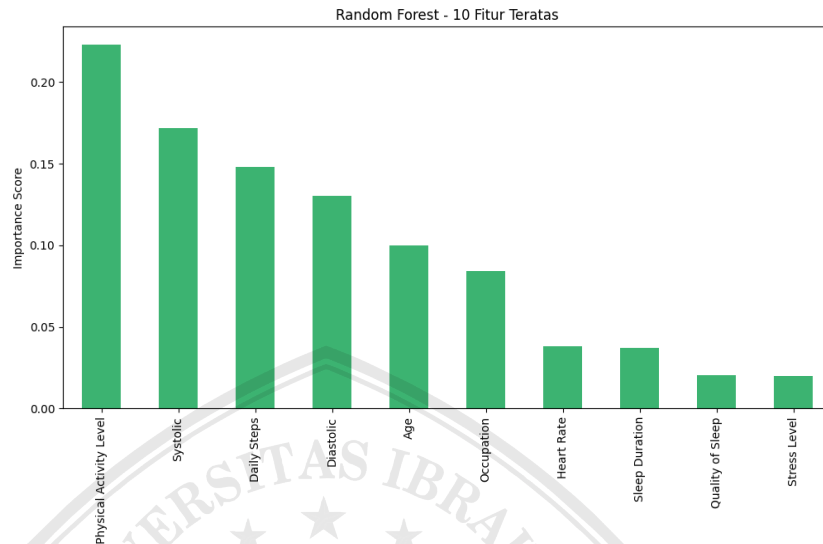
Segmen Program 3.12 10 Fitur Teratas *Random Forest*

```
importances =
pd.Series(rf_model.feature_importances_,
index=X_train.columns)
importances =
importances.sort_values(ascending=False)

plt.figure(figsize=(12,6))
importances.head(10).plot(kind='bar',
color='mediumseagreen')
plt.title("Random Forest - 10 Fitur
Teratas")
plt.ylabel("Importance Score")
plt.show()
```

Kode ini digunakan untuk menampilkan tingkat kepentingan variabel dalam model *Random Forest*, di mana nilai kepentingan dihitung berdasarkan kontribusi masing-masing fitur terhadap keputusan yang dihasilkan oleh seluruh pohon pada model. Informasi tersebut diperoleh melalui atribut *rf_model.feature_importances_*, kemudian dikonversi menjadi objek *pandas*. Series dengan nama variabel sebagai indeks dan diurutkan dari nilai tertinggi hingga terendah menggunakan *sort_values()*. Dari hasil tersebut, sepuluh variabel dengan pengaruh terbesar divisualisasikan dalam bentuk diagram batang, dengan ukuran grafik diatur melalui *plt.figure(figsize=(12,6))* serta diberi warna untuk memperjelas tampilan. Penambahan judul dan label sumbu membantu interpretasi, sehingga visualisasi ini dapat

memudahkan peneliti dalam mengidentifikasi fitur-fitur utama yang berperan penting dalam klasifikasi gangguan tidur



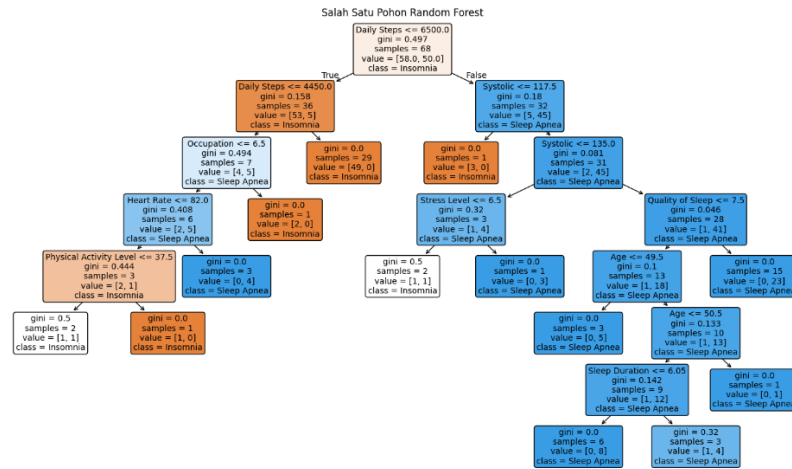
Gambar 3.6 10 Fitur Teratas *Random Forest*

Berdasarkan hasil analisis *Random Forest*, variabel yang paling berpengaruh terhadap klasifikasi gangguan tidur adalah tingkat aktivitas fisik. Faktor ini diikuti oleh tekanan darah *sistolik* dan jumlah langkah harian, yang menunjukkan pentingnya kondisi kardiovaskular serta kebiasaan hidup aktif dalam memengaruhi kualitas tidur. Faktor lain seperti usia, pekerjaan, tekanan darah diastolik, dan detak jantung turut memberikan kontribusi, meskipun dengan tingkat pengaruh yang lebih rendah. Hasil ini menguatkan pemahaman bahwa gangguan tidur tidak hanya dipengaruhi oleh durasi tidur, melainkan juga erat kaitannya dengan kondisi fisik dan pola hidup sehari-hari.

Segmen Program 3.13 Salah Satu Pohon *Random Forest*

```
plt.figure(figsize=(18,10))
plot_tree(
    rf_model.estimators_[0],
    filled=True,
    feature_names=X_train.columns,
    class_names=[str(c) for c in
rf_model.classes_],
    rounded=True,
    fontsize=10
)
plt.title("Salah Satu Pohon Random
Forest")
plt.show()
```

Kode ini berfungsi untuk menampilkan salah satu pohon keputusan dari model *Random Forest* dengan memanggil pohon pertama melalui `rf_model.estimators_[0]`. Visualisasi dilakukan menggunakan `plot_tree()` yang menampilkan variabel pada percabangan, label kelas pada *node*, serta pewarnaan sesuai kelas dominan agar lebih mudah dipahami. Ukuran grafik diperbesar dengan `plt.figure(figsize=(18,10))`, ditambahkan judul, dan hasilnya ditampilkan menggunakan `plt.show()`. Dengan cara ini, peneliti dapat melihat bagaimana sebuah pohon dalam *Random Forest* membuat keputusan, meskipun secara keseluruhan model bekerja dengan menggabungkan banyak pohon sekaligus.



Gambar 3.7 Salah Satu Pohon *Random Forest*

Salah satu pohon dalam model *Random Forest* menunjukkan bahwa jumlah langkah harian merupakan titik awal yang digunakan untuk memisahkan data. Responden dengan aktivitas fisik rendah lebih cenderung mengalami *insomnia*, sementara mereka yang memiliki aktivitas fisik lebih tinggi tetapi disertai tekanan darah *sistolik* yang tinggi serta kualitas tidur rendah lebih sering diklasifikasikan mengalami *sleep apnea*. Faktor tambahan seperti usia, detak jantung, dan durasi tidur semakin memperkuat pemisahan ini. Temuan ini menegaskan bahwa gangguan tidur dipengaruhi oleh kombinasi antara gaya hidup dan indikator kesehatan tubuh.

c) *Naive Bayes*

Segmen Program 3.14 Naive Bayes

```
from sklearn.naive_bayes import
GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
```

Segmen Program 3.14 Lanjutan

```
nb_pred = nb_model.predict(X_test)
from sklearn.metrics import
classification_report
print(classification_report(y_test,
nb_pred))
```

Kode ini menerapkan algoritma *Naive Bayes* dengan menggunakan kelas *GaussianNB* dari pustaka *scikit-learn*. Model *Naive Bayes* dibuat dengan perintah `nb_model = GaussianNB()`. Kemudian model dilatih menggunakan data *training* yaitu *X_train* dan *Y_train* melalui metode `.fit()` agar model bisa mempelajari pola dari data tersebut. Setelah *training* selesai, model diterapkan untuk memprediksi kelas pada data *testing* *X_test* memakai metode `.predict()`. Hasil prediksi ini disimpan dalam variabel *nb_pred*. Untuk mengevaluasi performa model, digunakan fungsi *classification_report* yang membandingkan hasil prediksi *nb_pred* dengan label sebenarnya *Y_test*. Fungsi ini menghasilkan laporan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score* yang menggambarkan tingkat efektifitas model dalam melakukan klasifikasi data.

Hasilnya menunjukkan :

Tabel 3.8 Hasil Pengujian *Naive Bayes*

<i>Naïve Bayes</i>				
<i>Accuracy: 0.7659574468085106</i>				
	<i>pr</i> <i>ecision</i>	<i>r</i> <i>ecall</i>	<i>F</i> <i>l-score</i>	<i>s</i> <i>upport</i>

0	0. 77	0 .80	0. 78	2 5
---	----------	----------	----------	--------

Tabel 3.8 Tabel Lanjutan

1	0. 76	0 .73	0. 74	2 2
<i>accuracy</i>			0. 77	4 7
<i>macro avg</i>	0. 77	0 .76	0. 76	4 7
<i>weighted avg</i>	0. 77	0 .77	0. 77	4 7

Berdasarkan hasil evaluasi, model *Naive Bayes* berhasil mengklasifikasi data dengan akurasi sebesar 0.77 atau 77% dari 47 data yang diuji. Pada kelas 0 kelompok yang tidak mengalami masalah tidur *precision* sebesar 0.77 atau 77% dari 25 data yang diprediksi memang benar berasal dari kelas 0. Untuk *recall* sebesar 0.80 atau 80% yang menunjukkan bahwa model sukses menemukan 80% dari seluruh data kelas 0 dengan tepat. Nilai tengah atau *f1-score* yang didapatkan pada kelas 0 menunjukkan hasil yang seimbang diantara *precision* dan *recall* sebesar 0.78 atau 78%. Sedangkan untuk kelas 1 yang terdiri dari 22 data, *precision* sebesar 0.76 atau 76% data diprediksi tepat termasuk kelas 1, *recall* 0.73 atau 73% data yang mampu dikenali model sebagai data kelas 1, dan *f1-score* sebesar 0.74 atau 74%. Nilai *macro avg* dan *weighted avg* antara 0.76 atau 76% sampai 0.77 atau 77% yang menunjukkan bahwa model cukup seimbang pada kedua kelas.

Dari hasil tersebut maka *confusion matrix* yang diperoleh yaitu:

Tabel 3.9 *Confusion Matrix Naive Bayes*

	Predicted = 0	Predicted = 1
Actual = 0	TN = 20	FP = 5
Actual = 1	FN = 6	TP = 16

Dari tabel diatas, terlihat bahwa dari semua sampel data aktual, pada kelas 0 terdapat 20 data yang berhasil diklasifikasikan dengan benar sebagai data kelas 0 (*True Negative*), dan terdapat 5 data yang diprediksi salah sebagai kelas 1 (*False Positive*). Sedangkan pada kelas 1, sebanyak 16 data berhasil dikenali dengan tepat sebagai data kelas 1 (*True Positive*), dan ada 6 data yang diprediksi salah sebagai kelas 0 (*False Negative*).

Segmen Program 3.15 *Confusion Matrix Heatmap Naive Bayes*

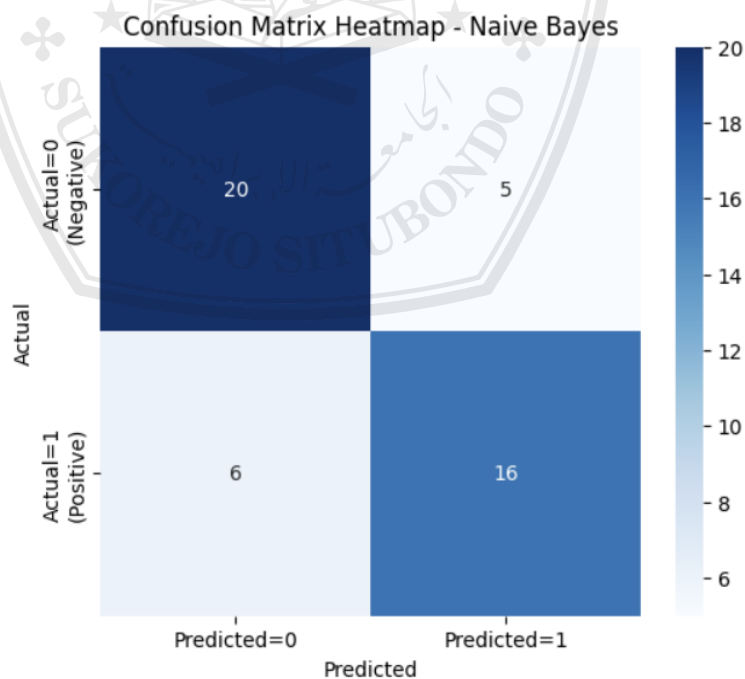
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
X = np.random.rand(100, 5)
y = np.random.choice([0, 1], size=100)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
conf_matrix = np.array([[20, 5],
                        [6, 16]])

labels = ['Negatif (0)', 'Positif (1)']
plt.figure(figsize=(6,5))
```

Segmen Program 3.15 Lanjutan

```
sns.heatmap(conf_matrix, annot=True,
fmt='d', cmap='Blues',
xticklabels=['Prediksi=0', 'Prediksi=1'],
yticklabels=['Aktual=0\n(Negatif)',
'Aktual=1\n(Positif)'])
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix Heatmap -
Naive Bayes')
plt.show()
```



Gambar 3.8 Confusion Matrix Heatmap Naive Bayes

Rumus dasar perhitungan *accuracy*, *precision*, *recall*, dan *f1-score* pada algoritma *Naive Bayes* sebagai berikut :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{16 + 20}{16 + 20 + 5 + 6} = \frac{36}{47} \approx 0.766$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{16}{16 + 5} = \frac{16}{21} \approx 0.762$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{16}{16 + 6} = \frac{16}{22} \approx 0.727$$

$$\begin{aligned} \text{F1 Score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.762 \cdot 0.727}{0.762 + 0.727} \\ &= 2 \cdot \frac{0.554}{1.489} \approx 0.744 \end{aligned}$$

Hasil yang didapat pada algoritma *Naive Bayes* berupa *Probabilitas Prior (Class Prior)* dan *Probabilitas Prediksi (Posterior Probability)* sebagai berikut :

Segmen Program 3.16 *Class Prior & Posterior Probability*

```
print("Probabilitas Prior Kelas
(Naive Bayes):")
for cls, prior in
zip(nb_model.classes_,
nb_model.class_prior_):
    print(f"Kelas {cls}:
{prior:.4f}")

sample = X_test.head(5)
preds = nb_model.predict(sample)
```

```
probs =
nb_model.predict_proba(sample)

df_probs = pd.DataFrame(probs,
columns=[f"P(kelas={cls})" for cls in
nb_model.classes_])
df_probs["Prediksi"] = preds
print("\nProbabilitas Prediksi (5
sampel pertama):")
print(df_probs)
```

Kode ini digunakan untuk menampilkan *probabilitas prior* dan hasil prediksi pada model *Naive Bayes*. Pertama, ditampilkan distribusi awal tiap kelas melalui *nb_model.class_prior_*, yang menunjukkan peluang dasar sebelum mempertimbangkan fitur. Selanjutnya, lima sampel dari data uji diprediksi menggunakan *predict* dan *predict_proba*, sehingga diperoleh probabilitas setiap kelas beserta label akhir yang dipilih model. Hasil tersebut disajikan dalam bentuk tabel, sehingga memudahkan peneliti melihat tingkat keyakinan model dalam menentukan klasifikasi.

Tabel 3. 10 Class Prior & Posterior Probability

Probabilitas Prior Kelas (Naive Bayes) :			
Kelas <i>Insomnia</i>	0.4815		
Kelas <i>Sleep Apnea</i>	0.5185		
Probabilitas Prediksi (Posterior Probability) 5 Sampel Pertama			
	P(kelas= <i>Insomnia</i>)	P(kelas = <i>Sleep Apnea</i>)	Prediksi
0	9.999 943e-01	5.6540 56e-06	<i>Insomnia</i>

1	5.003 594e-16	1.0000 00e+00	<i>Sleep Apnea</i>
2	9.999 992e-01	8.3464 99e-07	<i>Insomnia</i>
3	9.999 992e-01	8.1684 74e-07	<i>Insomnia</i>
4	9.999 773e-01	2.2748 91e-05	<i>Insomnia</i>

Berdasarkan hasil klasifikasi menggunakan algoritma *Naive Bayes*, diperoleh nilai *probabilitas prior* sebesar 48.15% untuk *insomnia* dan 51.85% untuk *sleep apnea*. Distribusi ini menunjukkan bahwa data relatif seimbang antara kedua kategori. Pada lima sampel uji pertama, model mampu memberikan prediksi dengan tingkat keyakinan yang sangat tinggi. Sebagai contoh, sampel kedua diprediksi mengalami *sleep apnea* dengan *probabilitas* 100%, sedangkan sampel lainnya mayoritas diprediksi *insomnia* dengan *probabilitas* mendekati 100%. Hasil ini menunjukkan bahwa *Naive Bayes* mampu membedakan kelas dengan tingkat kepastian yang kuat meskipun hanya menggunakan distribusi *probabilistik* dari fitur yang ada.

5. Visualisasi Hasil dan Perbandingan

Segmen Program 3.17 Visualisasi Hasil dan Perbandingan

```
import matplotlib.pyplot as plt
import seaborn as sns
comparison = pd.DataFrame({
    "Model": ["Decision Tree", "Random
Forest", "Naive Bayes"],
    "Akurasi": [
        accuracy_score(y_test, dt_pred),
        accuracy_score(y_test, rf_pred),
        accuracy_score(y_test, nb_pred)
    ],
    "Presisi": [
```

```

        precision_score(y_test, dt_pred,
average='weighted', zero_division=0),
        precision_score(y_test, rf_pred,
average='weighted', zero_division=0),
        precision_score(y_test, nb_pred,
average='weighted', zero_division=0)
    ],
    "Recall": [
        recall_score(y_test, dt_pred,
average='weighted', zero_division=0),
        recall_score(y_test, rf_pred,
average='weighted', zero_division=0),
        recall_score(y_test, nb_pred,
average='weighted', zero_division=0)
    ],
    "F1 Score": [
        f1_score(y_test, dt_pred,
average='weighted', zero_division=0),
        f1_score(y_test, rf_pred,
average='weighted', zero_division=0),
        f1_score(y_test, nb_pred,
average='weighted', zero_division=0)
    ]

```

Segmen Program 3.17 Lanjutan

```

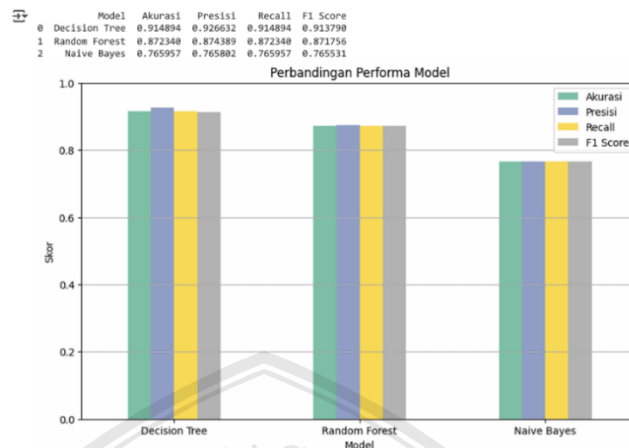
    })
    print(comparison)
    comparison.set_index("Model").plot(kind='bar',
figsize=(10,6), colormap='Set2')
    plt.title("Perbandingan Performa Model")
    plt.ylabel("Skor")
    plt.xticks(rotation=0)
    plt.ylim(0, 1)
    plt.grid(axis='y')
    plt.show()

```

Untuk mengevaluasi dan membandingkan keefektifan tiga model klasifikasi yaitu algoritma *Decision Tree*, *Random Forest*, dan *Naive Bayes*, memakai kode diatas dengan metrik evaluasi utama seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil dari evaluasi tersebut

akan ditampilkan dalam format tabel dan juga grafik yang memudahkan proses analisis. Proses ini diawali dengan mengimport *library matplotlib, Pyplot* dan *seaborn* untuk memvisualisasikan data. Sementara *pandas* digunakan untuk membuat *DataFrame* yang menjadi tempat menyimpan hasil dari evaluasi. *DataFrame comparison* disusun dengan kolom yang menunjukkan model dan nilai metrik yang didapatkan dari fungsi – fungsi yang tersedia di *sklearn.metrics*. Fungsi – fungsi ini mencakup *accuracy_score*, *precision_score*, *recall_score*, dan *f1-score*. Masing – masing fungsi membutuhkan label asli (*Y_test*) dan hasil prediksi dari model (*dt_pred*, *rf_pred*, *nb_pred*). Untuk menghitung metrik dengan metode rata-rata yang mempertimbangkan proporsi masing – masing kelas, digunakanlah parameter *average='weighted'*. Selain itu, parameter *zero_division=0* digunakan untuk menghindari error jika ada kelas yang tidak memiliki prediksi. Untuk membuat grafik yang membandingkan skor dari setiap model, menggunakan fungsi *.plot(kind='bar')* setelah *DataFrame* dicetak. Ada berbagai pengaturan seperti ukuran grafik, pilihan warna, judul, label sumbu, rotasi label sumbu-x, rentang sumbu-y dari 0 hingga 1, dan garis grid pada sumbu-y agar mudah dibaca. Grafik ini akan ditampilkan dengan *plt.show()* yang dapat memudahkan dalam memahami performa dari ketiga algoritma dengan jelas.

Output yang dihasilkan dari proses evaluasi hasil dan perbandingan sebagaimana gambar berikut :



Gambar 3.9 Output Evaluasi Hasil dan Perbandingan

Gambar diatas menunjukkan hasil dari performa tiga algoritma yang di evaluasi yaitu *Decision Tree*, *Random Forest*, dan *Naive Bayes* yang diukur dengan mempertimbangkan empat metrik evaluasi yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil yang didapatkan menunjukkan bahwa algoritma *Decision Tree* dengan struktur algoritma berupa pohon keputusan membuatnya menjadi fleksibel dalam mengolah data sehingga mampu menghasilkan nilai tertinggi dengan *accuracy* sebesar 0.914898, *precision* sebesar 0.926632, *recall* sebesar 0.914894, dan *f1-score* sebesar 0.913790. Algoritma *Random Forest* yang dikenal sebagai algoritma yang stabil dalam menghasilkan prediksi, namun pada penelitian ini berada pada urutan kedua setelah algoritma *Decicion Tree* dengan perolehan nilai yang tak jauh berbeda yaitu *accuracy* sebesar 0.872340, *precision* sebesar 0.874389, *recall* sebesar 0.872340, dan *f1-score* sebesar 0.871756. Hal ini bisa

disebabkan karena jumlah data yang dipakai pada penelitian ini sedikit dan terbatas sehingga menghambat proses analisis secara optimal. Sedangkan algoritma *Naive Bayes* yang dikenal sebagai algoritma yang cepat dan sederhana, justru menunjukkan performa paling rendah diantara dua algoritma lainnya yaitu *accuracy* sebesar 0.765957, *precision* sebesar 0.765802, *recall* sebesar 0.765957, dan *f1-score* sebesar 0.765531.

Tabel 3. 11 Ringkasan Perbandingan

Model	Akurasi	Presisi	Recall	F1 Score
Decision Tree	0.9149	0.9266	0.9149	0.9138
Random Forest	0.8723	0.8743	0.8723	0.8717
Naive Bayes	0.7659	0.7658	0.7659	0.7655

3.2.2 Identifikasi dan Analisis Kebutuhan

a. Identifikasi Kebutuhan Fungsional

Kebutuhan fungsional merupakan hal penting yang dibutuhkan agar penelitian dapat berjalan dengan baik. Identifikasi kebutuhan fungsional merupakan langkah untuk menemukan dan menjelaskan fungsi – fungsi serta layanan yang harus ada dalam sebuah sistem dengan tujuan agar sistem tersebut berjalan sesuai keinginan[29]. Identifikasi kebutuhan fungsional pada penelitian ini yaitu :

1. Sistem membutuhkan kumpulan data tentang gangguan tidur yaitu *Sleep Health and Lifestyle Dataset* yang mencakup informasi terkait

gangguan tidur atau atribut – atribut seperti *Gender*, *Age*, *Occupation*, *Sleep Duration* (durasi tidur), *Quality of Sleep* (skor kualitas tidur) dan atribut lainnya yang didapatkan melalui *open source data* yakni *Kaggle*.

2. Sistem harus melakukan pengolahan data seperti membersihkan data (*preprocessing data*), transformasi data, serta pembagian data menjadi data *testing* dan data *training* untuk memastikan bahwa data yang digunakan dalam proses klasifikasi berkualitas.
3. Sistem mengimplementasikan algoritma klasifikasi dalam proses klasifikasi.
4. Sistem harus memiliki kemampuan mengukur efektifitas model klasifikasi dengan menggunakan beberapa metrik evaluasi yakni *accuracy*, *precision*, *recall*, dan *f1-score*.
5. Sistem harus menampilkan hasil evaluasi dalam bentuk visual misalnya *confusion matrix* dan grafik untuk memudahkan dalam analisis dan pemahaman hasil penelitian.

b. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah proses menganalisis kebutuhan dari suatu sistem yang mencakup semua kebutuhan fungsional yang harus disediakan dalam pembuatan sistem. Analisis kebutuhan ini meliputi pendefinisian layanan atau aktivitas yang harus diberikan sistem saat menerima input tertentu atau berada dalam kondisi tertentu sehingga

sistem dapat dipahami dan didesain dengan tepat[30]. Pada tahap ini analisis kebutuhan fungsionalnya yaitu :

1. Sistem mampu menerima input data gangguan tidur dengan format yang sesuai seperti *CSV* atau *Excel*.
2. Sistem secara otomatis melakukan *processing data* pada data yang di inputkan seperti validasi nilai kosong, normalisasi, dan *encoding* pada atribut kategorikal.
3. Sistem menjalankan klasifikasi dengan menggunakan algoritma yang telah sediakan.
4. Sistem menyajikan laporan yang menunjukkan performa dari algoritma berdasarkan metrik evaluasi yang telah ditentukan sebelumnya.

3.3 Desain Sistem

Pada bagian ini dijelaskan pemodelan sistem yang akan dirancang yang terdiri dari desain output, desain input, desain proses, serta identifikasi dan desain *user interface*.

3.3.1 Desain Output

Desain output adalah hasil dari pemrosesan input oleh sebuah sistem yang menjadi hasil akhir yang bisa berupa data atau informasi yang berguna bagi pihak yang membutuhkan[31]. Desain output pada penelitian

ini, meliputi *preview* dari proses *upload dataset*, hasil dari klasifikasi dan grafik hasil klasifikasi.

sleep disorder

DASHBOARD KELOLA USER UPLOAD DATASET DATA KLASIFIKASI GRAFIK LOGOUT

PREVIEW DATA

Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
1	Male	27	Software Engineer	06.01	6	42	6	Overweight	126/83	77	4200	None
2	Male	28	Doctor	06.02	6	60	8	Normal	125/80	75	10000	None
3	Male	28	Doctor	06.02	6	60	8	Normal	125/80	75	10000	None
4	Male	28	Sales Representative	05.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
5	Male	28	Sales Representative	05.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
6	Male	28	Software Engineer	05.09	4	30	8	Obese	140/90	85	3000	Insomnia
7	Male	29	Teacher	06.03	6	40	7	Obese	140/90	82	3500	Insomnia
8	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
9	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
10	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
11	Male	29	Doctor	06.01	6	30	8	Normal	120/80	70	8000	None
12	Male	29	Doctor	07.08	7	75	6	Normal	120/80	70	8000	None
13	Male	29	Doctor	06.01	6	30	8	Normal	120/80	70	8000	None
14	Male	29	Doctor	6	6	30	8	Normal	120/80	70	8000	None

Click to Train | Test

Gambar 3.10 Desain Output *Preview*

sleep disorder

HASIL KLASIFIKASI GANGGUAN TIDUR

DATA INPUT PENGGUNA

Variabel	Nilai

HASIL KLASIFIKASI

Download Print Kembali

Gambar 3.11 Desain Output Hasil Klasifikasi



sleep disorder

[DASHBOARD](#)

[KELOLA USER](#)

[UPLOAD DATASET](#)

[DATA KLASIFIKASI](#)

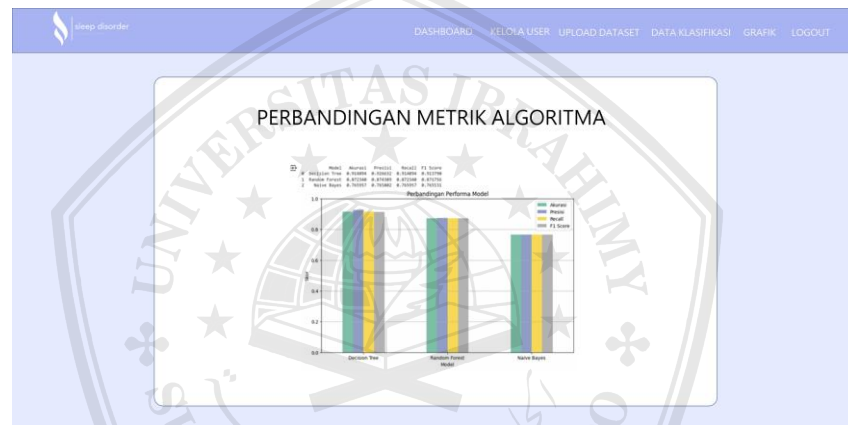
[GRAFIK](#)

[LOGOUT](#)

DATA KLASIFIKASI

ID	USERNAME	NAMA	USIA	JENIS KELAMIN	HASIL	AKSI
----	----------	------	------	---------------	-------	------

Gambar 3.12 Desain Data Klasifikasi



Gambar 3.13 Desain Output Grafik Hasil Perbandingan Model

3.3.2 Desain Input

Desain input pada penelitian ini meliputi pengunggahan data ke dalam sistem. Desain input disini sangat diperlukan untuk mencegah terjadinya kesalahan antara pemikiran dengan hasil yang diperoleh dalam penelitian[32].

Gambar 3.14 Desain Input Dataset

Gambar 3.15 Desain Input Data Klasifikasi

3.3.3 Desain Proses

a. Identifikasi Desain Proses

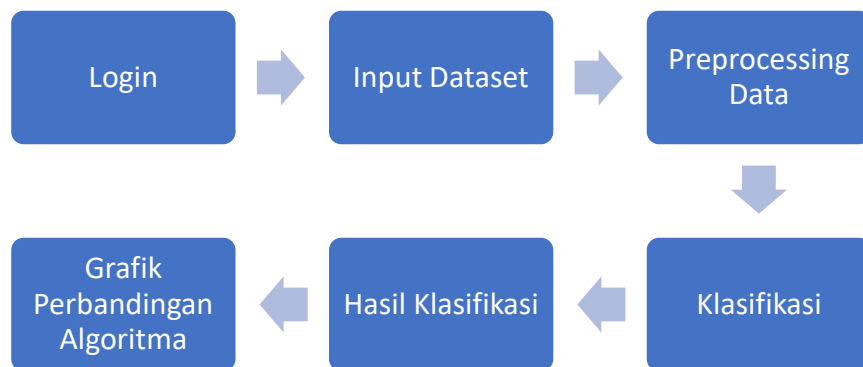
Identifikasi proses merupakan bagian yang menjelaskan proses apa saja yang akan dirancang pada sistem yang dibuat.

Tabel 3.12 Identifikasi Desain Proses

Nama Proses	Deskripsi Proses	Input Proses	Output Proses
Login	Proses awal dalam mengakses sistem dengan menginputkan username dan password	Username, password, dan hak akses	Varabel cookies dan hak akses sistem
Input Dataset	Proses penginputan dataset yang akan dianalisis sistem	Dataset	Dataset tersimpan
<i>Preprocessing Data</i>	Proses persiapan data	Dataset	Data telah diproses
Klasifikasi	Proses klasifikasi data berdasarkan metode yang dipilih	Data hasil preprocessing	Hasil klasifikasi
Hasil Klasifikasi	Proses menampilkan hasil dari klasifikasi	Hasil klasifikasi	Data hasil klasifikasi
Grafik Perbandingan Algoritma	Proses visualisasi hasil perbandingan algoritma dalam bentuk grafik	Data hasil perbandingan algoritma	Grafik hasil perbandingan algoritma

b. Arsitektur Aplikasi

Struktur dari arsitektur aplikasi yang direncanakan dalam penelitian ini, sebagai berikut :

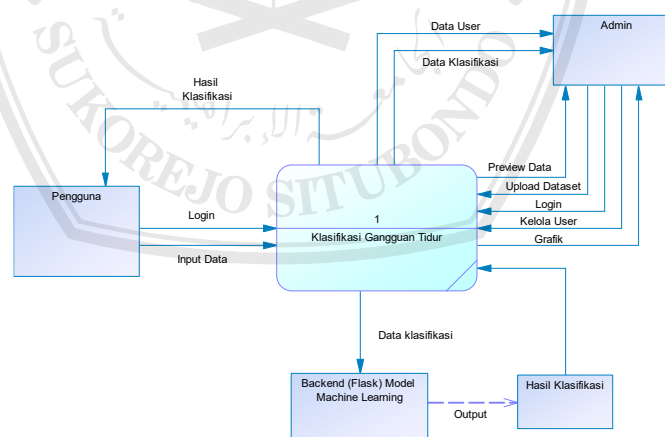


Gambar 3.16 Arsitektur Aplikasi

c. Pemodelan Sistem

1. *Context Diagram*

Pada sistem ini, *context diagram* menampilkan hubungan utama antara pengguna dengan sistem. Berikut ini *context diagram* pada sistem yang akan dibuat :



Gambar 3.17 *Context Diagram*

Dalam diagram ini, pengguna sebagai entitas eksternal yang terhubung ke sistem melalui proses login dan input data yang diperlukan dalam proses klasifikasi. Data yang dimasukkan akan diproses oleh sistem yang kemudian diteruskan ke *Backend (Flask)*

model *machine learning* untuk menjalankan proses klasifikasi menggunakan model yang telah dilatih sebelumnya. Hasil klasifikasi akan dikirim kembali ke sistem utama yang kemudian akan diteruskan ke pengguna beserta dengan visualisasi grafik perbandingan model. Admin memperoleh akses untuk mengetahui data *user*; menginputkan dataset yang akan digunakan dalam proses klasifikasi, mengolah data klasifikasi dari pengguna, dan melihat hasil grafik perbandingan dari algoritma yang digunakan.

3.3.4 Identifikasi dan Desain *User Interface*

a. Identifikasi *User Interface*

Identifikasi *user interface* dilakukan untuk menganalisa *interface* yang akan digunakan, meliputi jenis *interface*, bentuk *interface*, dan deskripsi *interface*.

Nama *Interface* : Klasifikasi Gangguan Tidur

Jenis *Interface* : GUI

Bentuk *Interface* : Gambar

Deskripsi *Interface* : Halaman beranda dan login

b. Desain *User Interface*

Desain *user interface* adalah proses merancang tampilan pengguna pada aplikasi atau sistem yang akan dibuat dengan tujuan menghasilkan antarmuka yang mudah digunakan. Desain *interface* menggabungkan aspek estetika dan fungsionalitas agar pengguna dapat berinteraksi dengan

sistem secara nyaman dan efisien[33]. Desain *interface* menampilkan halaman aplikasi sebagai berikut :

1. Desain *User Interface Home*

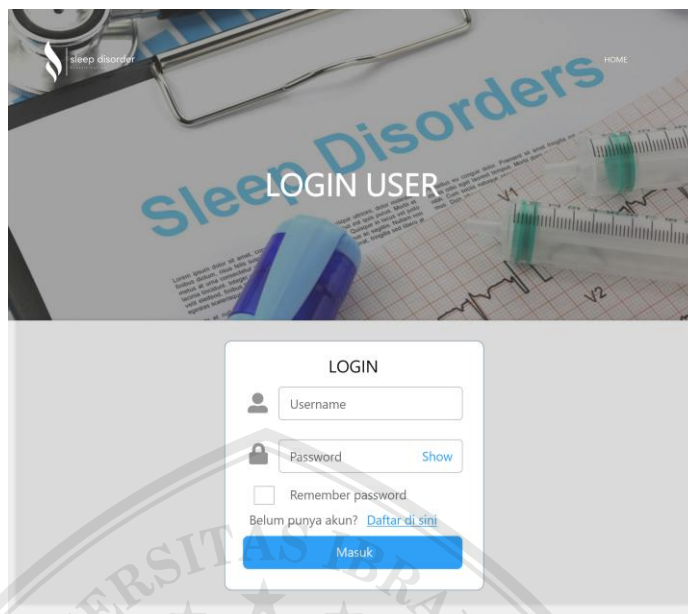


Gambar 3.18 Desain *User Interface Home*

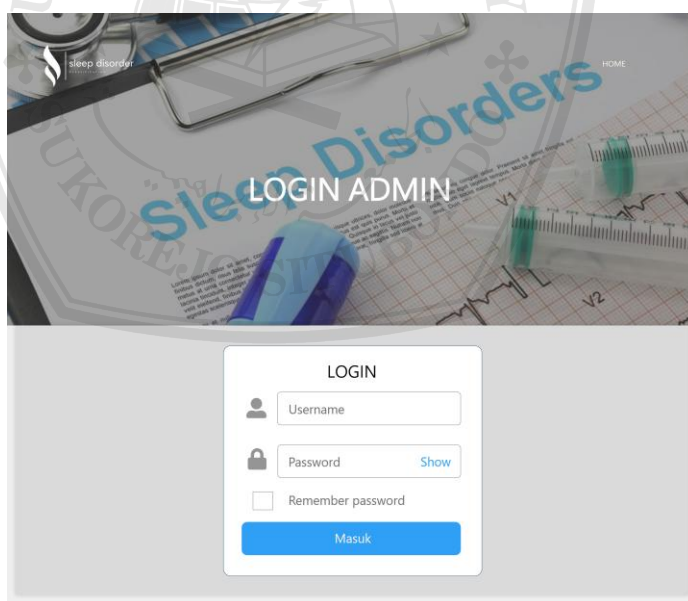


Gambar 3. 19 Desain *Dashboard Admin*

2. Desain *User Interface Login*



Gambar 3.20 Desain *Login User*



Gambar 3.21 Desain *Login Admin*

BAB IV

IMPLEMENTASI SISTEM

4.1 Konstruksi Sistem

Pada bagian ini menjelaskan langkah – langkah dalam proses pembuatan aplikasi dimulai dari kebutuhan sistem, proses instalasi, hingga implementasi segmen program pada penelitian ini yang didefinisikan secara rinci terkait struktur sistem secara keseluruhan. Konstruksi sistem bertujuan untuk menghasilkan sistem yang bisa bermanfaat bagi pengguna dan masyarakat umum[34].

4.1.1 Kebutuhan Sistem

Kebutuhan sistem mengacu pada spesifikasi perangkat, meliputi perangkat lunak maupun perangkat keras yang dibutuhkan dalam pembuatan aplikasi. Aplikasi klasifikasi gangguan tidur ini dikembangkan berbasis website agar dapat diakses dengan mudah. Aplikasi ini memanfaatkan HTML, CSS, dan *JavaScript* di bagian *frontend* nya, dan menggunakan *Phyton (Flask)* di bagian *backend* nya. Perangkat yang dibutuhkan antara lain:

- a. Kebutuhan Perangkat Keras
 1. Komputer/laptop dengan *processor* minimal *Intel Core i3* atau setara
 2. RAM minimal 4 GB
 3. Ruang penyimpanan minimal 10 GB
 4. Koneksi internet untuk mengakses aplikasi dan dataset
- b. Kebutuhan Perangkat Lunak
 1. Sistem operasi seperti Windows, Linux, atau MacOS
 2. Web browser

3. *Phyton*
 4. *Framework* web *Phyton* (*Flask*)
 5. *Library Python* seperti *scikit-learn*, *pandas*, *numpy*, *matplotlib*, dan *imblearn*
 6. Code editor misalnya VS Code
 7. HTML, CSS, dan *JavaScript* untuk pengembangan *frontend*
- c. Kebutuhan Data
1. Dataset gangguan tidur yang diunduh dari *platform Kaggle*

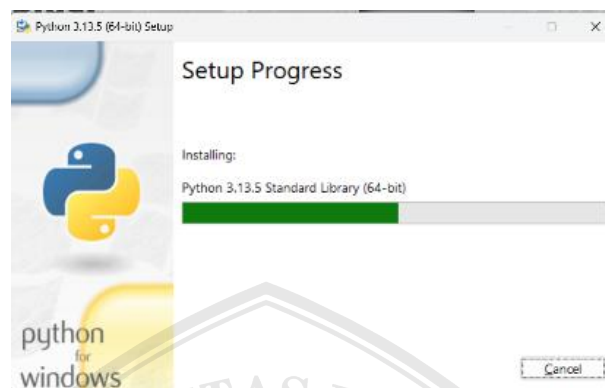
4.1.2 Instalasi Sistem

Proses instalasi sistem adalah serangkaian langkah untuk mengatur dan menyesuaikan perangkat lunak tertentu pada komputer sehingga dapat berjalan dengan optimal sesuai dengan kebutuhan. Pada tahapan ini dilakukan untuk memastikan semua komponen pembuatan sistem telah terpasang dengan benar sehingga sistem yang dibuat akan berjalan dengan efektif dan sesuai keinginan.

a. Instalasi Python

Instalasi Python diperlukan agar pengguna bisa menjalankan dan membuat program menggunakan bahasa pemrograman Python langsung di komputer mereka. Setelah Python terpasang, pengguna dapat menulis, mencoba, dan menjalankan kode tanpa perlu koneksi internet atau perangkat tambahan. Python juga menyediakan banyak pustaka dan modul yang membantu dalam pembuatan aplikasi, baik dalam bentuk yang sederhana maupun yang kompleks. Selain itu, Python sangat populer karena mudah digunakan dan memiliki dukungan dari komunitas yang

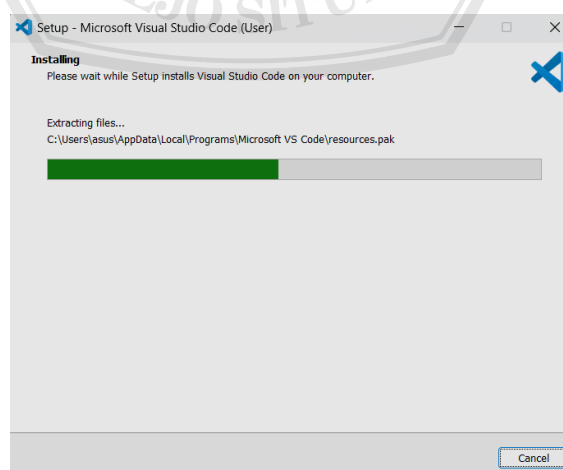
besar, sehingga banyak dipilih untuk belajar dan mengembangkan berbagai jenis perangkat lunak[35]. Berikut adalah proses instalasi python :



Gambar 4.1 Instalasi Python

b. Instalasi *Visual Studio Code*

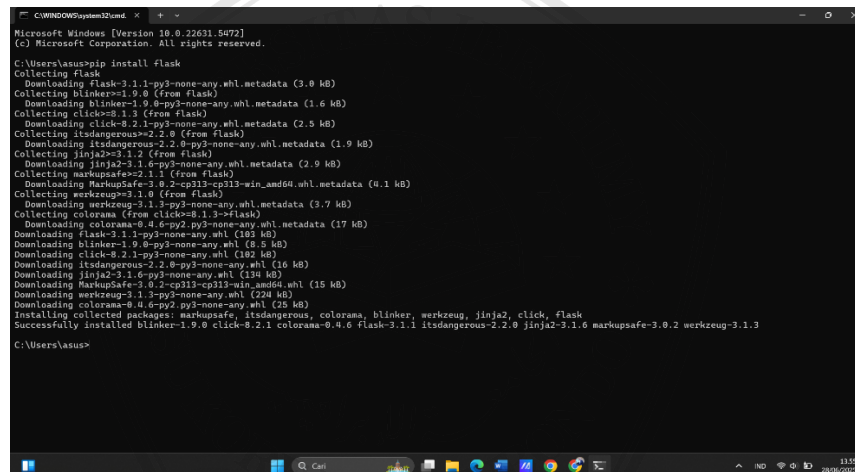
Instalasi *visual studio code* diperlukan agar memudahkan pengguna dalam menulis, mengedit, serta menjalankan kode dari berbagai bahasa pemrograman dalam pembuatan sebuah sistem. Berikut instalasi *visual studio code* :



Gambar 4.2 Instalasi Visual Studio Code

c. Instalasi *Web Framework Flask*

Instalasi *flask* sangat penting dilakukan agar pengguna dapat melakukan pengembangan aplikasi secara lokal pada komputer. *Flask* mempermudah pengguna dalam membuat, menguji dan menjalankan aplikasi web tanpa terhubung ke server eksternal. Karena kemudahan penggunaannya, dokumentasi yang lengkap serta komunitas yang aktif, *flask* banyak digunakan oleh pemula maupun pengembang profesional dalam membangun aplikasi web berbasis *python*[36]. Berikut adalah proses instalasi *flask* :



```

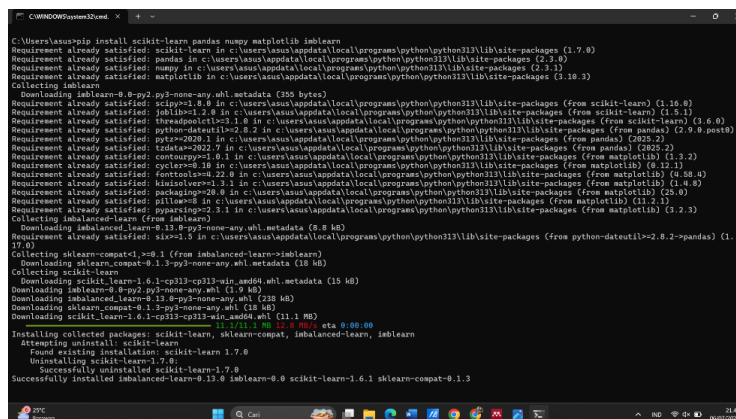
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22631.5472]
(c) Microsoft Corporation. All rights reserved.

C:\Users\asus> pip install flask
Collecting flask
  Downloading flask-3.1.1-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click-8.1.3 (from flask)
  Downloading click-8.1.3-py3-none-any.whl.metadata (2.5 kB)
Collecting itsdangerous-2.2.0 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2-3.1.2 (from flask)
  Downloading jinja2-3.1.2-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe-2.1.1 (from flask)
  Downloading MarkupSafe-2.1.1-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
Collecting werkzeug-3.1.0 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting colorama (from click-8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloading flask-3.1.1-py3-none-any.whl (105 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.1.3-py3-none-any.whl (113 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.2-py3-none-any.whl (134 kB)
Downloading MarkupSafe-2.1.1-cp313-cp313-win_amd64.whl (15 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: markupsafe, itsdangerous, colorama, blinker, werkzeug, jinja2, click, flask
Successfully installed blinker-1.9.0 click-8.1.3 colorama-0.4.6 flask-3.1.1 itsdangerous-2.2.0 jinja2-3.1.2 markupsafe-2.1.1 werkzeug-3.1.3
C:\Users\asus>
  
```

Gambar 4.3 Instalasi *Flask*

d. Instalasi *Scikit-learn, Pandas, Numpy, Matplotlib, Imblearn*

Untuk menggunakan berbagai *library python* dalam pengolahan data dan pembelajaran mesin seperti *scikit-learn, pandas, numpy, matplotlib*, dan *imblearn*, tahap awal yang harus dilakukan adalah menjalankan proses instalasi melalui *package manager pip* pada terminal atau *Command Prompt*. Berikut ini instalasi *scikit-learn, pandas, numpy, matplotlib*, dan *imblearn* :



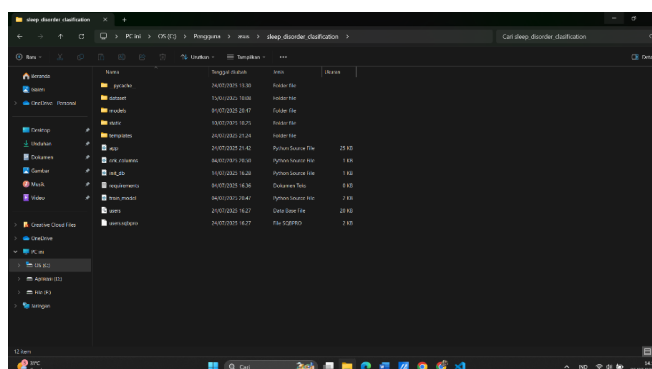
Gambar 4.4 Instalasi Scikit-learn, Pandas, Numpy, Matplotlib,

Imblearn

e. Menjalankan Sistem di Browser

Sebelum mengoperasikan sistem, terlebih dahulu harus memastikan bahwa sistem tersebut dapat dijalankan pada *browser*. Berikut cara untuk menjalankan sistem ini pada *browser* :

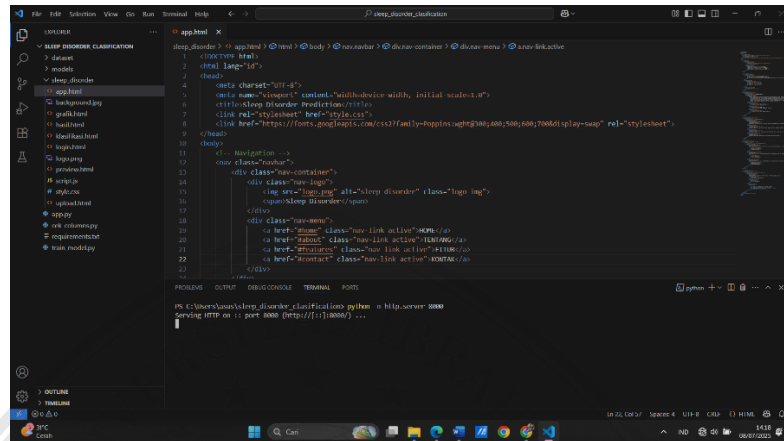
1. File sistem harus tersimpan pada folder yang benar. Agar saat sistem dipanggil pada *localhost*, sistem dapat berjalan. Misalnya file sistem ini terletak pada *drive C:\Users\Asus* sebagaimana gambar dibawah ini :



Gambar 4.5 Folder Sistem

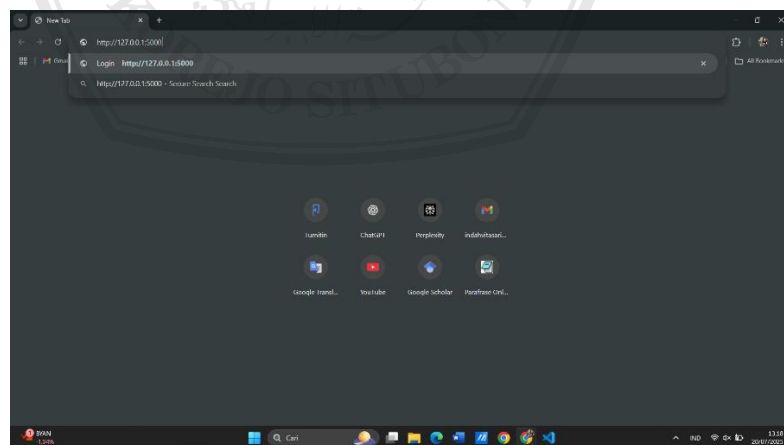
2. Untuk menjalankan sistem, buka *visual studio code* dan buka folder

sistem. Setelah itu ke menu *terminal* klik *new terminal*. Ketikkan `python -m http.server 8000` untuk membuat sistem bisa dijalankan di *browser*. Jika muncul tulisan *Serving HTTP on : : port 8000* ([http://\[::\]:8000/](http://[::]:8000/)), sistem siap dijalankan di *browser*.



Gambar 4.6 Terminal Sistem

3. Buka *browser* dan ketikkan URL sistem yang akan dijalankan yaitu <http://127.0.0.1:5000>.



Gambar 4.7 URL pada Browser

4. Apabila sistem berfungsi dengan benar, maka halaman awal sistem akan muncul sebagaimana gambar berikut :



Gambar 4.8 Halaman Awal Sistem

4.1.3 Segmen Program

Segmen program merupakan suatu bagian atau kumpulan kode dalam sebuah sistem perangkat lunak yang dirancang untuk menjalankan fungsi atau tujuan tertentu secara spesifik. Bagian ini biasanya terdiri dari potongan-potongan kode yang membentuk fitur atau tugas khusus dalam sebuah aplikasi atau sistem secara keseluruhan. Dalam konteks pembahasan ini, akan diuraikan beberapa contoh segmen program yang diambil dari sistem yang sedang dikaji, yang dipilih berdasarkan pentingnya fungsi tersebut menurut penilaian peneliti sebagai representasi utama dari sistem yang dikembangkan.

a. Segmen Home

Segmen home merupakan *source code* untuk menampilkan halaman utama sebelum pengguna melakukan login. Dimana pada halaman ini pengguna dapat melihat informasi mengenai apa itu gangguan tidur, fitur pada aplikasi, jenis gangguan tidur, dan kontak informasi yang bisa dihubungi.

Adapun *source code* untuk halaman home sebagai berikut :

Segmen Program 4.1 Home

1	<!DOCTYPE html>
2	<html lang="id">
3	<head>
4	<meta charset="UTF-8">
5	<meta name="viewport" content="width=device-
6	width, initial-scale=1.0">
7	<title>Sleep Disorder Prediction</title>
8	<link rel="stylesheet" href="{{
9	url_for('static', filename='style.css') }}">
10	<link
11	href="https://fonts.googleapis.com/css2?family=
12	Poppins:wght@300;400;500;600;700&display=swap"
13	rel="stylesheet">
14	</head>
15	<body>
16	<!-- Navigation -->
17	<nav class="navbar">
18	<div class="nav-container">
19	<div class="nav-logo">
20	
23	Sleep Disorder
	</div>
	<div class="nav-menu">
	<a href="{{ url_for('home') }}"
	class="nav-link active">HOME
	<a href="#about" class="nav-link
	active">TENTANG

Segmen Program 4.1 Lanjutan

```

15         <a href="#features" class="nav-link
16         active">FITUR</a>
17         <a href="#contact" class="nav-link
18         active">KONTAK</a>
19         <a href="{{ url_for('login_admin')
20         }}" class="nav-link active">ADMIN</a>
21     </div>
22 </div>
23 </nav>
24 <!-- Hero Section -->
25 <section id="home" class="hero">
26     <div class="hero-content">
27         <div class="hero-text">
28             <h1 class="hero-title">KLASIFIKASI
29             GANGGUAN TIDUR</h1>
30             <p class="hero-subtitle">Menggunakan
31             Machine Learning</p>
32             <button type="button" class="login-
33             btn" onclick="window.location.href='{{
34             url_for('login_user') }}'">Login</button>
35         </div>
36     </div>
37     <div class="hero-bg">
47         <div class="medical-icons">
48             <div class="icon chart-
49             icon">📊</div>

```

Segmen Program 4.1

Lanjutan

```

38         <div class="icon heart-
39 icon">❤️</div>
40         <div class="icon brain-
41 icon">🧠</div>
42         <div class="icon graph-
43 icon">📊</div>
44     </div>
45     <div class="scroll-indicator">
46         <span>Scroll Down</span>
47         <div class="scroll-arrow">↓</div>
48     </div>
49 </section>
50
51 <!-- About Section -->
52 <section id="about" class="about">
53     <div class="container">
54         <h2 class="section-title">GANGGUAN
55 TIDUR</h2>
56
57         <div class="about-content">
58             <div class="about-text">
59                 <p>Menurut data National
60 Institute of Neurological Disorders and Stroke,
61 tidur merupakan kondisi di mana aktivitas
        mental dan fisik melambat,
    
```

Segmen Program 4.1 Lanjutan

63	</div>
64	</section>
65	
66	<!-- About Section -->
67	<section id="about" class="about">
68	<div class="container">
69	<h2 class="section-title">GANGGUAN
70	TIDUR</h2>
71	<div class="about-content">
72	<div class="about-text">
73	<p>Menurut data National
74	Institute of Neurological Disorders and Stroke,
75	tidur merupakan kondisi di mana aktivitas
76	mental dan fisik melambat, kesadaran berkurang,
77	dan interaksi dengan lingkungan menjadi minim.
78	Namun, otak tetap bekerja aktif untuk mendukung
79	proses belajar, membentuk ingatan, serta
80	membersihkan racun. Tidur sangat penting bagi
81	kesehatan tubuh dan pikiran, berpengaruh pada
82	berbagai sistem tubuh seperti metabolisme,
83	imunitas, dan suasana hati. Manusia
84	menghabiskan sekitar 25% hingga 33% dari
85	hidupnya untuk tidur. Gangguan tidur merupakan
86	suatu kondisi yang terjadi pada seseorang
87	dimana berkurangnya kualitas tidur yang baik.
88	Kondisi ini mencakup pada durasi, kualitas dan
	pola tidur seseorang. </p>
	</div>
	<div class="about-stats">
	<div class="stat-item">
	<div class="stat-number">25% -
	30%</div>

Segmen Program 4.1 Lanjutan

```

89         <div class="stat-
90 label">Kebutuhan tidur</div>
91     </div>
92     <div class="stat-item">
93         <div class="stat-
94 number">60M+</div>
95         <div class="stat-label">Orang
96 terdampak di dunia</div>
97     </div>
98     <div class="stat-item">
99         <div class="stat-
100 number">3</div>
101         <div class="stat-label">Jenis
102 gangguan utama</div>
103     </div>
104 </div>
105 </section>
106 <!-- Features Section -->
107 <section id="features" class="features">
108     <div class="container">
109         <h2 class="section-title">FITUR
110 APLIKASI</h2>
111         <div class="features-grid">
112             <div class="feature-card">
113                 <div class="feature-
114 icon">🚗</div>

```

Segmen Program 4.1 Lanjutan

```

115 <div class="container">
116     <h2 class="section-title">FITUR
117     APLIKASI</h2>
118     <div class="features-grid">
119         <div class="feature-card">
120             <div class="feature-
121             icon">🧠</div>
122             <h3>Machine Learning</h3>
123             <p>Menggunakan algoritma Decision
124             Tree, Random Forest, dan Naive Bayes untuk
125             klasifikasi akurat</p>
126             </div>
127             <div class="feature-card">
128                 <div class="feature-
129                 icon">📊</div>
130                 <h3>Analisis Data</h3>
131                 <p>Upload dan analisis dataset
132                 kesehatan tidur dengan visualisasi yang mudah
133                 dipahami</p>
134                 </div>
135                 <div class="feature-card">
136                     <div class="feature-
137                     icon">🏠</div>
138                     <h3>Diagnosis Cepat</h3>
139                     <p>Klasifikasi gangguan tidur
140                     berdasarkan parameter kesehatan dalam hitungan
                     detik</p>
                     </div>
                     <div class="feature-card">

```

Segmen Program 4.1 Lanjutan

```

141         <div class="feature-
142 icon">📊 </div>
143         <h3>Performa Model</h3>
144         <p>Evaluasi akurasi model dengan
145 confusion matrix dan classification report</p>
146     </div>
147 </div>
148 </div>
149 </section>
150 <!-- Sleep Disorders Info -->
151 <section id="disorders" class="disorders">
152     <div class="container">
153         <h2 class="section-title">JENIS
154 GANGGUAN TIDUR</h2>
155         <div class="disorders-grid">
156             <div class="disorder-card healthy">
157                 <h3>😴 Normal/Healthy</h3>
158                 <p>Pola tidur yang sehat dengan
159 durasi 7-9 jam per malam, kualitas tidur baik,
160 dan tidak ada gangguan berarti.</p>
161                 <ul>
162                     <li>Durasi tidur optimal</li>
163                     <li>Kualitas tidur baik</li>
164                     <li>Tidak ada gangguan</li>
165                 </ul>
166             </div>
167             <div class="disorder-card insomnia">

```

Segmen Program 4.1 Lanjutan

```

167         <h3>😫 Insomnia</h3>
168         <p>Kesulitan untuk tertidur atau
169 mempertahankan tidur, menyebabkan kelelahan dan
gangguan aktivitas sehari-hari.</p>
170         <ul>
171             <li>Sulit tertidur</li>
172             <li>Sering terbangun</li>
173             <li>Tidur tidak nyenyak</li>
174         </ul>
175     </div>
176     <div class="disorder-card apnea">
177         <h3>😴 Sleep Apnea</h3>
178         <p>Gangguan pernapasan saat tidur
179 yang menyebabkan henti napas berulang dan
180 menurunkan kualitas tidur.</p>
181         <ul>
182             <li>Henti napas saat
tidur</li>
183             <li>Mendengkur keras</li>
184             <li>Mengantuk berlebihan</li>
185         </ul>
186     </div>
187 </div>
188 </div>
189 </section>
190
191 <!-- Contact Section -->
192

```

Segmen Program 4.1 Lanjutan


```

193     <section id="contact" class="contact">
194         <div class="container">
195             <h2 class="section-title">KONTAK &
196 INFORMASI</h2>
197             <div class="contact-content">
198                 <div class="contact-info">
199                     <h3>Hubungi Kami</h3>
200                     <p>Untuk informasi lebih lanjut
201 tentang aplikasi klasifikasi gangguan tidur</p>
202                     <div class="contact-item">
203                         <span>✉️</span>
204                         <p>info@sleepclasification.com
205 </p>
206                     </div>
207                     <div class="contact-item">
208                         <span>☎️</span>
209                         <p>+62 812-3456-7890</p>
210                     </div>
211                 </div>
212                 <div class="contact-form">
213                     <h3>Kirim Pesan</h3>
214                     <form>
215                         <input type="text"
216 placeholder="Nama Lengkap" required>
217                         <input type="email"
218 placeholder="Email" required>
219                         <textarea placeholder="Pesan

```

Segmen Program 4.1 Lanjutan

```

218 Anda" rows="4" required></textarea>
219         <button type="submit">Kirim
220 Pesan</button>
221     </form>
222 </div>
223 </div>
224 </div>
225 </section>
226 <script src="{{ url_for('static',
227 filename='script.js') }}"></script>
227 </body>
228 </html>

```

b. Segmen Login

Segmen login berisi *source code* yang memverifikasi pengguna sistem sebelum mereka menggunakan sistem berdasarkan tipe login yang digunakan. Dalam sistem ini, terdapat dua segmen login, yakni login user dan login admin. Berikut segmen login dari user dan admin.

1. Segmen Login User

Segmen Program 4.2 Login User

```

1 <section class="login-section">
2     <div class="login-card">
3         {% with messages =
4 get_flashed_messages(with_categories=true)
5 %}
6         {% if messages %}
7         <script>

```

Segmen Program 4.2 Lanjutan

```

7      {% for category, message in messages
8      %}
9          alert("{{ message }}");
10         {% endfor %}
11     </script>
12     {% endif %}
13     {% endwith %}
14     <h2>LOGIN</h2>
15     <div class="login-container">
16         <form id="loginForm" action="{{
17 url_for('login_user') }}" method="post">
18         <div class="input-group">
19             <span class="input-icon">
20                 <!-- SVG ICON -->
21             </span>
22             <input type="text"
23 name="username" placeholder="Username User"
24 required>
25         </div>
26         <div class="input-group">
27             <span class="input-icon">
28                 <!-- SVG ICON -->
29             </span>
30             <input type="password"
31 name="password" placeholder="Password"
32 id="password" required>
33             <span class="show-pass"
34 onclick="togglePassword()">Show</span>
35         </div>

```

Segmen Program 4.2 Lanjutan

```

33         <div style="display: flex; align-
34 items: center; padding: 8px;">
35             <input type="checkbox"
36 id="rememberMe" style="width: 10%; padding:
37 8px;">
38             <label for="rememberMe">Remember
39 password</label>
40         </div>
41         <p>Belum punya akun? <a href="{{
42 url_for('register') }}">Daftar di sini</a></p>
43         <button type="submit" class="login-
44 btn">Login</button>
45     </form>
46 </div>
47 </section>
48 <script src="{{ url_for('static',
49 filename='script.js') }}"></script>
50 <script>
51     function togglePassword() {
52         const pass =
53         document.getElementById('password');
54         if (pass.type === "password") {
55             pass.type = "text";
56         } else {
57             pass.type = "password";
58         }
59     }
60 </script>

```

2. Segmen Login Admin

Segmen Program 4.3 Login Admin

```

1 <section class="login-section">
2     <div class="login-card">
3         {% with messages =
4         get_flashed_messages() %}
5         {% if messages %}
6             <ul class="flash-messages">
7                 {% for message in messages %}
8                     <li>{{ message }}</li>
9                 {% endfor %}
10            </ul>
11        {% endif %}
12        {% endwith %}
13        <h2>LOGIN </h2>
14        <div class="login-container">
15            <form id="loginForm" action="{{
16            url_for('login_admin') }}" method="post">
17                <div class="input-group">
18                    <span class="input-icon">
19                        <!-- SVG ICON -->
20                    </span>
21                    <input type="text"
22                    name="username" placeholder="Username"
23                    required>
24                </div>
25                <div class="input-group">
                    <span class="input-icon">

```

Segmen Program 4.3 Lanjutan

```

26         <!-- SVG ICON -->
27     </span>
28     <input type="password"
29     name="password" placeholder="Password"
30     id="password" required>
31     <span class="show-pass"
32     onclick="togglePassword()">Show</span>
33 </div>
34 <div style="display: flex; align-
35 items: center; padding: 8px;">
36     <input type="checkbox"
37     id="rememberMe" style="width: 10%; padding:
38     8px;">
39     <label for="rememberMe">Remember
40     password</label>
41 </div>
42 <button type="submit" class="login-
43 btn">Login</button>
44 </form>
45 </div>
46 </div>
47 </section>
48
49 <script src="{{ url_for('static',
50 filename='script.js') }}"></script>
51
52 <script>
53     function togglePassword() {
54         const pass =
55         document.getElementById('password');
56         if (pass.type === "password") {
57             pass.type = "text";

```

Segmen Program 4.3 Lanjutan

```

53         </form>
54     </div>
55 </div>
56 </section>
57
58     <script src="{{ url_for('static',
59 filename='script.js') }}"></script>
60     <script>
61         function togglePassword() {
62             const pass =
63 document.getElementById('password');
64             if (pass.type === "password") {
65                 pass.type = "text";
66             } else {
67                 pass.type = "password";
68             }
69         }
70     </script>
71

```

c. Segmen Register User Baru

Segmen register merupakan *source code* yang menampilkan halaman pendaftaran akun bagi pengguna baru agar bisa masuk kedalam sistem.

Segmen Program 4.4 Register

```

1 <section class="login-section">
2     <div class="login-card">
3         {% with messages =
4         get_flashed_messages(with_categories=true) %}
5         {% if messages %}
6             <script>
7                 {% for category, message in messages
8                 %}
9                     alert("{} message {}");
10                    {% endfor %}
11                </script>
12                {% endif %}
13                {% endwith %}
14                <h2>REGISTER</h2>
15                <div class="login-container">
16                    <form id="register" action="{%
17                    url_for('register') %}" method="post">
18                        <div class="input-group">
19                            <span class="input-icon">
20                                <!-- SVG ICON -->
21                            </span>
22                            <input type="text"
23                            name="username" placeholder="Username"
24                            required>
25                        </div>
26                        <div class="input-group">
27                            <span class="input-icon">
28                                <!-- SVG ICON -->
29                            </span>

```

Segmen Program 4.4 Lanjutan

28	<code><input type="password"</code>
29	<code>name="password" placeholder="Password"</code>
	<code>id="password" required></code>
30	<code><span class="show-pass"</code>
31	<code>onclick="togglePassword()">Show</code>
32	<code></div></code>
33	<code><button type="submit" class="login-</code>
34	<code>btn">Register</button></code>
35	<code><p>Sudah punya akun? <a href="{{</code>
	<code>url_for('login_user') }}">Login di sini</p></code>
36	<code></form></code>
37	<code></div></code>
38	<code></div></code>
39	<code></section></code>

d. Segmen Klasifikasi

Segmen klasifikasi merupakan *source code* yang menampilkan halaman bagi *user* untuk menginputkan data yang dibutuhkan dalam proses klasifikasi.

Segmen Program 4.5 Klasifikasi

1	<code><form method="POST" action="{{</code>
2	<code>url_for('klasifikasi') }}"></code>
3	<code><div class="form-group"></code>
4	<code><label class="form-label">Nama</label></code>
5	<code><input type="text" class="form-</code>
	<code>control" name="nama" required></code>
6	<code></div></code>
7	<code><div class="form-group"></code>

Segmen Program 4.5 Lanjutan

8	<code><label class="form-label">Usia</label></code>
9	<code><input type="number" class="form-</code>
10	<code>control" name="usia" min="4" max="120"</code>
11	<code>required></code>
12	<code></div></code>
13	<code><div class="form-group"></code>
14	<code><label class="form-label">Jenis</code>
15	<code>Kelamin</label></code>
16	<code><select class="form-select"</code>
17	<code>name="jenis_kelamin" required></code>
18	<code><option</code>
19	<code>value="Perempuan">Perempuan</option></code>
20	<code><option value="Laki-laki">Laki-</code>
21	<code>laki</option></code>
22	<code></select></code>
23	<code></div></code>
24	<code><div class="form-group"></code>
25	<code><label class="form-label">Bagaimana</code>
26	<code>kualitas tidur Anda belakangan ini?</label></code>
27	<code><select class="form-select"</code>
28	<code>name="quality_of_sleep" required></code>
29	<code><option value="2">Sangat</code>
30	<code>buruk</option></code>
31	<code><option value="4">Buruk</option></code>
32	<code><option value="6">Cukup</option></code>
	<code><option value="8">Baik</option></code>
	<code><option value="10">Sangat</code>
	<code>baik</option></code>
	<code></select></code>

e. Segmen Hasil Klasifikasi

Segmen hasil klasifikasi merupakan *source code* yang menampilkan halaman bagi *user* untuk hasil klasifikasi dari data yang diinputkan.

Segmen Program 4.6 Hasil Klasifikasi

```

1  <table class="table table-data table-bordered
2  table-hover mb-4">
3      <thead>
4          <tr>
5              <th>Variabel</th>
6              <th>Nilai</th>
7          </tr>
8      </thead>
9      <tbody>
10         {# --- Cari dan tampilkan "Nama" dulu
11         --- #}
12         {% set nama_ditemukan = false %}
13         {% set usia_ditemukan = false %}
14         {% for label, value in data %}
15             {% if label == 'Nama lengkap Anda' %}
16                 <tr>
17                     <td>{{ label }}</td>
18                     <td>{{ value }}</td>
19                 </tr>
20                 {% set nama_ditemukan = true %}
21             {% endif %}
22             {% if label == 'Berapa usia Anda' %}

```

Segmen Program 4.6 Lanjutan

```

23         <td>{{ label }}</td>
24         <td>{{ value }}</td>
25     </tr>
26     {% set usia_ditemukan = true %}
27     {% endif %}
28     {% endfor %}
29     {% # --- Tampilkan sisanya, kecuali
30     "Nama" dan "Usia" --- #}
31     {% for label, value in data %}
32         {% if label not in ['Nama lengkap
33         Anda', 'Berapa usia Anda'] %}
34         <tr>
35             <td>{{ label }}</td>
36             <td>{{ value }}</td>
37         </tr>
38         {% endif %}
39         {% endfor %}
40     </tbody>
41 </table>
42 <div class="mt-4 mb-4">
43     <i class="bi bi-check-circle-fill
44     result-icon"></i>
45     <h3 class="mb-3">Hasil Klasifikasi</h3>
46     <div class="prediction-badge">
47         {{ hasil }}
48     </div>

```

Segmen Program 4.6 Lanjutan

48	</div>
49	<div class="mt-4 pt-3 border-top">
50	<a href="{{
51	url_for('download_hasil_pdf') }}" class="btn
52	btn-success">
53	<i class="bi bi-file-earmark-pdf"></i>
54	Download PDF
55	
56	<button onclick="window.print()"
57	class="btn btn-action btn-print">
58	<i class="bi bi-printer"></i> Cetak
59	Hasil
60	</button>
61	<a href="{{ url_for('klasifikasi') }}"
62	class="btn btn-action btn-back">
63	<i class="bi bi-arrow-left"></i>
64	Kembali
65	
66	</div>
67	</div>
68	</div>
69	<div class="mt-4 text-muted">
70	<small>
71	<i class="bi bi-info-circle"></i> Hasil
72	ini bersifat prediktif dan tidak menggantikan
73	diagnosis medis.
74	</small>
	</div>
	</div>

f. Segmen *Dashboard Admin*

Segmen *dashboard* admin berisi *source code* yang menampilkan halaman utama ketika admin berhasil login ke sistem.

Segmen Program 4.7 *Dashboard Admin*

1	<div class="dashboard-cards">
2	<div class="card stat-card">
3	<div class="card-icon"><i class="fas fa-users"></i></div>
4	<div class="card-title">Total User</div>
5	<div class="card-value">{{ total_user
6	}}</div>
7	</div>
8	<div class="card stat-card">
9	<div class="card-icon"><i class="fas fa-database"></i></div>
10	<div class="card-title">Dataset</div>
11	<div class="card-value">{{ total_dataset
12	}}</div>
13	</div>
14	<div class="card stat-card">
15	<div class="card-icon"><i class="fas fa-clipboard-check"></i></div>
16	<div class="card-title">Klasifikasi</div>
17	<div class="card-value">{{ total_klasifikasi
18	}}</div>
19	</div>
20	</div>
21	</div>

g. Segmen Kelola *User*

Segmen kelola *user* berisi *source code* yang menampilkan halaman dimana admin bisa melihat daftar *user*, menghapus, mengedit dan menambah daftar *user*.

Segmen Program 4.8 Kelola User

```

1  <table class="table table-hover align-middle">
2    <thead class="table-primary">
3      <tr>
4        <th>ID</th>
5        <th>Username</th>
6        <th>Status</th>
7        <th>Aksi</th>
8      </tr>
9    </thead>
10   <tbody>
11     {% for user in users %}
12       <tr>
13         <td>{{ user.id }}</td>
14         <td>{{ user.username }}</td>
15         <td><span class="badge bg-
16 success">Aktif</span></td>
17         <td class="d-flex gap-2">
18           <a href="{{ url_for('edit_user',
19 user_id=user.id) }}" class="btn btn-warning
20 btn-sm">Edit</a>
21           <form method="POST" action="{{
22 url_for('delete_user', user_id=user.id) }}"
23 onsubmit="return confirm('Hapus user ini?');">

```

Segmen Program 4.8 Lanjutan

22	<code><button type="submit" class="btn btn-danger btn-sm">Hapus</button></code>
23	<code></code>
24	<code></form></code>
25	<code></td></code>
26	<code></tr></code>
27	<code>{% endfor %}</code>
28	<code></tbody></code>
29	<code></table></code>
30	<code>Tambah User</code>

h. Segmen *Upload Dataset*

Segmen program ini berisi *source code* dimana admin dapat mengunggah file data CSV melalui form yang telah disediakan. Setelah file dipilih, form akan mengirim data ke *backend* untuk diproses lebih lanjut.

Segmen Program 4.9 *Upload Dataset*

1	<code><form method="POST" action="{{ url_for('upload') }}" enctype="multipart/form-data"></code>
2	<code></code>
3	<code><div class="mb-3"></code>
4	<code><input type="file" name="file" accept=".csv"</code>
5	<code>required class="form-control" /></code>
6	<code></div></code>
7	<code><button type="submit" class="btn btn-success">Upload</button></code>
8	<code></form></code>
9	<code></code>

i. Segmen Data Klasifikasi

Segmen ini berisi *source code* dimana admin bisa melihat seluruh hasil klasifikasi yang diinputkan oleh *user*, dan bisa mengedit, menambah ataupun menghapus data.

Segmen Program 4.10 Data Klasifikasi

```

1 <table class="table table-hover align-middle">
2   <thead class="table-primary">
3     <tr>
4       <th>ID</th>
5       <th>Username</th>
6       <th>Nama</th>
7       <th>Usia</th>
8       <th>Jenis Kelamin</th>
9       <th>Hasil</th>
10      <th>Aksi</th>
11    </tr>
12  </thead>
13  <tbody>
14    {% for item in data %}
15    <tr>
16      <td>{{ item.id }}</td>
17      <td>{{ item.username }}</td>
18      <td>{{ item.nama }}</td>
19      <td>{{ item.usia }}</td>
20      <td>{{ item.jenis_kelamin }}</td>
21      <td>

```

Segmen Program 4.10 Lanjutan

```

22         {% if item.hasil == 'Insomnia' %}
23         <span class="badge bg-danger">{{
24         item.hasil }}</span>
25         {% elif item.hasil == 'Sleep Apnea' %}
26         <span class="badge bg-warning text-
27         dark">{{ item.hasil }}</span>
28         {% else %}
29         <span class="badge bg-success">{{
30         item.hasil }}</span>
31         {% endif %}
32     </td>
33     <td class="d-flex gap-2">
34         <a href="#" class="btn btn-warning btn-
35         sm">Edit</a>
36         <form method="POST" action="#">
37             <button type="submit" class="btn btn-
38             danger btn-sm">Hapus</button>
39         </form>
40     </td>
41 </tr>
42     {% else %}
43     <td colspan="7" class="text-center">Tidak
44     ada data klasifikasi.</td>
45 </tr>
46     {% endfor %}
47 </tbody>
</table>

```

j. Segmen Grafik

Segmen program grafik yaitu *source code* untuk menampilkan hasil dari perbandingan metrik evaluasi, yaitu akurasi, presisi, recall dan f1-score.

Segmen Program 4.11 Grafik

```

1 <div class="container mt-5">
2   <div class="row justify-content-center">
3     <div class="col-lg-10">
4       <div class="card shadow border-0">
5         <div class="card-header bg-primary text-
6         white">
7           <h4 class="my-2">🇮🇩 Perbandingan Metrik
8           Algoritma</h4>
9         </div>
10        <div class="card-body text-center">
11          <div class="position-relative"
12          style="height: 60vh;">
13            <canvas id="metricsChart"></canvas>
14          </div>
15          <div class="mt-3 text-center">
16            <span class="badge bg-
17            primary">Akurasi</span>
18            <span class="badge bg-
19            success">Presisi</span>
20            <span class="badge bg-warning text-
21            dark">Recall</span>
22            <span class="badge bg-info">F1-
23            Score</span>
24          </div>
25        </div>
26      </div>
27    </div>
28  </div>
29 </div>

```

Segmen Program 4.11 Lanjutan

```

23 </div>
24     <p class="text-muted mt-3">
25         Grafik ini menampilkan perbandingan
26         akurasi, presisi, recall, dan F1-score tiga
27         algoritma klasifikasi.
28     </p>
29     <a href="{{
30         url_for('classification_data') }}" class="btn
31         btn-outline-primary mt-2">
32         Kembali ke Data Klasifikasi
33     </a>
34 </div>
35 </div>
36 </div>
37 <script
38 src="https://cdn.jsdelivr.net/npm/chart.js"></s
39 cript>
40 <script>
41     // Data dari Flask
42     const labels = JSON.parse('{{ labels | tojson
43 | safe }}');
44     const accuracies = JSON.parse('{{ accuracies
45 | tojson | safe }}');
46     const precisions = JSON.parse('{{ precisions
47 | tojson | safe }}');
48     const recalls = JSON.parse('{{ recalls |
49 tojson | safe }}');
50 
```

Segmen Program 4.11 Lanjutan

```

48     const fl_scores = JSON.parse('{{ fl_scores |
49     tojson | safe }}');
50
51     document.addEventListener('DOMContentLoaded',
52     function() {
53         const ctx =
54         document.getElementById('metricsChart').getCont
55         ext('2d');
56         new Chart(ctx, {
57             type: 'bar',
58             data: {
59                 labels: labels,
60                 datasets: [
61                     {
62                         label: 'Akurasi',
63                         data: accuracies,
64                         backgroundColor: 'rgba(54, 162, 235,
65                         0.8)',
66                         borderColor: 'rgba(54, 162, 235, 1)',
67                         borderWidth: 2,
68                         borderRadius: 6
69                     },
70                     {
71                         label: 'Presisi',
72                         data: precisions,
73                         backgroundColor: 'rgba(75, 192, 192,
74                         0.8)',

```

Segmen Program 4.11 Lanjutan

```

73         borderColor: 'rgba(75, 192, 192, 1)',
74         borderWidth: 2,
75         borderRadius: 6
76     },
77     {
78         label: 'Recall',
79         data: recalls,
80         backgroundColor: 'rgba(255, 206, 86,
81 0.8)',
82         borderColor: 'rgba(255, 206, 86, 1)',
83         borderWidth: 2,
84         borderRadius: 6
85     },
86     {
87         label: 'F1-Score',
88         data: f1_scores,
89         backgroundColor: 'rgba(153, 102, 255,
90 0.8)',
91         borderColor: 'rgba(153, 102, 255, 1)',
92         borderWidth: 2,
93         borderRadius: 6
94     }
95 ]
96 },
97 options: {
98     responsive: true,

```

Segmen Program 4.11 Lanjutan

```

99      maintainAspectRatio: false,
100     animation: {
101       duration: 1500,
102       easing: 'easeOutBounce',
103     },
104     scales: {
105       y: {
106         beginAtZero: true,
107         max: 1.05,
108         title: {
109           display: true,
110           text: 'Nilai',
111           font: { size: 16 }
112         },
113       },
114     },
115     plugins: {
116       legend: {
117         position: 'bottom',
118         labels: {
119           font: { size: 14 }
120         }
121       },
122       tooltip: {
123         backgroundColor: 'rgba(0,0,0,0.8)',

```

Segmen Program 4.11 Lanjutan

```

124         titleFont: { size: 16, weight: 'bold'
125     },
126         bodyFont: { size: 14 },
127         titleColor: '#fff',
128         bodyColor: '#fff'
129     }
130 }
131 }
132 });
133 });
134 </script>

```

4.2 Skenario Pengujian

Skenario pengujian merupakan tahapan yang dibuat untuk mengidentifikasi cara kerja sistem apakah sistem dapat berfungsi dengan baik atau terjadi kesalahan saat sistem dijalankan. Metode yang dipakai dalam skenario pengujian ini adalah metode *black box*, dimana pengujian dilakukan dengan fokus pada fungsi dan *output* sistem.

4.2.1 Pengujian Terhadap *Login*

Pengujian terhadap *login* diperlukan untuk memastikan bahwa *username* dan *password* dapat berfungsi sebagaimana mestinya.

a. Pengujian *Login User*

Tabel 4.1 Skenario *Login User*

NO	Skenario Pengujian	Text Case	Hasil yang
----	--------------------	-----------	------------

			Diharapkan
1	<i>Username</i> dan <i>Password</i> tidak diisi,	<i>Username</i> (kosong),	Sistem menampilkan pesan <i>error</i> : “ <i>isi bidang</i> ”

Tabel 4.1 Tabel Lanjutan

	lalu klik <i>Login</i>	<i>Password</i> (kosong)	<i>ini</i> ”
2	<i>Username</i> tidak sesuai, <i>Password</i> benar lalu klik <i>Login</i>	<i>Username</i> (salah), <i>Password</i> (benar)	Sistem menampilkan pesan <i>error</i> : “ <i>Username</i> atau <i>Password</i> salah”
3	<i>Username</i> benar, <i>Password</i> <i>tidak sesuai</i> lalu klik <i>Login</i>	<i>Username</i> (benar), <i>Password</i> (salah)	Sistem menampilkan pesan <i>error</i> : “ <i>Username</i> atau <i>Password</i> salah”
4	<i>Username</i> dan <i>Password</i> benar lalu klik <i>Login</i>	<i>Username</i> (benar), <i>Password</i> (benar)	Sistem mengizinkan akses dan <i>user</i> diarahkan ke halaman klasifikasi
5	Belum punya akun, klik link “Daftar di sini”	-	<i>User</i> diarahkan ke halaman Register

b. Pengujian *Login* Admin

Tabel 4.2 Skenario *Login* Admin

NO	Skenario Pengujian	Text Case	Hasil yang Diharapkan
1	<i>Username</i> dan <i>Password</i> tidak diisi, lalu klik <i>Login</i>	<i>Username</i> (kosong), <i>Password</i> (kosong)	Sistem menampilkan pesan <i>error</i> : “ <i>isi bidang ini</i> ”

2	<i>Username</i> tidak sesuai, <i>Password</i> benar lalu klik <i>Login</i>	<i>Username</i> (salah), <i>Password</i> (benar)	Sistem menampilkan pesan <i>error</i> : “ <i>Username</i> atau <i>Password</i> salah”
3	<i>Username</i> benar, <i>Password</i> tidak sesuai lalu klik <i>Login</i>	<i>Username</i> (benar), <i>Password</i> (salah)	Sistem menampilkan pesan <i>error</i> : “ <i>Username</i> atau <i>Password</i> salah”

Tabel 4.2 Tabel Lanjutan

4	<i>Username</i> dan <i>Password</i> benar lalu klik <i>Login</i>	<i>Username</i> (benar), <i>Password</i> (benar)	Sistem mengizinkan akses dan <i>user</i> diarahkan ke halaman <i>Dashboard Admin</i>
---	--	--	--

4.2.2 Pengujian Terhadap Transaksi

Pengujian terhadap transaksi yang ada pada sistem meliputi berbagai tindakap untuk memastikan sistem berfungsi dengan optimal.

a. Klasifikasi

Tabel 4.3 Skenario Klasifikasi

NO	Skenario Pengujian	Text Case	Hasil yang Diharapkan
1	Tidak mengisi semua kolom lalu klik Kirim & Klasifikasi	Nama, usia dan lainnya kosong	Sistem meminta pengguna mengisi data yang diperlukan

2	Kolom opsional tidak diisi lalu klik Kirim & Klasifikasi	“Tekanan Darah” dan “Detak Jantung” kosong	Sistem tetap menjalankan proses klasifikasi sesuai data yang diisi
3	Usia diisi diluar rentang 4-120 lalu klik Kirim & Klasifikasi	Usia : 3 atau 121	Sistem menolak dan form tidak tersubmit
4	Detak jantung diisi diluar rentang 40-200 lalu klik Kirim & Klasifikasi	Detak jantung : 39 atau 201	Sistem menolak, namun kolom opsional boleh kosong, jika diisi harus sesuai ketentuan
5	Semua kolom diisi sesuai ketentuan lalu klik Kirim & Klasifikasi	Semua data diisi sesuai petunjuk	Sistem menerima input, proses klasifikasi dijalankan dan <i>user</i> diarahkan ke halaman hasil klasifikasi

b. Hasil Klasifikasi

Tabel 4.4 Skenario Hasil Klasifikasi

NO	Skenario Pengujian	Hasil yang Diharapkan
1	Semua data yang diinput lengkap dan valid	Halaman hasil menampilkan seluruh data pengguna dan hasil klasifikasi
2	Tekan <i>Download</i> PDF	Hasil klasifikasi dan data pada halaman hasil klasifikasi berhasil diunduh dalam bentuk format PDF
3	Tekan Cetak	Printer dialog tampil dan hasil klasifikasi siap dicetak
4	Tekan Kembali	<i>User</i> diarahkan ke halaman klasifikasi dan siap mengisi data baru

c. Kelola *User*

Tabel 4.5 Skenario Kelola *User*

NO	Skenario Pengujian	Hasil yang Diharapkan
1	Halaman diakses tanpa data <i>user</i>	Tabel kosong karena tidak adanya data <i>user</i> yang terdaftar
2	Ada data <i>user</i>	Semua <i>user</i> ditampilkan dalam tabel
3	Klik tombol tambah <i>user</i>	Akan diarahkan ke halaman untuk menambahkan <i>user</i> baru
4	Klik tombol edit pada <i>user</i> tertentu	Akan diarahkan ke halaman edit data <i>user</i> yang dipilih
5	Klik tombol hapus pada <i>user</i> tertentu	Muncul <i>popup</i> konfirmasi untuk menghapus <i>user</i> yang dipilih

d. Upload Dataset

Tabel 4.6 Skenario Upload Dataset

NO	Skenario Pengujian	Hasil yang Diharapkan
1	Tidak memilih file lalu klik Upload	Sistem menampilkan pesan “pilih file”
2	Memilih file bukan CSV	Sistem akan meminta untuk memilih file CSV
3	Memilih file CSV	File diterima server dan berhasil di <i>upload</i>

e. Data Klasifikasi

Tabel 4.7 Skenario Data Klasifikasi

NO	Skenario Pengujian	Hasil yang Diharapkan
1	Tidak ada data klasifikasi	Menampilkan kolom dengan pesan “Tidak ada data

		klasifikasi”
2	Ada data klasifikasi	Semua data akan ditampilkan
3	Klik tombol edit pada suatu data	Mengarah ke halaman edit data klasifikasi
4	Klik tombol hapus pada suatu data	Menampilkan konfirmasi hapus data

f. Grafik

Tabel 4.8 Skenario Grafik

NO	Skenario Pengujian	Hasil yang Diharapkan
1	Tidak ada data grafik	Halaman tidak menampilkan grafik
2	Ada data algoritma	Diagram batang menunjukkan perbandingan algoritma berupa akurasi, presisi, <i>recall</i> dan <i>f1-score</i>
3	Klik tombol kembali	Kembali ke halaman klasifikasi
4	Data berubah di <i>backend</i>	Grafik otomatis berubah sesuai dengan data terbaru

4.3 Pengujian

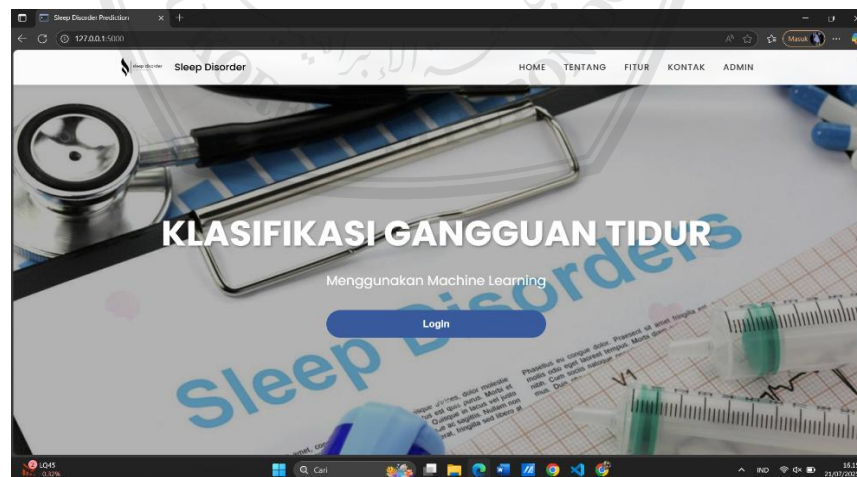
Pada tahap pengujian program akan diuji secara menyeluruh untuk memastikan bahwa sistem berfungsi sesuai dengan tujuan dan kriteria yang telah ditentukan, dan akan dilakukan pengecekan terhadap kemungkinan kesalahan atau error pada sistem sebelum digunakan lebih lanjut. Langkah ini merupakan tahapan terakhir yang harus dilakukan setelah pembuatan sistem selesai dengan tujuan memastikan kualitas dan keandalan sistem sebelum dioperasikan oleh pengguna.

4.3.1 Cara Kerja Sistem

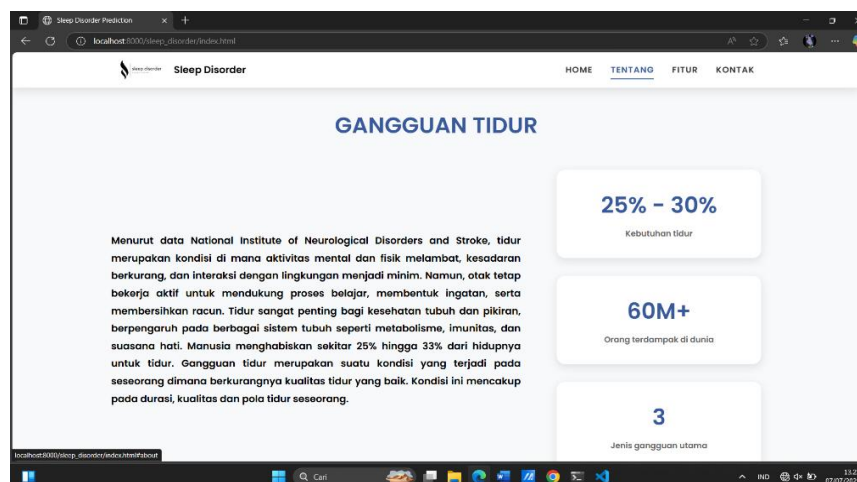
Cara kerja sistem menjelaskan tentang bagaimana sistem berjalan untuk mencapai tujuan yang diharapkan. Berikut ini cara kerja sistem Klasifikasi Gangguan Tidur Menggunakan *Machine Learning* :

a. *Home*

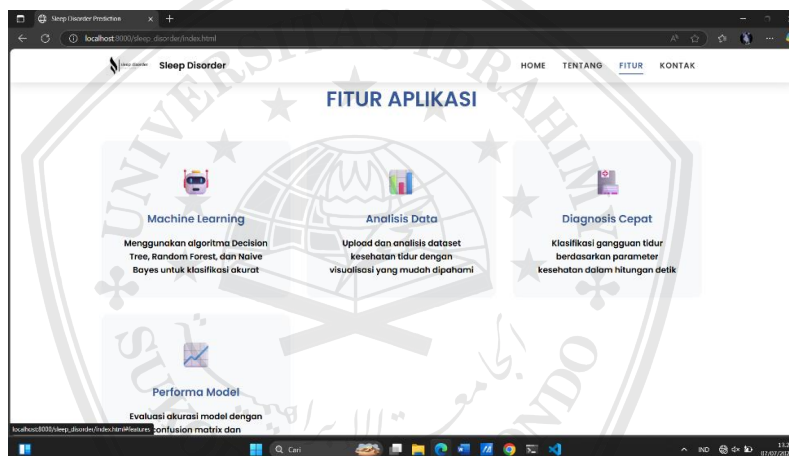
Sebelum melakukan *login* pengguna akan diarahkan terlebih dahulu pada halaman *home*. Dimana pada halaman ini ada empat menu yang bisa diakses oleh pengguna secara umum, dan satu menu untuk diakses oleh admin. Menu *home* menyediakan tombol *login* yang akan diarahkan ke halaman *login*. Menu Tentang yang berisi mengenai informasi mengenai gangguan tidur. Menu Fitur akan menampilkan fitur apa saja yang dimiliki oleh sistem. Dan menu Kontak, pengguna dapat mengirim pesan untuk mengetahui informasi lebih lanjut mengenai sistem. Berikut ini tampilan halaman *home* :



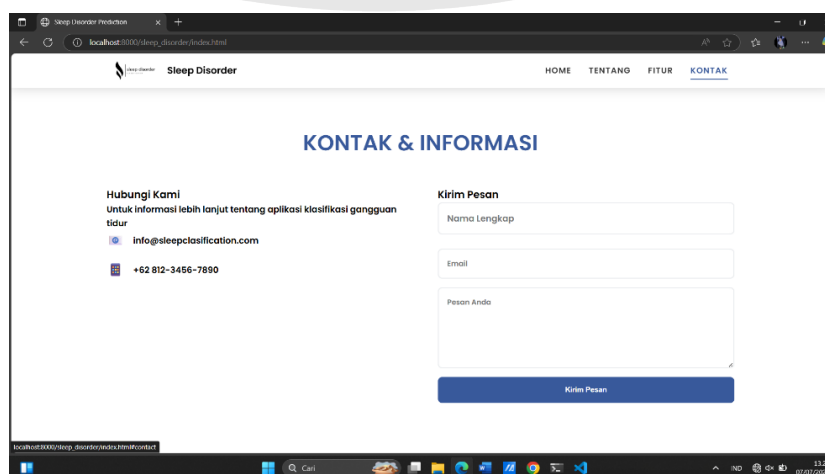
Gambar 4.9 *Home*



Gambar 4.10 *Home Tentang*



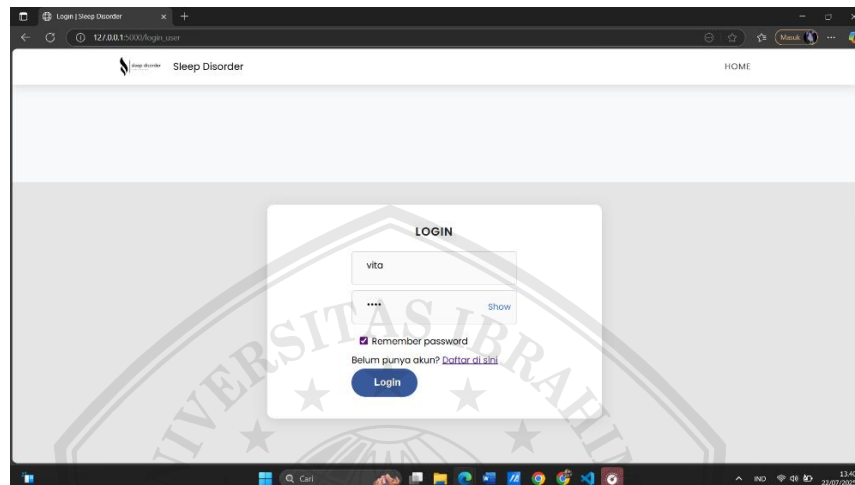
Gambar 4.11 *Home Fitur Aplikasi*



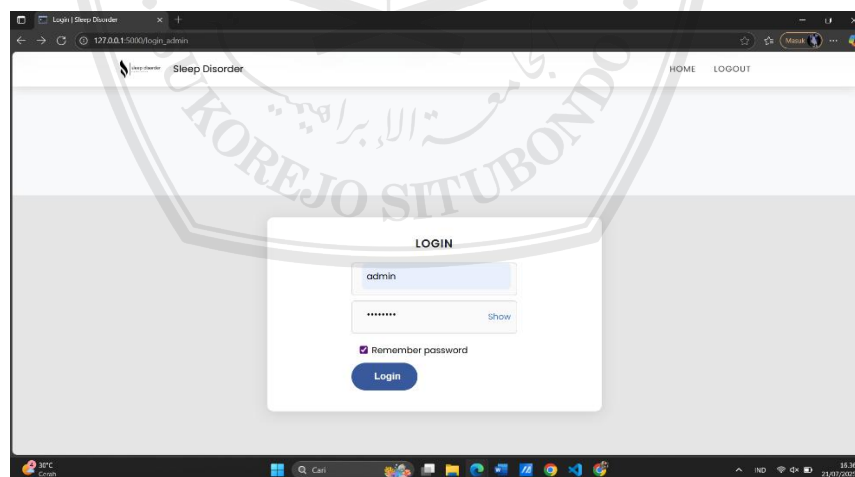
Gambar 4.12 *Home Kontak & Informasi*

b. *Login*

Agar dapat masuk kedalam sistem, baik pengguna maupun admin harus melakukan proses *login* terlebih dahulu dengan memasukkan *username* dan *password*. Halaman *login* ditunjukkan pada gambar berikut :



Gambar 4.13 Login User

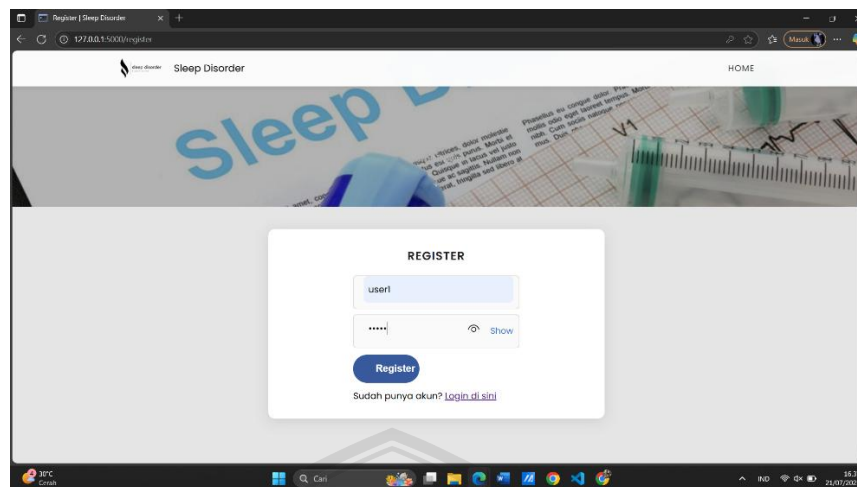


Gambar 4.14 Login Admin

c. *Register*

Bagi pengguna yang masih belum mempunyai akun, harus melakukan proses register agar bisa login ke dalam sistem. Halaman *register* sebagai

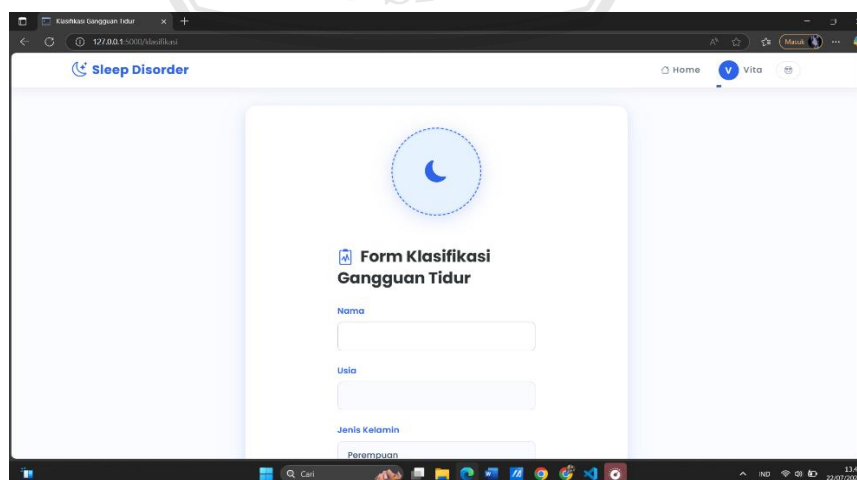
berikut :



Gambar 4.15 Register

d. Klasifikasi

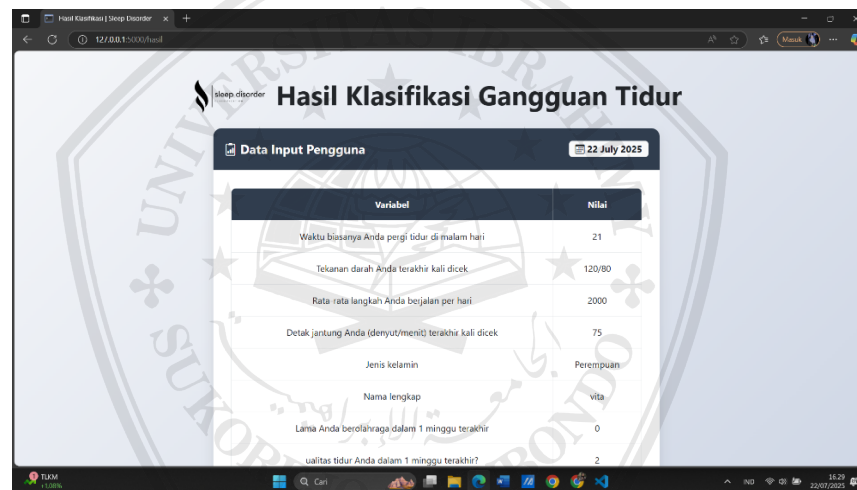
Halaman klasifikasi ini dibuat untuk memudahkan pengguna dalam melakukan klasifikasi gangguan tidur dengan menginputkan informasi sesuai dengan kolom input yang tersedia. Setelah semua kolom terisi, pengguna dapat menekan tombol klasifikasikan untuk memulai proses klasifikasi. Berikut tampilan halaman klasifikasi :



Gambar 4.16 Klasifikasi

e. Hasil Klasifikasi

Pada halaman ini akan menampilkan hasil dari klasifikasi gangguan tidur yang dialami oleh *user* berdasarkan data yang diinputkan. Hasil klasifikasi akan menunjukkan *healty* atau tanpa gangguan tidur, *insomnia*, atau *sleep apnea*. Pada halaman ini *user* bisa mengklik tombol cetak hasil untuk mencetak hasil klasifikasi maupun mengunduhnya dalam bentuk pdf. Berikut halaman hasil klasifikasinya :

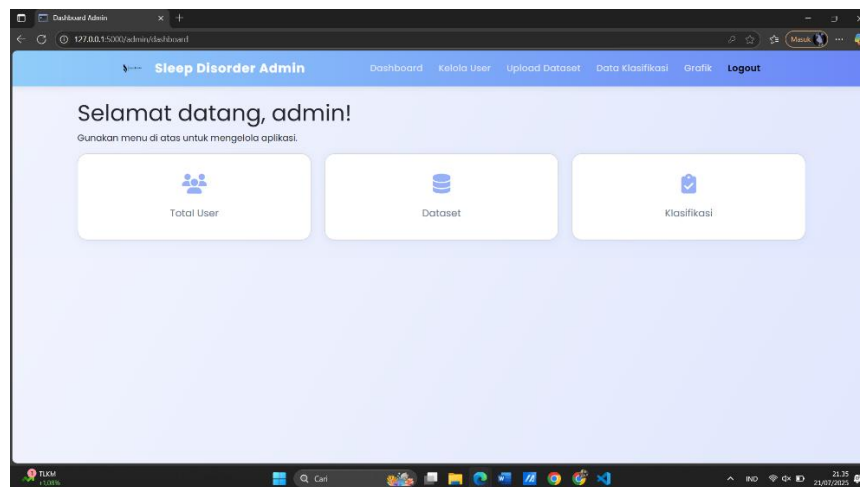


Variabel	Nilai
Waktu biasanya Anda pergi tidur di malam hari	21
Tekanan darah Anda terakhir kali dicek	120/80
Rata-rata langkah Anda berjalan per hari	2000
Detak jantung Anda (denyut/menit) terakhir kali dicek	75
Jenis kelamin	Perempuan
Nama lengkap	vita
Lama Anda berolahraga dalam 1 minggu terakhir	0
Waktu tidur Anda dalam 1 minggu terakhir?	2

Gambar 4.17 Hasil Klasifikasi

f. Dashboard Admin

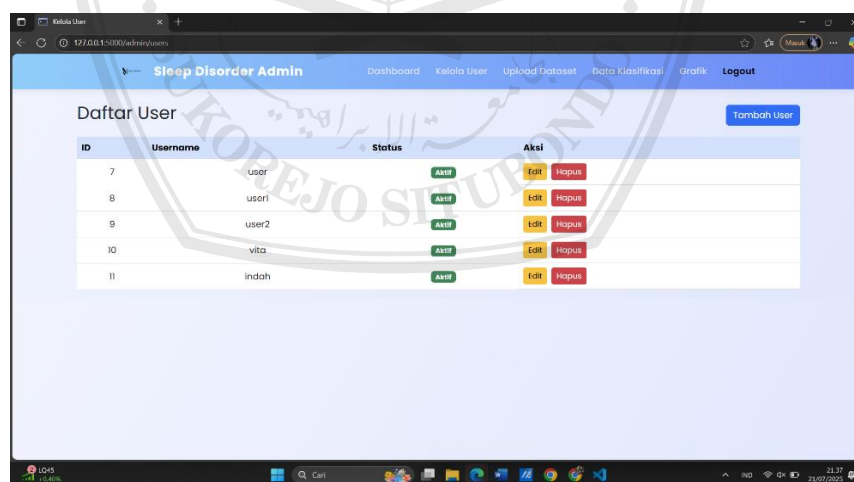
Halaman ini menampilkan tampilan awal ketika admin berhasil login ke sistem. Halaman *dashboard* admin sebagaimana gambar dibawah ini :



Gambar 4.18 Dashboard Admin

g. Kelola User

Halaman ini memungkinkan admin untuk mengelola data *user* yang menggunakan sistem. Dimana admin bisa melakukan menambah, mengedit, dan menghapus data *user*. Berikut halaman kelola *user* :

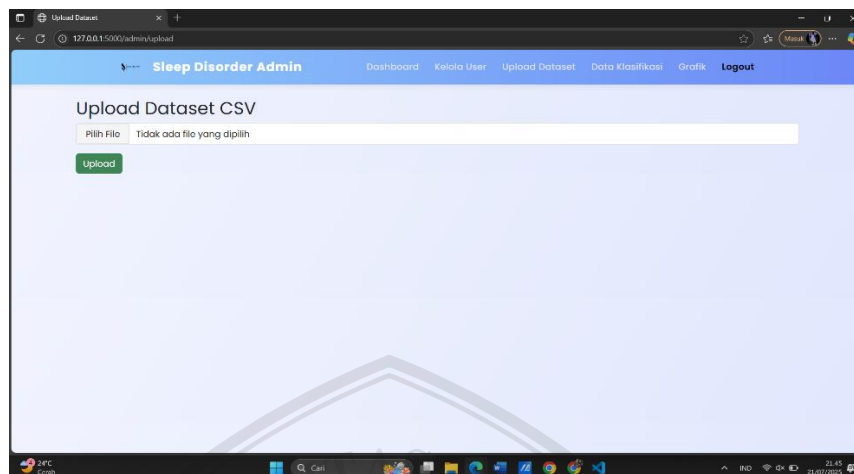


Gambar 4.19 Kelola User

h. Upload Dataset

Pada halaman ini, admin menginputkan *dataset* yang akan digunakan dalam proses klasifikasi. Setelah proses *upload* selesai, data akan

ditampilkan, dan admin dapat melakukan proses *training* dan *testing* pada data . Halaman ini ditampilkan sebagai berikut :



Gambar 4.20 Upload Dataset

ID	Jenis Kelamin	Usia	Pekerjaan	Durasi Tidur	Kualitas Tidur	Aktivitas Fisik	Level Stres	Kategori BMI	Tekanan Darah	Detak Jantung	Langkah Harian	Gangguan Tidur
1	Male	27	Software Engineer	6.01	6	42	6	Overweight	126/83	77	4200	NaN
2	Male	28	Doctor	6.02	6	60	8	Normal	125/80	75	10000	NaN
3	Male	28	Doctor	6.02	6	60	8	Normal	125/80	75	10000	NaN
4	Male	28	Sales Representative	5.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
5	Male	28	Sales Representative	5.09	4	30	8	Obese	140/90	85	3000	Sleep Apnea
6	Male	28	Software Engineer	5.09	4	30	8	Obese	140/90	85	3000	Insomnia
7	Male	29	Teacher	6.03	6	40	7	Obese	140/90	82	3500	Insomnia
8	Male	29	Doctor	7.08	7	75	6	Normal	120/80	70	8000	NaN
9	Male	29	Doctor	7.08	7	75	6	Normal	120/80	70	8000	NaN
10	Male	29	Doctor	7.08	7	75	6	Normal	120/80	70	8000	NaN

Gambar 4.21 Preview Data

Preview Data CSV | Sleep Disorde... | X

127.0.0.1:5000/admin/preview

Mask

6	18	Male	29	Doctor	6.00	6	30	8	Normal	120/80	70	8000	Sleep Apnea
7	19	Female	29	Nurse	6.05	5	40	7	Normal Weight	132/87	80	4000	Insomnia
8	31	Female	30	Nurse	6.04	5	35	7	Normal Weight	130/86	78	4100	Sleep Apnea
9	32	Female	30	Nurse	6.04	5	35	7	Normal Weight	130/86	78	4100	Insomnia
10	50	Male	31	Doctor	7.07	7	75	6	Normal	120/80	70	8000	Sleep Apnea

Training & Testing Data

Hasil Training & Testing:

Model	Akurasi	Presisi	Recall	F1 Score
Decision Tree	0.9149	0.9266	0.9149	0.9138
Random Forest	0.8723	0.8743	0.8723	0.8717
Naive Bayes	0.7659	0.7658	0.7659	0.7655

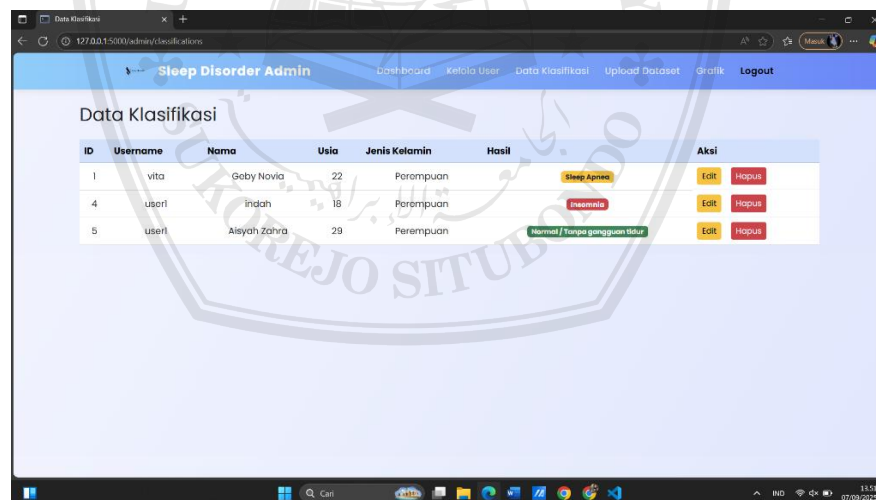
Download Report

Gambar 4.22 Hasil *Training* dan *Testing*

i. Data Klasifikasi

Pada halaman ini, admin dapat melihat data klasifikasi dari setiap *user*.

Berikut tampilan halaman data klasifikasi :



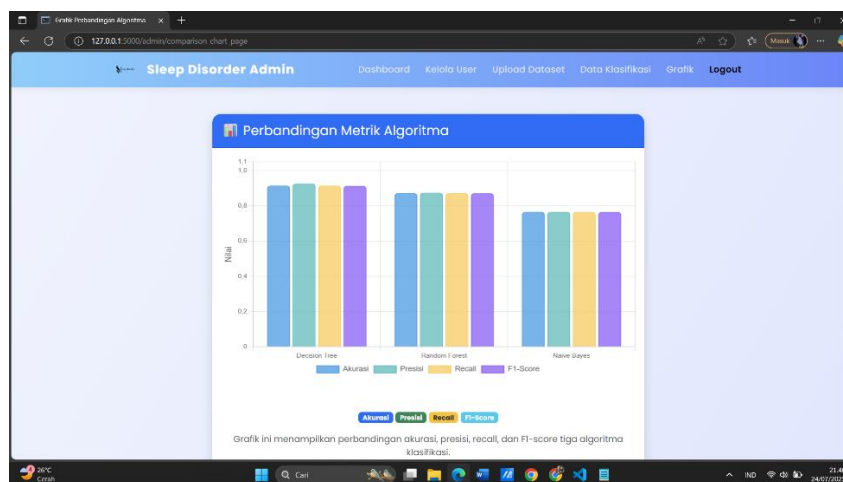
ID	Username	Nama	Usia	Jenis Kelamin	Hasil	Aksi
1	vita	Gaby Nova	22	Perempuan	Sleep Apnea	Edit Hapus
4	user1	Indah	18	Perempuan	Insomnia	Edit Hapus
5	user1	Aisyah Zahra	29	Perempuan	Normal / Tanpa gangguan tidur	Edit Hapus

Gambar 4.23 Data Klasifikasi

j. Grafik Perbandingan Model

Pada halaman ini, akan ditampilkan grafik hasil perbandingan dari model

klasifikasi yang digunakan. Berikut tampilan grafik perbandingan mode :



Gambar 4.24 Grafik Perbandingan Model

4.3.2 Hasil Pengujian

Hasil pengujian sistem didapatkan setelah melakukan pengujian terhadap sistem untuk melihat cara kerja sistem. Hasil pengujian sistem sebagaimana tabel dibawah ini :

Tabel 4.9 Hasil Pengujian

No	Nama Form	Aktivitas Pengujian	Hasil Pengujian	Keterangan
1	Home	Sebelum melakukan login, pengguna akan masuk ke halaman home.	Berhasil	-
2	Login	Untuk melakukan tahap klasifikasi, pengguna dan admin wajib melakukan login terlebih dahulu dengan menginputkan	Berhasil	-

		<i>username</i> dan <i>password</i> .		
--	--	---------------------------------------	--	--

Tabel 4.9 Tabel Lanjutan

3	Klasifikasi	Isi kolom berdasarkan pertanyaan yang sudah ditentukan. Lalu klik tombol klasifikasikan untuk memulai klasifikasi.	Berhasil	-
4	Hasil Klasifikasi	Hasil akan ditampilkan. Klik tombol cetak halaman untuk mencetak hasil klasifikasi. Klik tombol <i>download</i> PDF untuk <i>mendownload</i> hasil klasifikasi dalam bentuk PDF.	Berhasil	-

5	Kelola <i>User</i>	Melihat daftar <i>user</i> , mengedit, menghapus dan menambah <i>user</i>	Berhasil	-
6	<i>Upload</i> <i>Dataset</i>	Klik tombol pilih file maka akan masuk ke <i>file explorer</i> untuk memilih dataset yang akan digunakan, lalu klik tombol <i>upload</i> untuk mengupload file.	Berhasil	-
7	Data <i>Preview</i>	Data yang di <i>upload</i> akan ditampilkan, dan dibagian paling bawah ada tombol <i>Train Test</i> . Untuk menyiapkan data agar siap digunakan.	Berhasil	-
8	Data Klasifikasi	Melihat data klasifikasi yang diinput <i>user</i> , mengedit dan menghapus data klasifikasi	Berhasil	-

Tabel 4.9 Tabel Lanjutan

7	Grafik Perbandingan Model	Klik sub menu grafik maka akan tampil grafik perbandingan model.	Berhasil	-
---	---------------------------	--	----------	---

Skenario pengujian ini dilakukan dengan menjalankan sendiri sistem yang sudah dibuat. Berdasarkan hasil pengujian, dapat dilihat bahwa sistem sudah dapat berjalan lancar dan dapat digunakan.

4.4 *Maintenance*

Maintenance (pemeliharaan) merupakan serangkaian kegiatan untuk memastikan sistem tetap berfungsi dengan baik, melakukan perbaikan jika dibutuhkan, serta mengembangkan sistem agar selalu efektif sesuai kebutuhan. Kegiatan ini meliputi pembaruan dataset secara rutin guna memperoleh data baru yang lebih akurat, evaluasi performa algoritma agar hasil klasifikasi lebih optimal, serta melakukan pencadangan data sebagai langkah antisipasi terhadap kemungkinan kehilangan data dikemudian hari. Dengan pemeliharaan yang teratur, sistem dapat terus berjalan optimal dan efektif dalam jangka yang panjang.

BAB V

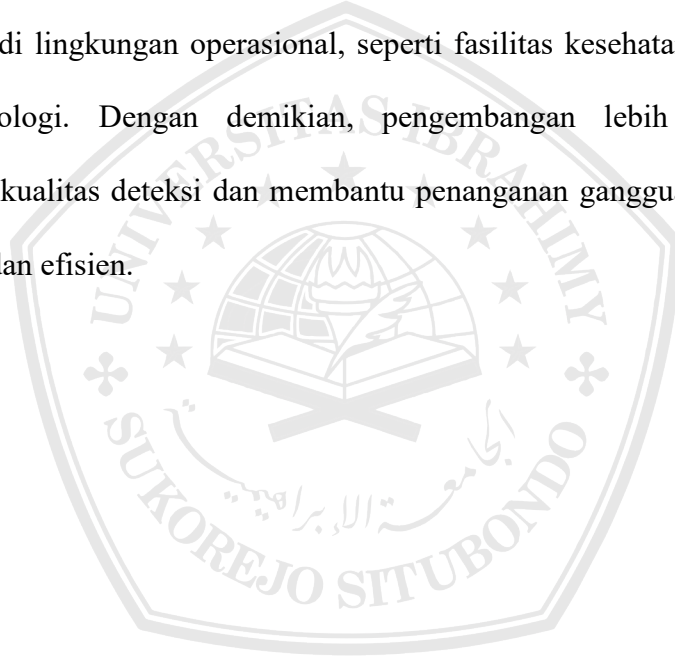
PENUTUP

5.1 Kesimpulan

Mengacu pada hasil klasifikasi yang telah dilakukan terhadap algoritma Decision Tree, Random Forest, dan Naïve Bayes dalam mengklasifikasikan gangguan tidur, dapat disimpulkan bahwa algoritma Decision Tree memiliki kinerja yang optimal dibandingkan algoritma Random Forest dan Naïve Bayes. Hal ini ditunjukkan oleh hasil accuracy tertinggi yang diperoleh pada algoritma Decision Tree yaitu sebesar 91.49%, serta kemampuan precision dan recall yang baik dalam membedakan penderita gangguan tidur dan tanpa gangguan tidur. Sedangkan algoritma Random Forest menghasilkan akurasi sebesar 87.23% dan algoritma Naïve Bayes dengan akurasi sebesar 76.59%. Penilaian kinerja masing – masing model dilakukan dengan memanfaatkan beberapa metrik evaluasi yakni accuracy, precision, recall, dan f1-score baik secara otomatis melalui Google Colab maupun secara manual menggunakan rumus statistik, dan hasilnya menunjukkan konsistensi. Secara keseluruhan, algoritma Decision Tree direkomendasikan sebagai algoritma yang efektif dan optimal dalam klasifikasi gangguan tidur berdasarkan data yang digunakan. Sehingga pembuatan sistem klasifikasi gangguan tidur menggunakan *machine learning* dapat dilakukan mulai dari perancangan, penerapan model, pengujian hingga evaluasi akhir. Dengan demikian sistem yang dibuat dapat berfungsi sebagai pendukung atau alat bantu dalam proses identifikasi gangguan tidur.

5.2 Saran

Penelitian ini belum sepenuhnya mencapai kata sempurna dan memiliki berbagai kekurangan. Diharapkan untuk penelitian – penelitian selanjutnya dapat membandingkan algoritma lainnya untuk mendapatkan hasil analisis yang lebih optimal dan akurat. Selain itu, hasil dari penelitian ini diharapkan dapat digunakan sebagai referensi atau acuan dalam proses pengembangan sistem deteksi gangguan tidur yang tidak hanya efektif secara teoritis, tetapi juga dapat di implementasikan secara praktis di lingkungan operasional, seperti fasilitas kesehatan atau aplikasi berbasis teknologi. Dengan demikian, pengembangan lebih lanjut dapat meningkatkan kualitas deteksi dan membantu penanganan gangguan tidur secara lebih optimal dan efisien.



DAFTAR PUSTAKA

- [1] A. Yogiarto, A. Homaidi, and Z. Fatah, "Implementasi Metode K-Nearest Neighbors (KNN) untuk Klasifikasi Penyakit Jantung," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1720–1728, 2024, doi: 10.33379/gtech.v8i3.4495.
- [2] V. Renganathan, *Machine Learning Algorithms for Data Scientists: An Overview*. Vinaitheerthan Renganathan, 2021.
- [3] A. Prabowo, S. Wardani, R. Wijaya Dewantoro, and W. Wesly, "Komparasi Tingkat Akurasi Random Forest dan Decision Tree C4.5 Pada Klasifikasi Data Penyakit Infertilitas," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 1, pp. 218–224, 2023, doi: 10.30865/klik.v4i1.1115.
- [4] D. Fuji Astari, Y. Herry Chrisnanto, and M. Melina, "Klasifikasi Tingkat Stres Saat Tidur Menggunakan Algoritma Random Forest," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 5, pp. 3676–3684, 2024, doi: 10.36040/jati.v7i5.7750.
- [5] Y. Brianorman and S. Sucipto, "Prediction of Prospective New Students Using Decision Tree, Random Forest, and Naive Bayes," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 18, no. 3, pp. 1433–1446, 2024, doi: 10.30598/barekengvol18iss3pp1433-1446.
- [6] M. B. Arya Darmawan, F. Dewanta, and S. Astuti, "Analisis Perbandingan Algoritma Decision Tree, Random Forest, dan Naïve Bayes untuk Prediksi Banjir di Desa Dayeuhkolot," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 9, no. 1, pp. 52–61, 2023, doi: 10.15575/telka.v9n1.52-61.
- [7] M. Yasir and R. Suraji, "Perbandingan Metode Klasifikasi Naive Bayes, Decision, Tree, Random Forest Terhadap Analisis Sentimen Kenaikan Biaya Haji 2023 pada Media Sosial Youtube," *J. Cahaya Mandalika*, vol. 3, no. 2, pp. 180–192, 2023.
- [8] N. I. of N. D. and Stroke, "Brain Basics: Understanding Sleep." Accessed: Feb. 15, 2023. [Online]. Available: <https://www.ninds.nih.gov/>
- [9] P. Dr. H. Fuad Nashori, S.Psi., M.Si., M.Ag. and P. Etik Dwi Wulandari, S.Psi., M.Psi., "Psikologi Tidur: Dari kualitas tidur hingga insomnia," vol. 11, no. 1, pp. 1–14, 2017.
- [10] D. J. P. Kesehatan, "Mengenal Gangguan Tidur." [Online]. Available: https://yankes.kemkes.go.id/view_artikel/502/yuks-mengenal-gangguan-tidur
- [11] A. Gofir, "Buku Pintar Kasus Neurologi," Gadjah Mada University Press, 2024.
- [12] M. Zed, *Metode Penelitian Kepustakaan (Library Research)*. Yayasan Pustaka Obor Indonesia, 2008, 2008.

- [13] MUO, "A Beginner's Guide to Kaggle for Data Science." [Online]. Available: <https://www.makeuseof.com/beginners-guide-to-kaggle/>
- [14] A. Goldbloom, "Kaggle." [Online]. Available: <https://g.co/kgs/aYoZCrk>
- [15] A. . Rahmat, M. . Ladjamuddin, and T. . Awaludin, "Perbandingan Algoritma Decision Tree, Random Forest Dan Naive Bayes Pada Prediksi Penilaian Kepuasan Penumpang Maskapai Pesawat Menggunakan Dataset Kaggle," *J. Rekayasa Inf.*, vol. 12, no. 2, pp. 150–159, 2023, [Online]. Available: www.kaggle.com,
- [16] M. K. Amril Mutoi Siregar, S.Kom., M.Kom. DAN Adam Puspabhuana, S.Kom., *DATA MINING Pengolahan Data Menjadi Informasi dengan RapidMiner*. CV Kekata Group.
- [17] A. V. M. Taufiq, *Analisis Sistem Informasi (Konsep, Metodologi, dan Pendekatan)*. CV Ananta Vidya, 2023.
- [18] J. P. Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques*. Elsevier Science.
- [19] oko A. Nursiyono, "MACHINE LEARNING dengan R Teori & Praktikum," Media Nusa Creative (MNC Publishing), 2023.
- [20] N. Salmi and Z. Rustam, "Naïve Bayes Classifier Models for Predicting the Colon Cancer," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 546, no. 5, 2019, doi: 10.1088/1757-899X/546/5/052068.
- [21] I. A. Prabowo, D. Remawati, and A. P. W. Wardana, "Klasifikasi Tingkat Gangguan Tidur Menggunakan Algoritma Naïve Bayes," *J. Teknol. Inf. dan Komun.*, vol. 8, no. 2, 2020, doi: 10.30646/tikomsin.v8i2.519.
- [22] Yuhefizar, *CMM Website Interaktif MCMS Joomla(CMS)*. Elex Media Komputindo.
- [23] D. A. F. Wahyudi Agustiono, Yudha Dwi Putr N, *Analisa dan Desain Sistem Informasi*. Media Nusa Creative (MNC Publishing).
- [24] E. Bisong, "Google Colaboratory BT - Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners," Berkeley, CA: Apress, Berkeley, CA, 2019, pp. 59–64.
- [25] S. Bin Uzayr, *Mastering Visual Studio Code: A Beginner's Guide*. Taylor & Francis Group, 2022.
- [26] K. Niglas, "Media Review: Microsoft Office Excel Spreadsheet Software," *J. Mix. Methods Res.*, vol. 1, no. 3, pp. 297–299, [Online]. Available: <https://journals.sagepub.com/doi/abs/10.1177/1558689807301250>
- [27] B. U. Manalu, "Modul : Pengantar Python dengan Google Colab," pp. 1–2, 2020, [Online]. Available: https://figshare.com/articles/book/Modul_Pengantar_Python_dengan_Goog

le_Colab/12683627?file=24018779

- [28] A. Agung, A. Daniswara, I. Kadek, and D. Nuryana, "Data Preprocessing Pola Pada Penilaian Mahasiswa Program Profesi Guru," *J. Informatics Comput. Sci.*, vol. 05, pp. 97–100, 2023.
- [29] L. Setiyani and E. Tjandra, "Analisis Kebutuhan Fungsional Aplikasi Penanganan Keluhan Mahasiswa Studi Kasus: Stmik Rosma Karawang," *J. Inov. Pendidik. dan Teknol. Inf.*, vol. 2, no. 1, pp. 8–17, 2021, doi: 10.52060/pti.v2i01.465.
- [30] R. Utami *et al.*, *Buku Ajar Analisis Perancangan Sistem*. PT. Sonpedia Publishing Indonesia, 2024.
- [31] F. Soufitri, S.Kom., MMSI, *Konsep Sistem Informasi*. PT Inovasi Pratama Internasional.
- [32] H. Sihotang, *Metode Penelitian Kuantitatif*. 2023. [Online]. Available: <http://www.nber.org/papers/w16019>
- [33] N. R. Wiwesa, "User Interface Dan User Experience Untuk Mengelola Kepuasan Pelanggan," *J. Sos. Hum. Terap.*, vol. 3, no. 2, pp. 17–31, 2021, [Online]. Available: <https://scholarhub.ui.ac.id/jsht/vol3/iss2/2>
- [34] I. K. D. G. Supartha *et al.*, *BUKU AJAR ANALISA PERANCANGAN SISTEM*. PT. Sonpedia Publishing Indonesia, 2023.
- [35] M. Hikmal Maulana, "Python Bahasa Pemrograman Yang Ramah Bagi Pemula," *JISCO (Journal Inf. Syst. Comput.*, vol. 2, no. 2, pp. 73–78, 2024, [Online]. Available: <https://jurnal.fst.uinjambi.ac.id/index.php/jisco/index>
- [36] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. "O'Reilly Media, Inc.," 2014.

1. Surat Keterangan Hasil Cek Plagiasi



PONDOK PESANTREN SALAFIYAH SYAFI'YAH SUKOREJO
UNIVERSITAS IBRAHIMY
PERPUSTAKAAN IBRAHIMY
 N P P . 3 5 1 2 1 4 2 F 2 0 0 6 5 6 7
 Jl. KHR. Syamsul Arifin No. 1-2 PO. Box. 2 Kode Pos. 68374 Phone (0338) 452666 Fax. (0338) 453068
 SUMBEREJO BANYUPUTIH SITUBONDO JAWA TIMUR



**SURAT KETERANGAN
HASIL PEMERIKSAAN PLAGIASI**

Yang bertanda tangan di bawah ini

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

Menyatakan dengan sebenarnya bahwa:

NIM : 2021502067
 Nama : INDAH NOVITA SARI
 Fakultas : Sains dan Teknologi
 Prodi : Sistem Informasi
 Kecamatan : BONDOWOSO
 Kabupaten : BONDOWOSO
 Provinsi :
 Judul Skripsi : Analisis Perbandingan Algoritma Decision Tree, Random Forest, Dan Naive Bayes Dalam Klasifikasi Gangguan Tidur

Dengan dosen Pembimbing :

1. Lukman Fakhri Lidimilah, M.Kom
2. Ahmad Lutfi, M.Kom

Telah dilakukan cek plagiasi di Perpustakaan Universitas Ibrahimiy dengan persentase plagiasi terakhir sebesar **25%**.

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Sukorejo, 3 Agustus 2025
 Kepala Perpustakaan,

 Muhammad Ali Ridla, M.Kom.


 Dokumen ini telah ditandatangani secara elektronik.



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

www.lib.ibrahimiy.ac.id
library@ibrahimiy.ac.id
[Perpustakaan Ibrahimiy](#)
[@ibrahimiy_lib](#)

2. Surat Keterangan Kesiapan Publikasi

LEMBAR PERNYATAAN KESEDIAAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama : **INDAH NOVITA SARI**
NIM/NPM : 2021502067
Program Studi : Sistem Informasi
Fakultas : Sains dan Teknologi
Jenis Karya Ilmiah : Skripsi

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

"ANALISIS PERBANDINGAN ALGORITMA *DECISION TREE*, *RANDOM FOREST*, DAN *NAÏVE BAYES* DALAM KLASIFIKASI GANGGUAN *TIDUR*"

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

Demikian pernyataan ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Situbondo, 08 Juni 2026
Yang Menyatakan


INDAH NOVITA SARI