# BAB II
# LANDASAN TEORI

## 2.1 Tinjauan Pustaka

Identifikasi dini risiko diabetes merupakan tantangan penting dalam kesehatan global. Banyak penelitian telah menerapkan pendekatan *machine learning* untuk klasifikasi diabetes (Tigga dan Garg, 2020). Salah satu dataset yang sering digunakan dalam penelitian diabetes adalah PIMA Indian, yang menjadi standar benchmark dalam klasifikasi diabetes (Reddy dkk., 2024).

Penelitian yang dilakukan oleh Kumari dkk (2021) mengusulkan pendekatan *ensemble learning* dalam klasifikasi dan prediksi diabetes dengan memanfaatkan kombinasi algoritma seperti *decision tree, logistic regression,* dan *random forest.* Dengan menggunakan dataset PIMA Indian, pendekatan *ensemble* ini mampu meningkatkan akurasi klasifikasi hingga mencapai 79.08%, menunjukkan keunggulan model gabungan dalam mengatasi kompleksitas data medis dibandingkan dengan model tunggal. Sementara itu, penelitian oleh Verma dan Khatoon (2024) membahas penerapan dan evaluasi berbagai algoritma klasifikasi dalam diagnosis diabetes pada dataset yang sama. Empat algoritma *machine learning* yaitu, *logistic regression, support vector machine, k-nearest neighbor*, dan *random forest*. Hasil pengujian menunjukkan bahwa *random forest* merupakan algoritma dengan performa terbaik, menghasilkan akurasi sebesar 80,08%. Penelitian ini juga menekankan pentingnya tahapan pra-pemrosesan data dan pengujian berbagai algoritma untuk mengidentifikasi metode paling efektif. Hasil temuan ini memperkuat posisi *random forest* sebagai algoritma yang andal dan stabil dalam menangani data medis kompleks.

Penelitian lebih lanjut oleh Farsana dan Poulose (2024) mengusulkan pendekatan *hybrid* berbasis CNN yang dikombinasikan dengan algoritma seperti *random forest* dan *support vector machine.* Penelitian ini menekankan pentingnya pra-pemrosesan data, seperti imputasi nilai hilang dan transformasi *outlier.* Penelitian oleh Sivaranjani dkk. (2021) mengusulkan pendekatan prediksi diabetes dengan memanfaatkan algoritma *random forest* dan *support vector machine* dengan

menggunakan dataset PIMA Indian. Hasilnya menunjukkan bahwa *random forest* memberikan performa terbaik.

Pengimplementasian model *machine learning* dengan kumpulan data yang sesuai dapat membantu proses diagnosis diabetes dan menemukan model pengklasifikasian berbasis *machine learning* yang optimal. Dalam penelitian ini, algoritma yang dilatih meliputi *decision tree*, *naïve bayes*, *k-nearest neighbor*, *random forest*, *gradient boosting*, *logistic regression*, dan *support vector machine*. Hasil penelitian menunjukkan bahwa algoritma *random forest*, dengan preprocessing yang tepat pada data klinis dan penerapan klasifikasi berbasis *machine learning*, mampu memprediksi diabetes secara akurat dan efisien (Ahmed dkk., 2021). Model *machine learning* digunakan untuk menemukan pola dalam data dan menghasilkan prediksi yang dapat diandalkan. Tujuan penelitian lainnya adalah menerapkan berbagai algoritma *machine learning* untuk memprediksi diagnosis diabetes. Dalam penelitian ini, dataset yang digunakan adalah Diabetes Disease dari *Behavioral Risk Factor Surveillance System* (BRFSS), yang mencakup data populasi dalam skala besar di Amerika Serikat. Berdasarkan model yang diterapkan, *random forest* kembali menunjukkan performa terbaik dengan akurasi sebesar 82,26%, mengungguli algoritma lainnya (Chang dkk., 2022).

Meskipun *random forest* terbukti unggul, kualitas prediksinya dapat terpengaruh oleh ketidakseimbangan data dalam kumpulan data medis. Untuk mengatasi masalah ini, metode penyeimbangan data seperti ADASYN digunakan karena dapat secara signifikan meningkatkan kinerja model *machine learning* (Mitra dkk., 2023). Dalam sebuah penelitian, rekam medis dari 500 dataset pasien diabetes dengan kelas yang tidak seimbang dianalisis. Tujuan utama penelitian ini adalah untuk menghasilkan model prediktif berkinerja tinggi untuk nefropati. Penelitian ini menyarankan penggunaan ADASYN, yang sangat relevan dalam metode *machine learning* seperti *random forest*, *adaboost*, dan *bagging* (adabag). Teknik ADASYN mampu meningkatkan jumlah sampel pada kelas minoritas dan mengurangi bias dalam kumpulan data yang tidak seimbang (Muflikhah dkk., 2024). Dengan mengatasi ketidakseimbangan kelas minoritas dalam data asli,

penerapan ADASYN terbukti dapat meningkatkan kinerja deteksi dan prediksi model *machine learning* (Kim dkk., 2021).

Selain untuk deteksi diabetes, pendekatan *machine learning* yang dilengkapi dengan SHAP juga telah diterapkan dalam berbagai studi prediksi penyakit lainnya seperti *Acute Ischemic Stroke* (AIS) dan *CardiovascularDisease* (CVD). Salah satu penelitian yang dilakukan oleh M. Fu dkk (2025) bertujuan untuk mengidentifikasi faktor-faktor yang paling berpengaruh dalam memprediksi AIS. Metode SHAP digunakan untuk memberikan interpretasi baik secara lokal maupun global terhadap hasil model. Hasil studi tersebut menunjukkan bahwa model yang dikembangkan mampu memberikan performa prediktif yang baik dan dapat digunakan untuk mendukung pengambilan keputusan klinis yang lebih terarah. Penelitian lain Q. Fu dkk. (2024) berfokus pada pengembangan model prediksi risiko *CardiovascularDisease* (CVD) menggunakan data dari Survei Pemeriksaan Kesehatan dan Gizi Nasional (NHANES) tahun 2011-2018. Enam algoritma diuji, yaitu *random forest*, *light gradient boosting machine*, *decision tree*, *extreme gradient boosting*, *multi-layer perceptron*, dan *support vector machine*. Hasilnya menunjukkan bahwa algoritma *random forest* memberikan kinerja prediksi terbaik dengan nilai ROC sebesar 0.814. SHAP digunakan untuk menjelaskan kontribusi fitur terhadap *output* prediksi, baik secara keseluruhan maupun pada tingkat individu. Sehingga mempermudah interpretasi hasil model secara mendalam (Escriva dkk., 2025).

Menanggapi tren tersebut, Khokhar dkk. (2025) dalam tinjauan sistematis mereka secara tegas mengkritisi kecenderungan sebagian besar penelitian yang tidak melaporkan proses pengolahan data secara transparan. Mereka menemukan bahwa aspek penting seperti penyeimbangan kelas, dan interpretabilitas model masih sering diabaikan. Oleh karena itu, mereka merekomendasikan agar penelitian-penelitian selanjutnya mulai mengintegrasikan pendekatan *explainable AI* serta teknik penyeimbangan data guna meningkatkan keandalan model dan memperkuat kepercayaan pengguna, khususnya dalam konteks aplikasi klinis.

Berdasarkan tinjauan pustaka yang telah dilakukan, penelitian ini memiliki beberapa unsur kebaruan. Penelitian ini menerapkan teknik oversampling seperti

ADASYN untuk mengatasi ketidakseimbangan data pada kasus klasifikasi diabetes, yang masih jarang diaplikasikan secara menyeluruh dalam literatur sebelumnya. Model yang dibangun menggunakan algoritma *random forest* tidak hanya dievaluasi dari sisi performa, tetapi juga dianalisis menggunakan metode interpretasi SHAP untuk memberikan pemahaman mengenai kontribusi pada setiap fitur. Penelitian ini juga mengeksplorasi hasil klasifikasi berdasarkan subkelompok pasien seperti usia dan BMI, sehingga mendekati personalisasi prediksi diabetes. Temuan penelitian ini diinterpretasikan dengan merujuk pada literatur medis yang telah divalidasi sebelumnya, sehingga dapat memberikan relevansi klinis meskipun belum divalidasi secara langsung oleh praktisi medis.

## 2.2 Dasar Teori

### 2.2.1 Sejarah Dataset Pima Indian Diabetes

Dataset Pima Indians Diabetes berasal dari sebuah studi epidemiologis jangka panjang yang dilakukan oleh National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), yang merupakan bagian dari National Institute of Health (NIH) di Amerika Serikat. Studi ini dimuai pada tahun 165 dan difokuskan pada komunitas Pima Indian yang tinggal di wilayah Arizona bagian selatan, yaitu kelompok penduduk asli Amerika yang secara historis hidup dari pertanian dan aktivitas fisik yang tinggi. Namun, sejak pertengahan abad ke-20, komunitas ini mengalami perubahan gaya hidup secara drastis dari yang sebelumnya aktif dan konsumsi makanan sehat menjadi gaya hidup yang minim aktivitas dan konsumsi makanan olahan tinggi kalori. Perubahan ini menyebabkan kasus obesitas meningkat dan kasus diabetes. Dalam konteks inilah, suku Pima Indian menjadi populasi penting untuk diteliti, karena prevalensi diabetes di komunitas ini merupakan salah satu yang tertinggi di dunia.

Sebagai bagian dari penelitian medis, data dikumpulkan melalui serangkaian pemeriksaan terhadap wanita suku Pima Indian yang berusia minimal 21 tahun dan tidak sedang hamil. Hasil dari pengumpulan data ini kemudian dirangkum menjadi dataset yang saat ini dikenal sebagai Pima Indians Diabetes Dataset. Dataset ini pertama kali diperkenalkan secara luas melalui publikasi oleh

(Smith dkk., 1988) dalam simposium komputerisasi aplikasi medis dan juga didistribusikan melalui UCI *Machine Learning Repository*, yang menjadikannya tersedia secara terbuka bagi komunitas ilmiah dan pengembang teknlogi. Sejak saat itu, dataset ini telah menjadi salah satu *benchmark* yang populer dalam bilang machine learning dan klasifikasi medis, karena memiliki karakteristik yang mendukung pengembangan serta evaluasi berbagai algoritma prediksi. Meskipun data ini telah dikumpulkan beberapa dekade yang lalu, data ini tetap relevan dan sering digunakan hingga saat ini, baik dalam penelitian akademik maupun pelatihan sistem kecerdasan buatan, karena kestandaran datanya, kejelasan strukturnya, serta latar belakang medis dunia nyata yang kuat.

### 2.2.2 Penyakit Diabetes

Diabetes merupakan penyakit gangguan metabolik yang disebabkan oleh pankreas (organ di belakang perut) yang memproduksi sedikit insulin atau tidak memproduksi insulin sama sekali. Insulin adalah hormon alami yang dihasilkan oleh sel beta pankreas dan berfungsi membantu tubuh menggunakan gula sebagai sumber energi. Diabetes ditandai oleh gejala kadar gula darah yang tinggi dalam jangka waktu yang lama. Kondisi ini terjadi karena tubuh tidak memproduksi cukup hormon insulin, sehingga kadar gula dalam aliran darah meningkat. Diabetes terbagi menjadi dua jenis, yaitu diabetes tipe 1 dan tipe 2. Diabetes tipe 1 terjadi ketika tubuh tidak memproduksi insulin sama sekali. Penderita diabetes tipe 1 membutuhkan pengobatan berupa insulin, baik melalui injeksi maupun pompa insulin. Sementara itu, diabetes tipe 2 disebabkan oleh resistensi insulin, yaitu kondisi di mana sel-sel tubuh tidak dapat menggunakan insulin dengan baik. Pengobatan untuk diabetes tipe 2 mencakup kombinasi diet, olahraga, obat oral, atau kombinasi dari semuanya (Gunawan dkk., 2020). Beberapa faktor risiko diabetes meliputi usia, jenis kelamin, sering buang air kecil, sering merasa haus, penurunan berat badan secara tiba-tiba, kelemahan, penyembuhan luka yang lambat, dan obesitas. Diabetes yang tidak terdiagnosis atau tidak diobati dapat menyebabkan kerusakan pada organ vital, seperti mata, ginjal, saraf, jantung, dan kaki, serta berisiko menyebabkan kematian (Wardhani dan Akbar, 2022).

Menurut Federasi Diabetes Internasional (2021), jumlah kematian akibat diabetes mencapai 6,7 juta pada tahun 2021. Sebanyak 537 juta orang dalam rentang usia 20 hingga 79 tahun hidup dengan diabetes. Jumlah ini diperkirakan akan meningkat menjadi 643 juta pada tahun 2030 dan 783 juta pada tahun 2045. Berdasarkan artikel yang diterbitkan oleh Kementerian Kesehatan Republik Indonesia pada tahun 2020, Indonesia termasuk salah satu dari sepuluh negara dengan tingkat diabetes tertinggi pada tahun 2019 (Febrian dkk., 2022).

### 2.2.3 Konsep Dasar Klasifikasi

Klasifikasi adalah metode analisis data yang bertujuan untuk mengidentifikasi dan mengelompokkan objek atau gagasan ke dalam kategori tertentu yang telah ditentukan sebelumnya. Proses kerja klasifikasi terdiri dari dua langkah utama. Langkah pertama adalah langkah pembelajaran, di mana model klasifikasi dibangun berdasarkan data sebelumnya. Langkah kedua adalah langkah klasifikasi, di mana model yang telah dibuat digunakan untuk memprediksi label kelas pada data baru. Klasifikasi memiliki berbagai manfaat, seperti dalam diagnosis penyakit, deteksi penipuan, dan penentuan target pemasaran. Proses klasifikasi umumnya dilakukan dalam dua tahap. Pada tahap pertama, model klasifikasi dikembangkan berdasarkan data historis. Selanjutnya, pada tahap kedua, dilakukan evaluasi untuk menentukan apakah akurasi model sudah dapat diterima. Jika hasilnya memadai, model tersebut dapat digunakan untuk mengklasifikasikan data baru. Sebagai contoh, seorang peneliti medis ingin menganalisis data kanker payudara untuk memprediksi jenis perawatan yang harus diterima oleh pasien. Dalam hal ini, tugas klasifikasi adalah memprediksi label kelas, seperti perawatan A, perawatan B, atau perawatan C (Han dkk., 2012).



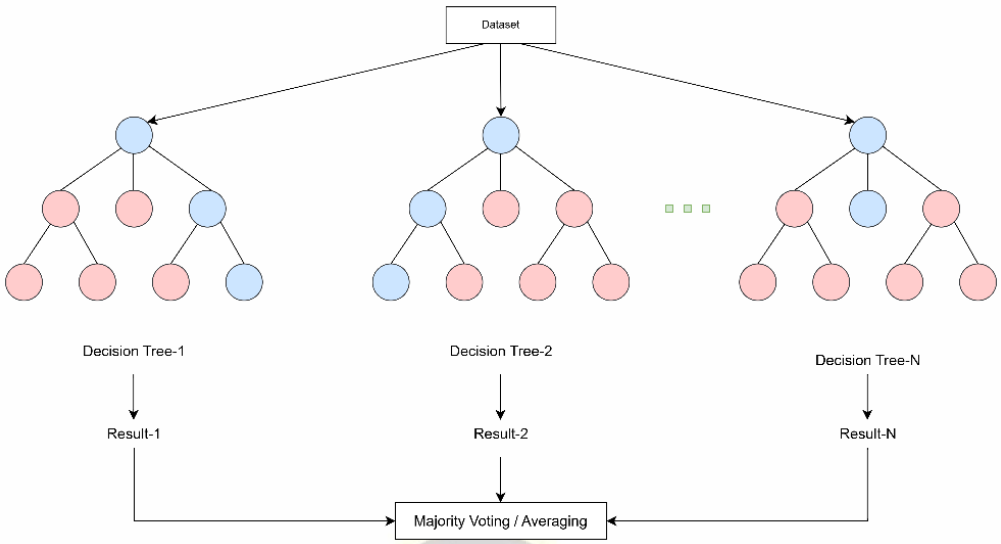
### 2.2.4 Algoritma *Decision Tree*

*Decision tree* adalah algoritma yang digunakan untuk klasifikasi dan regresi dengan cara membagi data secara berulang berdasarkan aturan tertentu hingga diperoleh hasil yang optimal. Algoritma *random forest* yang digunakan pada penelitian ini merupakan pengembangan dari algoritma *decision tree.* Pada subbab ini, penjelasan difokuskan hanya pada *decision tree* untuk keperluan klasifikasi.

Model ini bersifat nonparametrik, sehingga tidak memerlukan asumsi distribusi normal pada data, serta dapat secara otomatis memilih variabel yang paling berpengaruh dan mengabaikan variabel yang kurang relevan. Selain itu, *decision tree* mampu menangani data campuran, nilai yang hilang, serta outlier, menjadikannya fleksibel untuk berbagai jenis analisis. Cara kerjanya dimulai dengan menentukan simpul akar yang menjadi variabel utama dalam pembagian data, kemudian cabang-cabang terbentuk berdasarkan aturan pemisahan tertentu, seperti *Information Gain*, hingga mencapai simpul daun yang berisi hasil akhir klasifikasi atau prediksi. *Decision tree* juga dikenal karena hasil visualisasinya yang mudah dipahami, sehingga banyak digunakan dalam penelitian kesehatan untuk mengidentifikasi faktor risiko penyakit, seperti diabetes, dengan menganalisis hubungan antara berbagai variabel kesehatan. Dengan kemampuannya dalam menemukan pola kompleks, *decision tree* menjadi alat yang sangat berguna dalam analisis data dan pengambilan keputusan berbasis data (Cuesta dkk., 2019).

### 2.2.5 Algoritma *Random Forest*

*Random forest* adalah metode pembelajaran *ensemble* yang menggunakan subset acak dari prediktor dan pengamatan untuk membangun pohon keputusan pada setiap iterasi. Prediksi dari setiap pohon kemudian digabungkan untuk membentuk sebuah hutan (Ohanyan dkk, 2022). *Random forest* mampu menangani data numerik maupun kategoris dengan mudah. Salah satu keunggulan utama algoritma ini adalah kemampuannya untuk menghindari overfitting, bahkan ketika jumlah pohon dalam hutan ditingkatkan (Kumar dkk., 2021). Selain itu, *random forest* memiliki kemampuan untuk secara otomatis menghasilkan nilai akurasi, yang sangat penting dalam mengklasifikasikan kesalahan dan meningkatkan keandalan prediksi (Balaram dan Vasundra, 2022).




Gambar 2.1 *Random Forest*

Gambar 2.1 di atas merupakan gambaran cara kerja *random forest* (Saxena dkk., 2022).

a. Sampel acak dipilih dari dataset yang tersedia.
b. Pohon keputusan dibangun untuk setiap sampel, dan prediksi dibuat berdasarkan masing-masing pohon keputusan.
c. Setiap hasil prediksi akan melalui proses pemungutan suara.
d. Hasil dengan jumlah suara terbanyak akan menjadi prediksi akhir.

Entropi dari himpunan kasus $S$ adalah ukuran ketidakpastian atau keragaman dalam sebuah himpunan kasus $(S)$. Dalam konteks teori informasi, entropi dihitung dengan menggunakan rumus pada persamaan (2.1) yaitu:

$$Entropy(S) = \sum_{i=1}^{n}(-P_i \times log_2 P_i) \quad (2.1)$$

dimana $n$ adalah jumlah partisi dari himpunan $S$ dan $P_i$ adalah proporsi atau probabilitas partisi ke-$i$ terhadap keseluruhan himpunan $S$. Proporsi $P_i$ dihitung dengan membagi jumlah elemen partisi $S_i$ dengan jumlah elemen dalam $S$. Logaritma yang digunakan dalam perhitungan memiliki basis 2 $(log_2)$, sehingga satuan entropi yang dihasilkan adalah bit dimana satuan bit menunjukkan jumlah rata-rata keputusan biner yang diperlukan untuk memprediksi suatu keluaran berdasarkan distribusi probabilitasnya.

*Information Gain* $(S, A)$ adalah metrik yang digunakan dalam *machine learning*, khususnya dalam algoritma seperti pohon keputusan, untuk menentukan seberapa baik suatu variabel $(A)$ dapat memisahkan himpunan kasus $(S)$ berdasarkan entropi. Rumusnya terdapat pada persamaan 2.2 yaitu:

$$Gain\ (S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \times Entropy\ (S_i) \quad (2.2)$$

*Information* yang diperoleh $(Gain\ (S, A))$ mengukur penurunan dalam entropi setelah data dibagi berdasarkan variabel $(A)$. Nilai pertama dalam rumus yaitu $Entropy(S)$, mempresentasikan ketidakpastian awal dalam $S$. Kemudian, pengurangan entropi dihitung sebagai jumlah rata-rata bobot entropi dari masing masing partisi $S_i$ dengan bobot berdasarkan proporsi ukuran partisi $(|S_i|/|S|)$.

Berikut merupakan contoh sederhana langkah-langkah perhitungan *random forest*. Menggunakan dataset dengan 4 pasien yang disajikan pada Tabel 2.1 untuk memprediksi apakah seseorang menderita diabetes (1) atau tidak (0) berdasarkan dua fitur yaitu, Glukosa dan Usia.

Tabel 2.1 Dataset Diabetes

| ID | Glukosa (mg/dL) | Usia | Diabetes |
|---|---|---|---|
| 1 | 150 | 35 | 1 |
| 2 | 180 | 45 | 1 |
| 3 | 85 | 25 | 0 |
| 4 | 90 | 30 | 0 |

Dalam perhitungan *entropy* dan *information gain*, pertama-tama kita menghitung probabilitas kelas pada dataset yang diberikan. Misalnya, kita memiliki dataset dengan label diabetes yang terdiri dari 4 sampel, dua di antaranya adalah pasien diabetes (label = 1) dan dua lainnya adalah non-diabetes (label = 0). Oleh karena itu, probabilitas untuk kelas diabetes dan non-diabetes adalah:

$$P(Diabetes = 1) = \frac{2}{4} = 0.5$$

$$P(Diabetes = 0) = \frac{2}{4} = 0.5$$

Kemudian, dihitung entropynya menggunakan rumus (2.1) yang menghasilkan

$$H(S) = -(P_1 \times Log_2 P_1 + P_0 \times Log_2 P_0)$$
$$= -\left(0.5 \times Log_2 0.5 + 0.5 \times Log_2 0.5\right) = 1.0.$$

Hitung *information gain* untuk fitur-fitur yang ada, dimulai dengan fitur "Glukosa". Pada fitur ini, jika nilai glukosa lebih besar dari 120, maka semua label adalah 1 (diabetes), yang berarti *entropy* untuk subset ini adalah 0. Begitu juga jika nilai glukosa kurang dari atau sama dengan 120, seluruh label adalah 0 (non-diabetes), yang juga memiliki entropy 0.

$$IG(Glukosa) = H(S) - \left(\frac{2}{4} \times 0 + \frac{2}{4} \times 0\right)$$

$$IG(Glukosa) = 1.0 - 0 = 1.0$$

Selanjutnya, menghitung *information gain* untuk fitur "Usia". Pada subset dengan usia lebih besar dari 40, hanya terdapat 1 label diabetes (entropy = 0). Sedangkan pada subset usia kurang dari atau sama dengan 40, terdapat 3 sampel, dengan 1 label diabetes dan 2 label non-diabetes, sehingga kita perlu menghitung entropy untuk subset ini. Entropy dihitung dengan rumus 2.1.

$$H(Usia \leq 40) = -\left(\frac{1}{3} \times log_2 \frac{1}{3} + \frac{2}{3} \times log_2 \frac{2}{3}\right)$$

Untuk menghitung Information Gain (IG) pada fitur Usia dengan rumus 2.2.

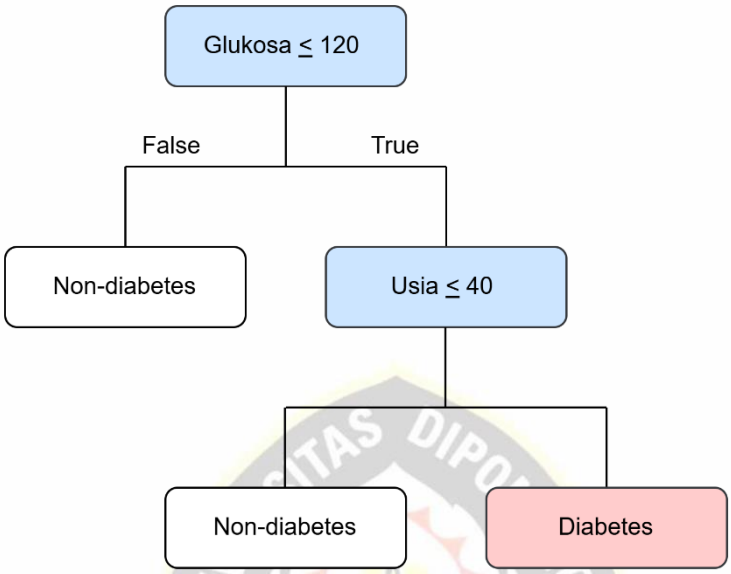
$$IG(Usia) = H(S) - \left(\frac{1}{4} \times 0 + \frac{3}{4} \times 0.918\right)$$



$$IG(Usia) \approx 1.0 - 0.689 = 0.311$$

akhirnya, didapat fitur dengan *information gain* tertinggi untuk digunakan sebagai fitur utama dalam pemisahan data. Dalam hal ini, fitur "Glukosa" memiliki information gain sebesar 1.0, sedangkan fitur "Usia" memiliki information gain sebesar 0.311. Oleh karena itu, fitur "Glukosa" dipilih sebagai fitur pertama untuk pemisahan pada model. Dengan demikian, Glukosa adalah fitur yang paling berpengaruh dalam membuat keputusan pemisahan data untuk prediksi diabetes pada dataset ini. Gambar 2.2 menunjukkan struktur pohon keputusan yang

terbentuk berdasarkan dataset simulasi. Fitur *Glucose* dipilih sebagai fitur utama karena menghasilkan *Information Gain* sebesar 1.0.




Gambar 2.2 Pohon Keputusan Berdasarkan Fitur Glukosa dan Usia

### 2.2.6 Teknik Penyeimbang Data dengan ADASYN

ADASYN merupakan versi yang ditingkatkan dari teknik *Synthetic Minority Oversampling Technique* (SMOTE), yang digunakan untuk menghindari *overfitting* ketika replika identik dari data minoritas ditambahkan ke dalam kumpulan data utama. Ide utama dari ADASYN adalah menggunakan kriteria distribusi kepadatan untuk secara otomatis menentukan jumlah sampel sintetis yang sesuai yang perlu dihasilkan untuk setiap contoh data minoritas (Xu dkk., 2020).

Untuk mengilustrasikan cara kerja ADASYN, berikut contoh sederhana dengan data berukuran kecil. Jika dataset tidak seimbang, kelas diabetes (1) memiliki lebih banyak data dibandingkan dengan kelas non-diabetes (0), maka ADASYN akan menghasilkan data sintetis untuk membantu menyeimbangkan distribusi kelas tersebut. Misalnya terdapat dataset dengan distribusi tidak seimbang seperti pada Tabel 2.2, dimana kelas diabetes (1) terdiri dari dua data, sementara kelas non-diabetes (0) hanya terdiri dari satu data.

Tabel 2.2 Dataset Diabetes yang Tidak Seimbang

| ID | Glukosa (mg/dL) | Usia | Diabetes |
|---|---|---|---|
| 1 | 150 | 35 | 1 |
| 2 | 180 | 45 | 1 |
| 3 | 85 | 25 | 0 |

Proporsi ketidakseimbangan dapat dihitung menggunakan rumus 2.3 dibawah ini.

$$r = \frac{Jumlah\ data\ kelas\ minoritas}{Jumlah\ data\ mayoritas} = \frac{1}{2} = 0.5 \quad (2.3)$$

Nilai rasio $r$ yang lebih kecil dari 1 menunjukkan adanya ketidakseimbangan pada distribusi kelas. Dalam hal ini, ADASYN akan menghasilkan sampel sintetis baru dari kelas minoritas menggunakan proses interpolasi terhadap data minoritas yang ada. Misalnya, jika data asli dari kelas minoritas memiliki nilai $Glukosa = 85,\ Usia = 25$, Maka data sintetis dari ADASYN akan menghasilkan $Glukosa = 87, Usia = 27$.

Tabel 2.3 menunjukkan hasil penyeimbang dataset setelah diterapkan metode ADASYN, dimana jumlah data untuk kelas minoritas menjadi seimbang dengan kelas mayoritas.

Tabel 2. 3 Hasil Penyeimbangan Dataset Menggunakan ADASYN

| ID | Glukosa (mg/dL) | Usia | Diabetes |
|---|---|---|---|
| 1 | 150 | 35 | 1 |
| 2 | 180 | 45 | 1 |
| 3 | 85 | 25 | 0 |
| 4 | 87 (sintetis) | 27 (sintetis) | 0 |



Dengan pendekatan ini, ADASYN tidak hanya membantu menyeimbangkan jumlah kelas dalam dataset, tetapi juga meningkatkan kemampuan model untuk belajar dari area yang sulit diprediksi, sehingga diharapkan performa klasifikasi terhadap kelas minoritas menjadi lebih baik.

Data sintetis yang dihasilkan oleh ADASYN dibentuk melalui proses interpolasi antara data minoritas yang ada dengan tetangga terdekatnya dari kelas yang sama. Proses ini dilakukan dengan prinsip interpolasi linier yang menghasilkan titik baru diantara dua titik data minoritas. Secara matematis,

interpolasi ADASYN dinyatakan dengan rumus 2.4 (Chen dkk., 2024) sebagai berikut:

$$x_{new} = x_i + |x_i - x_{zi}| \times \beta \quad (2.4)$$

dimana $x_i$ merupakan vektor fitur dari data asli yang berasal dari kelas minoritas, sedangkan $x_{zi}$ adalah tetangga terdekat dari kelas minoritas yang dipilih melalui algoritma *k-nearest neighbors.* Parameter $\beta$ adalah bilangan acak yang bernilai antara 0 dan 1, yang digunakan untuk menentukan posisi titik sintetis antara $x_i$ dan $x_{zi}$. Nilai $\beta$ yang bervariasi memungkinkan proses interpolasi menghasilkan data sintetis yang menyebar secara acak di sekitar data minoritas asli, sehingga membantu meningkatkan keberagaman dan representasi kelas minoritas dalam dataset yang telah diseimbangkan.

Sebagai contoh, jika titik data minoritas adalah $Glukosa = 85,\ Usia = 25$, serta tetangganya $Glukosa = 90,\ Usia = 30$, maka dengan $\beta = 0.4$, akan dihasilkan titik sintetis sebagai berikut:

$$Glukosa = 85 + 0.4\,(90 - 85) = 87$$
$$Usia = 25 + 0.4\,(30 - 25) = 27$$

Nilai $\beta$ tersebut dipilih secara acak dan akan berbeda pada setiap proses penambahan data sintetis, guna menghasilkan distribusi data baru yang bervariasi namun masih dalam batas wajar data asli. Pemilihan tetangga terdekat $x_j$ secara eksklusif terbatas pada titik-titik dari kelas minoritas. Hal ini dikarenakan ADASYN hanya bertujuan memperkuat representasi kelas minoritas dengan menambahkan data sintetis di sekitar titik-titik minoritas yang sulit dipelajari. Oleh karena itu, data seperti $Glukosa = 180, Usia = 45$ yang berasal dari kelas mayoritas tidak digunakan dalam proses interpolasi. Sebaliknya, data $Glukosa = 90,\ Usia = 30$ dipilih sebagai tetangga karena berada pada kelas yang sama (non-diabetes) dan memiliki kedekatan dalam ruang fitur.

Dengan pendekatan ini, ADASYN tidak hanya menyeimbangkan distribusi kelas, tetapi juga meningkatkan kemampuan model klasifikasi dalam mengenali pola kompleks pada kelas minoritas yang sering terabaikan oleh algoritma pembelajaran tradisional.

### 2.2.7 Teknik Interpretabilitas dengan SHAP

SHAP merupakan teknik visualisasi yang dapat menjelaskan sejauh mana setiap fitur berkontribusi pada setiap prediksi model, baik secara positif maupun negatif. Plot kepentingan fitur digunakan untuk menampilkan fitur yang memiliki dampak paling signifikan terhadap setiap prediksi model. Dua sampel dipilih untuk membuat grafik nilai SHAP, yang digunakan dalam memprediksi dan menafsirkan model pada sampel tunggal (J. Xu dkk., 2024).

Secara konseptual, SHAP memandang fitur sebagai pemain dalam suatu permainan kooperatif, sedangkan output model (misalnya probabilitas diabetes) dianggap sebagai nilai hadiah yang dibagi di antara para pemain. Nilai SHAP untuk setiap fitur dihitung berdasarkan kontribusi rata-rata marjinal fitur tersebut terhadap semua kemungkinan subset fitur lainnya. Rumus umum nilai Shapley 2.5 (Nourani dkk., 2025) dapat dinyatakan sebagai berikut:

$$\emptyset_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[ g_{S \cup \{i\}}(x_{S \cup \{i\}}) - g_S\,(x_S) \right] \quad (2.5)$$

Dalam rumus nilai *Shapley* yang digunakan pada metode SHAP, simbol $\emptyset_i$ mempresentasikan nilai SHAP dari fitur ke-$i$, yaitu besarnya kontribusi fitur tersebut terhadap hasil prediksi model. Subset $F$ melambangkan seluruh fitur yang ada dalam model, sementara $S$ adalah subset dari fitur $F$ yang tidak mencakup fitur $i$. Fungsi $g_S\,(x_S)$ adalah nilai prediksi model saat hanya menggunakan fitur dalam subset $S$, dan $g_{S \cup \{i\}}(x_{S \cup \{i\}})$ adalah prediksi saat fitur $i$ ditambahkan ke dalam subset tersebut. Selisih antara keduanya mencerminkan kontribusi marjinal fitur $i$ terhadap prediksi. Nilai kontribusi ini dihitung untuk semua subset $S$ yang memungkinkan dan kemudian rata-rata dengan bobot kombinatorial agar adil terhadap semua kemungkinan urutan fitur.

Perhitungan ini mempertimbangkan seluruh kemungkinan kombinasi subset fitur, namun karena kompleksitasnya eksponensial, SHAP mengadopsi algoritma efisien seperti TreeSHAP pada model pohon keputusan yang dapat menghitung nilai SHAP secara tepat dalam waktu polinomial (Pang, 2025).

Sebagai contoh, misalnya *random forest* telah dilatih memprediksi probabilitas diabetes berdasarkan beberapa fitur. Untuk prediksi seorang pasien

dengan Glukosa = 160, Usia = 40. Model memprediksi diabetes (1) dengan probabilitas 0,8. Jika nilai rata-rata prediksi semua pasien (*baseline)* adalah 0.5, maka selisih prediksi akhir terhadap *baseline* adalah 0,3. SHAP kemudian menghitung kontribusi setiap fitur terhadap prediksi. Rata-rata baseline probabilitas dengan rumus 2.6 yaitu, $Baseline = P(Diabetes) = 0.5$. Kontribusi fitur yaitu, Glukosa (160): +25% dan Usia (40): +5%. Prediksi akhir:

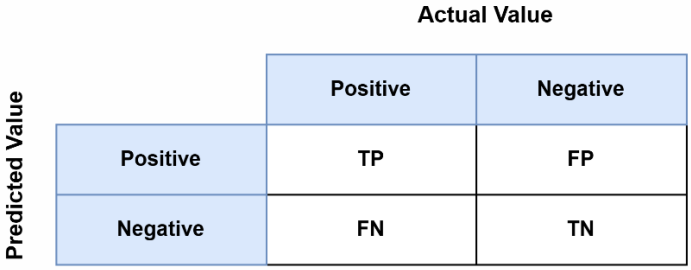
$$P(Diabetes) = Baseline + Glukosa + Usia \tag{2.6}$$

$$P(Diabetes) = 0.5 + 0.25 + 0.05 = 0.8\ (80\%)$$

Nilai-nilai ini kemudian divisualisasikan dalam bentuk *force plot* atau *waterfall plot* yang menunjukkan fitur mana yang mendorong model menuju prediksi positif atau negatif. SHAP memberikan interpretasi dalam dua tingkat, yaitu interpretasi global, yang merangkum fitur-fitur yang paling berpengaruh secara keseluruhan terhadap model, dan interpretasi lokal, yang menjelaskan kontribusi fitur terhadap hasil prediksi secara individual. Dengan demikian, SHAP memberikan transparansi terhadap model yang kompleks, menjadikannya alat yang penting dalam bidang medis dan aplikasi lain yang membutuhkan kepercayaan pengguna terhadap model.

### 2.2.8 Evaluasi *Confusion Matrix*

*Confusion matrix* merupakan salah satu metode yang dapat melakukan perbandingan untuk menganalisis seberapa baik model klasifikasi dalam mengidentifikasi data. Terdapat empat nilai yang dihasilkan *Confusion Matrix* seperti pada gambar 2.2*,* yaitu *True Positive* (TP) dan *True Negative* (TN) yang menunjukan jika model klasifikasi menghasilkan prediksi kelas yang benar. Lalu ada *False Positive* (FP) dan *False Negative* yang menunjukkan jika model prediksi kelas yang dihasilkan bernilai salah. Empat nilai yang telah dijelaskan dapat digunakan untuk menghitung berbagai ukuran baru untuk mengevaluasi model klasifikasi (Han dkk., 2012).




Gambar 2.3 *Confusion Matrix*

Evaluasi model klasifikasi dapat dihitung dengan empat nilai pada confusion matrix. Nilai-nilai ukuran evaluasi model klasifikasi yaitu akurasi, precison, *recall, dan* $F1 - score$ dari performa suatu metode.

1. Akurasi mengukur proporsi prediksi yang benar dibandingkan dengan jumlah total prediksi. Akurasi adalah metrik yang sederhana untuk mengevaluasi kinerja model. Perhitungan nilai akurasi terdapat pada Rumus (2.7).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.7)$$

2. *Precision* menunjukkan sejauh mana prediksi positif bener-benar relevan. Semakin tinggi nilai *precision*, semakin sedikit kesalahan prediksi positif yang dilakukan model. Nilai *precision* dapat dihitung berdasarkan Rumus (2.8).

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (2.8)$$



3. *Recall* atau disebut juga dengan sensitivitas menunjukkan seberapa baik model menangkap semua contoh positif. Semakin tinggi nilai *recall,* semakin sedikit data positif yang terlewat oleh model. Nilai *recall* dapat dihitung berdasarkan Rumus (2.9).

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.9)$$

4. F1-Score adalah ukuran kinerja model yang membantu menyeimbangkan *precision* dan *recall*. Semakin tinggi nilai F1-score, semakin baik model dalam

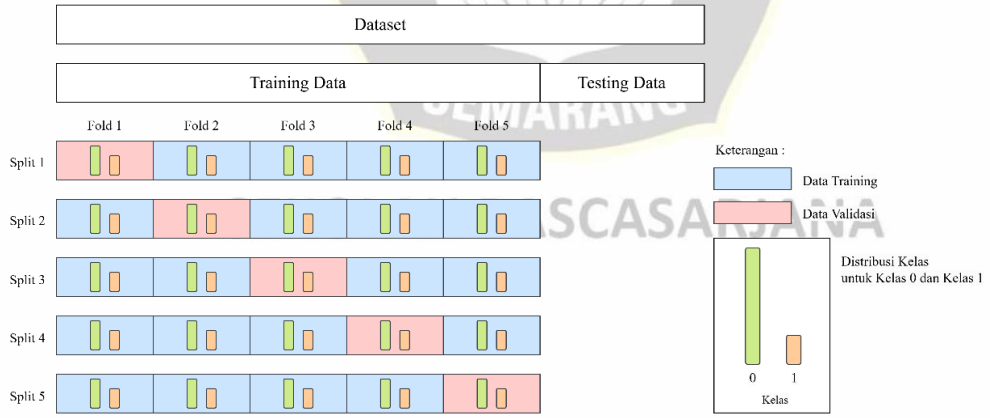
menyeimbangkan *precision* dan *recall*. Perhitungan nilai akurasi terdapat pada Rumus (2.10).

$$F1 - score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (2.10)$$

### 2.2.9 *Stratified K-fold Cross Validation*

*Cross Validation* adalah teknik evaluasi model di mana dataset dibagi berulang kali menjadi dua bagian, *training set* yang digunakan untuk melatih model dan *validation set* yang digunakan untuk mengukur performa model. Setiap kali pembagian dilakukan, model diuji, dan hasil evaluasi dari berbagai iterasi dirata-ratakan untuk mendapatkan estimasi performa akhir (Leinonen dkk., 2024).

Dalam penelitian ini, digunakan *stratified k-fold cross validation* yang merupakan pengembangan dari metode *cross validation*. Teknik ini memastikan bahwa proporsi kelas dalam setiap fold tetap sama seperti dataset asli. Oleh karena itu, metode ini sangat bermanfaat untuk dataset yang tidak seimbang, seperti pada kasus prediksi diabetes, di mana jumlah pasien sehat (kelas 0) jauh lebih banyak dibandingkan pasien yang menderita diabetes (kelas 1). Gambar 2.3 menunjukkan cara kerja *stratified k-fold cross validation.*

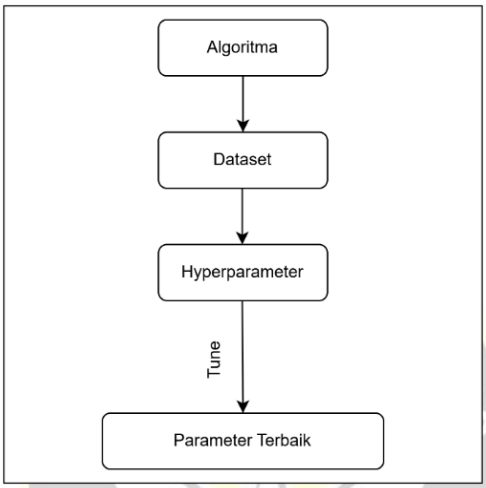



Gambar 2.4 *Stratified K-Fold Cross-Validation*

### 2.2.10 Optimalisasi *Hyperparameter Tuning*

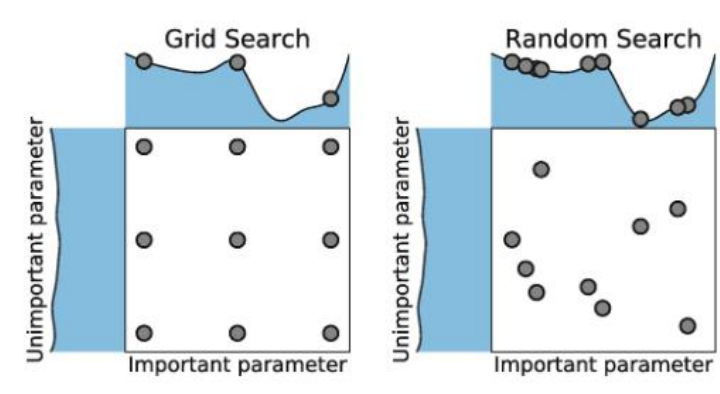
Untuk optimalisasi hyperparameter dalam algoritma, berbagai strategi telah ditemukan. Hyperparameter tuning adalah teknik pengembangan model *machine*

*learning*, karena dapat mempengaruhi kinerja keseluruhan model. *Hyperparameter* merujuk pada serangkaian parameter yang harus dipilih secara tepat untuk mengoptimalkan algoritma atau model tertentu. Pemilihan *hyperparameter* yang tepat sangat penting karena dapat meningkatkan akurasi model dan menghindari *overfitting* atau u*nderfitting* seperti yang dapat dilihat pada Gambar 2.4 (Nguyen dan Liu, 2025).




Gambar 2.5 *Hyperparameter*

*GridSearch* atau *random search* adalah dua metodologi yang sering digunakan sebagai dasar penelitian, seperti yang ditunjukkan pada Gambar 2.5.




Gambar 2.6. *GridSearch* dan *Random Search*

*GridSearch* merupakan teknik pencarian *hyperparameter* secara menyeluruh dengan mencoba semua kemungkinan kombinasi dari nilai-nilai parameter yang

telah ditentukan. Proses ini dilakukan dengan membuat *grid* atau kisi-kisi dari semua nilai yang mungkin, lalu model akan dilatih dan divalidasi pada setiap kombinasi tersebut. Meskipun menjamin menemukan kombinasi terbaik dalam ruang pencarian yang terbatas, *GridSearch* sangat memakan waktu dan sumber daya komputasi, terutama jika hyperparameter dan nilai yang diuji cukup besar (Siregar dkk., 2024).

Sebagai contoh sederhana, jika ingin melakukan optimasi *hyperparameter* pada algoritma *GridSearch*, khususnya terhadap dua parameter utama yaitu *n_estimators* dan *max_depth*. Dalam pendekatan *GridSearch*, akan ditentukan terlebih dahulu beberapa nilai kandidat untuk masing-masing *hyperparameter*. Misalnya *n_estimators* diatur untuk diuji pada nilai $[50, 100, 150]$, dan *max_depth* pada nilai $[3, 5, 7]$. Maka, *GridSearch* akan mengevaluasi semua kemungkinan kombinasi dari dua set nilai tersebut, sehingga menghasilkan total $3 \times 3 = 9$ kombinasi yang berbeda. Setiap kombinasi akan digunakan untuk melatih model, dan performanya akan dievaluasi menggunakan teknik *cross validation.* Model dengan kombinasi nilai hyperparameter yang menghasilkan performa terbaik akan dipilih sebagai model akhir.

Sebaliknya, *Random Search* merupakan teknik yang memilih kombinasi hyperparameter secara acak dari ruang pencarian yang telah ditentukan. Metode ini tidak menguji semua kombinasi, melainkan hanya sejumlah iterasi tertentu yang diambil secara acak. Misalnya *n_estimators* ditetapkan berada dalam rentang 50 hingga 150, dan *max_depth* antara 3 hingga 7. Daripada menguji semua kombinasi, random search hanya memilih sejumlah kombinasi acak, seperti lima kombinasi, untuk diuji. Contohnya (*n_estimators = 60, max_depth = 3*), (n*_estimators = 100, max_depth = 7*), (*n_estimators = 75, max_depth = 5*), dan seterusnya. Setiap kombinasi tersebut digunakan untuk melatih model dan dievaluasi dengan metrik performa tertentu (Gaffar dkk., 2024).