HADLA
MEDIA INFORMASI

IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR (K-NN)

UNTUK KLASIFIKASI DATA KESEHATAN

PENULIS :

FAHMI RUZIQ, S.T., M.KOM.

M. RHIFKY WAYAHDI, S.KOM., M.KOM.

IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* (K-NN) UNTUK KLASIFIKASI DATA KESEHATAN

Penulis

Fahmi Ruziq, S.T., M.Kom.
M. Rhifky Wayahdi, S.Kom., M.Kom.

PENERBIT:

HADLA
MEDIA INFORMASI

IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* (K-NN) UNTUK KLASIFIKASI DATA KESEHATAN

Tim Penulis:
Fahmi Ruziq, S.T., M.Kom.
M. Rhifky Wayahdi, S.Kom., M.Kom.

Editor:
Nurhadi

Desain Cover:
Sulaiman

Tata Letak:
Sulaiman

ISBN:
-

Cetakan Pertama:
Maret, 2026

PENERBIT:

HADLA
MEDIA INFORMASI

Website: www.media.hadlacorp.com

PRAKATA

Bismillahirrahmanirrahim

Alhamdulillah, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah SWT, karena atas rahmat, petunjuk, dan karunia-Nya, buku yang berjudul "Implementasi Algoritma *K-Nearest Neighbor* (K-NN) untuk Klasifikasi Data Kesehatan" ini dapat diselesaikan dan hadir ke hadapan para pembaca. Shalawat serta salam tak lupa senantiasa tercurahkan kepada teladan umat, Nabi Muhammad SAW.

Buku ini lahir dari sebuah renungan dan kepedulian mendalam terhadap krisis kesehatan global di era modern. Saat ini, Penyakit Tidak Menular (PTM) seperti diabetes melitus telah menjadi ancaman serius yang sering kali datang tanpa disadari. Sering buang air kecil di malam hari, rasa haus yang tak kunjung hilang, atau penurunan berat badan tanpa sebab, kerap dianggap sekadar kelelahan biasa. Ketidaktahuan ini menyebabkan hilangnya periode emas (*golden period*) penanganan medis. Dari sinilah muncul urgensi untuk mengubah paradigma kesehatan dari kuratif (mengobati) menjadi preventif (mencegah).

Sebagai akademisi yang berkecimpung di dunia teknologi dan rekayasa perangkat lunak, penulis melihat adanya peluang besar untuk ikut andil memecahkan persoalan tersebut. Selama ini, Kecerdasan Buatan (*Artificial Intelligence* / AI) dan *Machine Learning* sering dipandang sebagai menara gading—sebuah bidang eksklusif yang hanya bisa dijalankan pada superkomputer dan bahasa pemrograman tingkat tinggi. Melalui buku ini, penulis ingin mendobrak stigma tersebut.

Buku ini disusun secara sistematis untuk memandu pembaca, mulai dari mahasiswa, peneliti, hingga pengembang web (*web developer*), dalam memahami bagaimana data medis dapat diolah menjadi sistem pendukung keputusan yang cerdas. Kami tidak hanya mengupas tuntas anatomi matematis algoritma *K-Nearest Neighbor* (K-NN) secara filosofis, tetapi juga mendemonstrasikan bagaimana algoritma tersebut dapat diterjemahkan secara nyata ke dalam ekosistem pemrograman web yang sangat memasyarakat, yakni PHP. Langkah pragmatis ini diambil dengan harapan agar teknologi deteksi dini penyakit dapat dijangkau, direplikasi, dan dikembangkan oleh siapa saja, di mana saja.

Penyelesaian buku ini tentu tidak lepas dari dukungan moral, doa, serta sumbangsih pemikiran dari berbagai pihak. Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih dan apresiasi yang setinggi-tingginya kepada keluarga tercinta atas kesabaran dan dukungan tak terhingga. Ucapan terima kasih yang tulus juga penulis sampaikan kepada rekan-rekan sejawat, para mahasiswa yang selalu menjadi inspirasi untuk terus menggali ilmu, serta segenap sivitas akademika di institusi tempat penulis bernaung dan mengabdi. Lingkungan keilmuan yang kolaboratif adalah pupuk terbaik bagi lahirnya karya-karya nyata.

Akhir kata, layaknya pepatah tiada gading yang tak retak, penulis menyadari sepenuhnya bahwa buku ini masih jauh dari kata sempurna. Oleh karena itu, segala bentuk kritik yang membangun, saran, maupun ruang diskusi yang terbuka akan sangat penulis nantikan demi penyempurnaan karya di masa mendatang.

Semoga dedikasi kecil yang tertuang dalam lembaran-lembaran buku ini dapat membawa manfaat yang luas, menjadi amal jariyah yang tak putus, serta berkontribusi nyata dalam upaya mendemokratisasi layanan *e-health* untuk meningkatkan kualitas hidup masyarakat.

Selamat membaca dan berinovasi!

Medan, Februari 2026

Tim Penulis.

DAFTAR ISI

BAB I TRANSFORMASI DIGITAL DAN KECERDASAN BUATAN DALAM PELAYANAN KESEHATAN MODERN

1.1. Krisis Kesehatan Global dan Paradigma Preventif

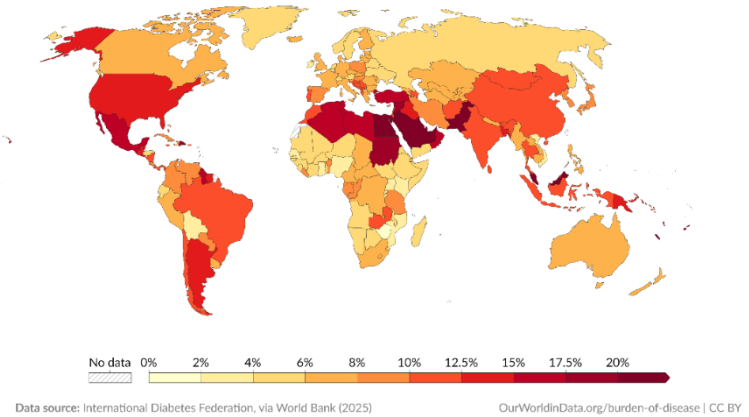
Kesehatan merupakan hak asasi manusia yang fundamental, namun tantangan dalam penyediaan layanan kesehatan yang merata dan berkualitas semakin kompleks seiring berjalannya waktu. Di abad ke-21 ini, dunia medis menghadapi transisi epidemiologi yang signifikan. Jika pada abad sebelumnya fokus utama dunia kesehatan adalah penyakit menular (infeksi), kini beban penyakit telah bergeser secara dramatis ke arah Penyakit Tidak Menular (PTM) atau *Non-Communicable Diseases* (NCDs) (Collins et al., 2021; Shu & Jin, 2023). Penyakit seperti diabetes melitus, hipertensi, gangguan kardiovaskular, dan kanker kini menjadi penyumbang terbesar angka kematian global (Arokiasamy et al., 2021; Vaduganathan et al., 2022).

Data dari berbagai organisasi kesehatan dunia menunjukkan tren yang mengkhawatirkan. Sebagai contoh kasus yang menonjol, Diabetes Melitus (DM) telah berkembang menjadi epidemi global. Estimasi menunjukkan bahwa ratusan juta orang dewasa di seluruh dunia hidup dengan kondisi ini, dengan angka prevalensi yang terus meningkat setiap tahunnya. Beban ini tidak hanya dirasakan secara fisik oleh penderita melalui komplikasi yang menyertainya—seperti gagal ginjal, kerusakan saraf, dan penyakit jantung—tetapi juga memberikan tekanan ekonomi yang luar biasa bagi sistem kesehatan nasional di berbagai negara (Almalki & Khan, 2025).

Masalah fundamental dalam penanganan penyakit kronis seperti diabetes adalah fenomena "Fenomena Gunung Es". Banyak kasus yang tidak terdiagnosis hingga mencapai stadium lanjut yang parah. Gejala-gejala awal yang sebenarnya muncul—seperti polidipsia (banyak minum), poliuria (banyak buang air kecil), dan kelelahan kronis—sering kali diabaikan oleh masyarakat karena dianggap sebagai keluhan ringan biasa. Ketidaktahuan dan kurangnya akses terhadap *screening* dini menyebabkan hilangnya "periode emas" penanganan, di mana intervensi gaya hidup sebenarnya masih bisa membalikkan atau mengontrol kondisi tersebut (Goldmann et al., 2023).

Oleh karena itu, paradigma pelayanan kesehatan modern harus bergeser dari Kuratif (mengobati orang sakit) menjadi Preventif (mencegah sebelum sakit parah). Dalam paradigma preventif ini, deteksi dini (*early detection*) adalah kunci utama. Namun, tantangannya adalah: Bagaimana kita bisa melakukan deteksi dini

secara massal, murah, dan akurat di tengah keterbatasan jumlah tenaga medis? Jawabannya terletak pada konvergensi antara ilmu kedokteran dan teknologi informasi.



Gambar 1.1. Persentase penderita diabetes tahun 2024 (Sumber: https://ourworldindata.org/grapher/diabetes-prevalence)

1.1.1. Disrupsi Teknologi: Kesehatan di Era Revolusi Industri 4.0 dan Society 5.0

Dunia saat ini sedang berada di tengah gelombang Revolusi Industri 4.0, sebuah era di mana batas antara dunia fisik, digital, dan biologis semakin kabur. Di sektor kesehatan, fenomena ini melahirkan istilah Health 4.0, yang dicirikan oleh interkonektivitas, otomasi, dan analisis data cerdas. Jika pada era sebelumnya teknologi kesehatan berfokus pada alat mekanik (seperti mesin rontgen analog), Health 4.0 berfokus pada *Cyber-Physical Systems* (CPS).

Dalam ekosistem ini, tubuh manusia tidak lagi dipandang sebagai entitas biologis yang terisolasi, melainkan sebagai sumber data yang terus-menerus memancarkan informasi. Melalui perkembangan *Internet of Things* (IoT) dan *Internet of Medical Things* (IoMT), parameter vital seperti detak jantung, kadar glukosa, dan pola tidur dapat dipantau secara real-time. Tantangannya bukan lagi bagaimana cara mengambil data, melainkan bagaimana mengubah banjir data ini menjadi wawasan medis.

Sejalan dengan itu, konsep Society 5.0 yang digagas oleh pemerintah Jepang menawarkan perspektif yang lebih humanis. Jika Industri 4.0 berfokus pada produksi, Society 5.0 berfokus pada manusia. Dalam konteks kesehatan, ini berarti teknologi—termasuk Big Data dan Artificial Intelligence (AI)—harus digunakan untuk menciptakan masyarakat yang sehat dan panjang umur, di mana lansia atau

penderita penyakit kronis tetap dapat hidup mandiri dengan bantuan sistem pemantauan cerdas.

Sistem prediksi risiko penyakit berbasis web yang dibahas dalam buku ini adalah manifestasi nyata dari semangat Society 5.0. Ia menggunakan kecanggihan algoritma (AI) namun dikemas dalam antarmuka web yang sederhana agar dapat diakses oleh siapa saja (*human-centric*), menghapus hambatan teknologi bagi masyarakat awam untuk mendapatkan layanan kesehatan preventif (Gordon & Mary Oluwaseun, 2025).

1.2. Evolusi Informatika Kesehatan (*Health Informatics*)

Informatika Kesehatan bukan lagi sekadar tentang mendigitalkan catatan pasien dari kertas ke komputer. Disiplin ilmu ini telah berevolusi menjadi jembatan vital yang menghubungkan data klinis dengan keputusan medis. Dalam satu dekade terakhir, kita menyaksikan ledakan data kesehatan (*big data healthcare*). Data ini bersumber dari rekam medis elektronik, hasil laboratorium, perangkat *wearable*, hingga data demografi pasien.

Namun, data hanyalah sekumpulan angka dan fakta mentah yang tidak memiliki makna jika tidak diolah. Di sinilah peran teknologi komputasi menjadi krusial. Sistem informasi kesehatan modern dituntut untuk memiliki kemampuan memproses data mentah tersebut menjadi informasi, dan lebih jauh lagi, menjadi pengetahuan (*knowledge*) yang dapat ditindaklanjuti (Yogesh & Karthikeyan, 2022).

1.2.1. Dari Data Menjadi Keputusan

Dalam konteks deteksi risiko penyakit, proses transformasi data mengikuti hierarki DIKW (*Data*, *Information*, *Knowledge*, *Wisdom*).

Gambar 1.2. Piramida DIKW (*Data*, *Information*, *Knowledge*, *Wisdom*)

a. Data: Fakta mentah, seperti usia pasien, jenis kelamin, atau jawaban Ya/Tidak terhadap gejala sering haus.
b. Informasi: Pengorganisasian data tersebut menjadi profil pasien yang terstruktur.
c. Pengetahuan: Penerapan algoritma untuk mengenali pola, misalnya: Pasien dengan usia >40 tahun yang mengalami poliuria dan penurunan berat badan drastis memiliki kemiripan pola dengan penderita diabetes.
d. Kebijaksanaan (Keputusan): Rekomendasi sistem kepada pengguna untuk segera memeriksakan diri ke dokter (Bratianu & Bejinaru, 2023).

Teknologi informasi memungkinkan proses ini berjalan otomatis. Sistem berbasis komputer dapat melakukan triase awal, memilah mana individu yang berisiko tinggi dan mana yang rendah, sehingga tenaga medis dapat memprioritaskan penanganan pada mereka yang paling membutuhkan.

1.3. Sistem Pendukung Keputusan Klinis (*Clinical Decision Support System*)

Salah satu implementasi nyata dari informatika kesehatan adalah *Clinical Decision Support System* (CDSS). CDSS dirancang untuk membantu tenaga profesional kesehatan maupun orang awam dalam membuat keputusan klinis. Penting untuk digarisbawahi bahwa sistem ini tidak bertujuan menggantikan peran dokter, melainkan bertindak sebagai opini kedua atau alat penapis (*screening tool*) yang objektif.

Keunggulan utama penggunaan sistem terkomputerisasi dalam diagnosis awal meliputi:

a. Konsistensi: Algoritma komputer akan memberikan hasil yang sama untuk input yang sama, berbeda dengan penilaian manusia yang bisa dipengaruhi oleh kelelahan atau bias kognitif.
b. Kecepatan: Komputer dapat memproses puluhan variabel gejala (atribut) dalam hitungan milidetik.
c. Aksesibilitas: Sistem dapat diduplikasi dan diakses dari mana saja, mengatasi kesenjangan geografis ketersediaan dokter spesialis.

Dalam praktiknya, CDSS membutuhkan otak untuk berpikir. Otak inilah yang disediakan oleh cabang ilmu komputer yang dikenal sebagai Kecerdasan Buatan atau *Artificial Intelligence* (AI), khususnya bidang Pembelajaran Mesin (*Machine Learning*) (Vasey et al., 2021).

1.3.1. Pergeseran Paradigma: Dari *Rule-Based System* ke *Data-Driven Approach*

Sebelum popularitas *Machine Learning* meroket, sistem pendukung keputusan klinis umumnya dibangun menggunakan pendekatan berbasis aturan atau *Rule-Based System* (sering disebut Sistem Pakar Konvensional). Dalam pendekatan ini, seorang programmer harus bekerja sama dengan dokter untuk memetakan logika Sebab-Akibat secara eksplisit ke dalam kode program. Contoh logika *Rule-Based*:

"JIKA Gula Darah > 200 mg/dL DAN Sering Haus = Ya, MAKA Diagnosis = Diabetes."

Meskipun pendekatan ini logis, ia memiliki kelemahan fatal: Kekakuan (*Rigidity*). Tubuh manusia sangat kompleks dan tidak selalu mengikuti aturan biner yang kaku. Bagaimana jika gula darah pasien 199 mg/dL? Bagaimana jika pasien tidak merasa haus tapi berat badannya turun drastis? Sistem berbasis aturan akan kesulitan menangani variasi dan ketidakpastian ini. Selain itu, seiring bertambahnya variabel gejala, jumlah aturan JIKA-MAKA akan meledak menjadi ribuan kombinasi yang mustahil dikelola secara manual oleh manusia.

Di sinilah pendekatan *Data-Driven* (Berbasis Data) melalui *Machine Learning* menawarkan terobosan. Alih-alih mendiktekan aturan kepada komputer, kita memberikan komputer sekumpulan data riwayat pasien (Data Latih) dan membiarkan algoritma menemukan aturannya sendiri.

Algoritma tidak menghafal aturan medis, melainkan mempelajari pola statistik. Ia mungkin menemukan korelasi tersembunyi yang luput dari pengamatan manusia, misalnya kombinasi spesifik antara usia, kerontokan rambut (alopecia), dan kekakuan otot yang ternyata memiliki probabilitas tinggi mengarah pada risiko diabetes. Pendekatan ini membuat sistem menjadi adaptif; semakin banyak data baru yang masuk, semakin cerdas sistem tersebut dalam mengenali pola penyakit. Inilah yang mendasari urgensi penerapan metode klasifikasi cerdas dibandingkan sekadar membuat kuesioner kesehatan digital biasa (Saturi, 2022).

1.4. Urgensi *Machine Learning* dalam Klasifikasi Medis

Metode statistik konvensional seringkali memiliki keterbatasan ketika berhadapan dengan data medis yang kompleks, non-linear, dan memiliki banyak variabel. Di sinilah *Machine Learning* (ML) menawarkan solusi yang elegan. ML memungkinkan komputer untuk belajar dari data historis (data pengalaman masa lalu) untuk membuat prediksi terhadap data baru.

Gambar 1.3. Ilustrasi Integrasi *Artificial Intelligence* (AI) dan Layanan Kesehatan

Dalam konteks klasifikasi penyakit, algoritma ML bekerja dengan cara memetakan pola input (gejala) ke output (diagnosis). Terdapat berbagai algoritma yang populer digunakan dalam ranah kesehatan, mulai dari yang kompleks seperti *Neural Networks* dan *Support Vector Machines* (SVM), hingga yang sederhana namun interpretabilitasnya tinggi seperti *K-Nearest Neighbor* (K-NN).

Mengapa algoritma klasifikasi seperti K-NN relevan untuk data kesehatan?

a. Kemampuan Menangani Data Non-Linear: Hubungan antara gejala penyakit seringkali tidak lurus (linear). K-NN, sebagai algoritma non-parametrik, sangat fleksibel dalam memodelkan batas keputusan yang tidak beraturan.
b. Kesederhanaan dan Transparansi: Dalam dunia medis, kotak hitam (*black box*) sering dihindari. Dokter dan pasien perlu memahami mengapa sebuah prediksi dibuat. K-NN menawarkan logika yang sangat intuitif: Pasien ini diprediksi sakit X karena karakteristik gejalanya mirip dengan tetangga terdekatnya (pasien lain) yang sudah terbukti sakit X.
c. Efektivitas pada Dataset Kecil hingga Menengah: Tidak semua kasus medis memiliki big data jutaan baris. Untuk dataset klinis awal dengan ratusan atau ribuan data, algoritma seperti K-NN seringkali memberikan performa yang kompetitif dibandingkan algoritma yang lebih berat (Allam et al., 2025; Badnjević & Spahić, 2026).

Tabel 1.1. Perbandingan Karakteristik Algoritma Klasifikasi Populer di Bidang Medis

Algoritma	**Tingkat Akurasi**	**Kompleksitas Komputasi**	**Interpretabilitas (Kemudahan Dipahami)**	**Kelebihan Utama**	**Kelemahan Utama**
K-Nearest Neighbor (K-NN)	Kompetitif (Baik)	Rendah (Ringan)	Sangat Tinggi (Intuitif)	Sangat sederhana, mudah diimplementasikan pada Web/PHP, tidak ada fase training berat.	Sensitif terhadap outlier dan data yang tidak seimbang (*imbalanced*)
Support Vector Machine (SVM)	Sangat Tinggi	Tinggi (Berat)	Rendah (Black Box)	Sangat akurat untuk dimensi tinggi.	Sulit dipahami logikanya oleh orang awam/dokter; komputasi berat.
Random Forest	Sangat Tignggi	Sangat Tinggi	Sedang	Menangani data besar dengan sangat baik.	Model yang dihasilkan sangat kompleks (banyak pohon keputusan), lambat saat prediksi *real-time*.
Decision Tree	Baik	Rendah	Tinggi	Mudah divisualisasikan	Cenderung *overfitting* (terlalu menghafal data latih).

Berdasarkan Tabel 1.1, meskipun algoritma kompleks seperti *Random Forest* atau SVM mungkin menawarkan akurasi yang sedikit lebih tinggi dalam beberapa kasus literatur, *K-Nearest Neighbor* (K-NN) menawarkan keseimbangan terbaik antara performa dan transparansi.

Dalam pengembangan sistem kesehatan berbasis web yang ditujukan untuk *screening* awal, kecepatan respon sistem dan kemudahan audit logika (mengapa pasien didiagnosis berisiko) adalah prioritas utama. K-NN bekerja dengan logika kemiripan tetangga yang sangat alami bagi pola pikir manusia, menjadikannya metode yang ideal untuk diadopsi dalam arsitektur perangkat lunak yang ringan seperti PHP, tanpa membebani server dengan proses komputasi yang berlebihan.

1.5. Peran Teknologi Web dalam Demokratisasi Layanan Kesehatan

Memiliki algoritma prediksi yang canggih tidak akan berguna jika tidak dapat diakses oleh masyarakat luas. Oleh karena itu, platform penyampaian (*delivery platform*) menjadi elemen kunci dalam arsitektur sistem kesehatan digital. Website adalah media komunikasi yang paling universal dan efektif saat ini.

1.5.1. Ekosistem Telemedicine dan *Remote Patient Monitoring*

Integrasi sistem prediksi ke dalam platform web menempatkannya sebagai bagian integral dari ekosistem Telemedicine. Pasca-pandemi global, definisi interaksi dokter-pasien telah berubah permanen. Layanan kesehatan tidak lagi dibatasi oleh tembok rumah sakit.

Teknologi web memungkinkan terciptanya *Continuum of Care* (Layanan Berkelanjutan). Dalam model konvensional, data kesehatan pasien terfragmentasi—tersimpan di kertas yang hilang atau di ingatan pasien. Dengan sistem berbasis web:

1. Rekam Jejak Digital: Setiap kali pengguna melakukan input gejala, data tersebut tersimpan dalam database (MySQL). Seiring waktu, ini membentuk grafik riwayat kesehatan (*Health History*) yang sangat berharga. Dokter dapat melihat tren: Apakah gejala poliuria pasien ini memburuk dalam 3 bulan terakhir?.
2. Interoperabilitas: Sistem web yang dibangun dengan standar terbuka (seperti PHP/API) memiliki potensi untuk berbicara dengan sistem lain. Di masa depan, hasil prediksi dari sistem ini bisa langsung dikirim ke sistem pendaftaran Puskesmas atau aplikasi asuransi kesehatan, menciptakan alur penanganan yang mulus (seamless).

Oleh karena itu, pemilihan platform web bukan semata-mata karena kemudahan coding, melainkan strategi jangka panjang untuk menjamin keberlanjutan data dan integrasi layanan kesehatan nasional.

Mengintegrasikan sistem prediksi penyakit ke dalam lingkungan berbasis Web (*Web-Based System*) memiliki implikasi strategis:

a. Kemandirian Pasien (*Self-Assessment*): Masyarakat dapat melakukan *screening* mandiri di rumah tanpa harus antre di rumah sakit hanya untuk konsultasi awal. Ini mendukung konsep *patient-centered care*.
b. Skalabilitas: Aplikasi web yang dibangun dengan standar teknologi yang matang (seperti PHP dan MySQL) dapat diakses oleh ribuan pengguna secara bersamaan tanpa memerlukan instalasi perangkat lunak khusus di sisi pengguna (*client-side*).

c. Efisiensi Sumber Daya: Di daerah dengan sumber daya terbatas (3T - Tertinggal, Terdepan, Terluar), sistem berbasis web dapat menjadi alat bantu triase yang vital bagi tenaga kesehatan lokal untuk memprioritaskan rujukan pasien.

Integrasi *Machine Learning* ke dalam teknologi web, misalnya menggunakan bahasa pemrograman PHP yang sangat populer di kalangan pengembang web, membuktikan bahwa kecerdasan buatan tidak harus eksklusif berjalan di lingkungan laboratorium superkomputer, tetapi bisa berjalan di server web standar yang melayani kebutuhan praktis masyarakat.

1.6. Tantangan Implementasi: *Data Preprocessing* dan Kualitas Data

Sebuah sistem cerdas hanya akan sebaik data yang dilatihnya (*Garbage In*, *Garbage Out*). Dalam implementasi sistem klasifikasi kesehatan, tantangan terbesar seringkali bukan pada pemilihan model algoritmanya, melainkan pada penyiapan datanya.

Data kesehatan dunia nyata seringkali kotor. Data tersebut mungkin memiliki format yang tidak seragam (misalnya: Usia dalam angka, tapi Jenis Kelamin dalam teks Pria/Wanita). Oleh karena itu, tahapan *Data Preprocessing* menjadi fondasi yang tidak boleh dilewatkan. Tahapan ini mencakup:

a. Konversi Data Kategorikal: Mengubah data teks menjadi format numerik agar bisa dihitung jarak matematisnya oleh komputer.
b. Normalisasi (*Scaling*): Menyamakan rentang nilai antar atribut. Tanpa normalisasi, atribut dengan nilai satuan besar (misalnya: Gula Darah = 200 mg/dL) akan mendominasi perhitungan dibandingkan atribut bernilai kecil (misalnya: Usia = 40 tahun), yang menyebabkan bias pada algoritma berbasis jarak seperti K-NN.
c. Penanganan *Missing Values*: Memastikan integritas dataset agar prediksi tidak meleset (Deshkar et al., 2024).

Pemahaman mendalam tentang karakteristik data ini sangat krusial sebelum kita melangkah ke tahap pemrograman atau implementasi algoritma.

1.6.1. Tantangan Faktor Manusia (*Human Factors*) dan Literasi Digital

Selain tantangan teknis dalam penyiapan data, implementasi sistem kesehatan digital juga menghadapi tantangan sosiologis, yaitu kesiapan penggunanya (*User Readiness*).

Sebuah sistem prediksi risiko diabetes yang dibangun dengan algoritma K-NN paling akurat sekalipun akan gagal jika antarmuka penggunanya (*User Interface*)

membingungkan atau jika masyarakat tidak percaya pada hasilnya. Studi menunjukkan bahwa hambatan utama adopsi *e-health* di negara berkembang meliputi:

1. Literasi Kesehatan Digital (*e-Health Literacy*): Kemampuan pengguna untuk mencari, memahami, dan mengevaluasi informasi kesehatan dari media elektronik. Pengembang sistem harus memastikan bahasa yang digunakan dalam aplikasi mudah dipahami oleh orang awam, menghindari jargon medis yang terlalu teknis tanpa penjelasan.
2. Kepercayaan terhadap AI (*Trust in AI*): Fenomena Algorithmic Aversion sering terjadi, di mana manusia lebih percaya pada saran manusia lain yang salah daripada saran algoritma yang benar. Oleh karena itu, sistem harus transparan (seperti keunggulan K-NN yang explainable) dan selalu menyertakan disclaimer bahwa hasil prediksi adalah alat bantu deteksi dini, bukan diagnosis final medis.
3. Kesenjangan Digital (*Digital Divide*): Meskipun penetrasi internet tinggi, akses terhadap perangkat yang memadai dan koneksi stabil di daerah pedesaan masih menjadi kendala. Membangun sistem web yang ringan (*lightweight*)—yang tidak memakan banyak kuota data dan cepat dimuat di jaringan lambat—adalah sebuah keharusan teknis yang berdampak sosial.

1.7. Etika dan Privasi dalam Pengolahan Data Medis

Dalam pengembangan sistem cerdas di bidang kesehatan, kemampuan teknis untuk memprediksi penyakit harus selalu diimbangi dengan tanggung jawab etis yang ketat. Data medis bukanlah sekadar deretan angka statistik; di balik setiap baris data terdapat privasi, martabat, dan kehidupan seorang pasien. Oleh karena itu, prinsip Primum non nocere (Pertama, jangan menyakiti) yang dipegang teguh oleh dunia kedokteran juga harus diadopsi oleh para pengembang perangkat lunak kesehatan (*Health Informatics Developers*).

Tantangan utama dalam digitalisasi kesehatan adalah menjaga Kerahasiaan (*Confidentiality*) dan Integritas (*Integrity*). Data kesehatan seperti riwayat diabetes, status HIV, atau kondisi mental adalah informasi sensitif yang jika bocor dapat menimbulkan stigma sosial atau diskriminasi bagi pemiliknya (Ali, 2025; Kirpalani & Kumar, 2024).

Dalam konteks penelitian dan pengembangan sistem prediksi, terdapat dua aspek etis utama yang harus diperhatikan:

1. Anonimisasi Data: Sebelum data digunakan untuk melatih algoritma *Machine Learning*, atribut yang dapat mengidentifikasi individu secara langsung (*Personally Identifiable Information* - PII) seperti Nama, NIK, atau Alamat Rumah harus dihapus atau disamarkan. Dalam buku ini, kita

menggunakan dataset publik (seperti *Early Stage Diabetes Risk Prediction* dari *Kaggle*) yang telah melalui proses anonimisasi. Dataset ini hanya menyisakan atribut klinis (gejala) dan demografi umum (usia, jenis kelamin) tanpa menyertakan identitas personal, sehingga aman untuk digunakan sebagai bahan pembelajaran dan riset terbuka.

2. Bias Algoritma dan Keadilan (*Fairness*): Sebuah sistem AI bisa menjadi "bias" jika data yang digunakan untuk melatihnya tidak representatif. Misalnya, jika sebuah sistem deteksi diabetes hanya dilatih menggunakan data pasien lanjut usia, sistem tersebut mungkin gagal mendeteksi gejala diabetes pada remaja atau dewasa muda. Penting bagi pengembang untuk memastikan dataset memiliki distribusi yang seimbang dan mencakup variasi populasi yang beragam agar prediksi yang dihasilkan adil dan akurat untuk semua kalangan.
3. Transparansi Keputusan (*Explainability*): Pasien berhak tahu mengapa sebuah sistem mendiagnosis mereka berisiko diabetes. Di sinilah keunggulan metode seperti *K-Nearest Neighbor* (K-NN) yang kita bahas. Berbeda dengan model Kotak Hitam (*Black Box*) yang sulit dijelaskan, K-NN menawarkan transparansi tinggi karena keputusannya didasarkan pada kemiripan gejala dengan kasus terdahulu yang nyata. Ini memenuhi standar etika medis di mana setiap diagnosis harus dapat dipertanggungjawabkan secara logis (Verma et al., 2025; Younas et al., 2026).

1.8. Evolusi PHP dalam Ranah *Data Science*: Sebuah Perspektif Alternatif

Dunia *Data Science* dan *Machine Learning* (ML) saat ini didominasi oleh bahasa pemrograman Python. Namun, dalam ekosistem pengembangan web, PHP memegang pangsa pasar yang sangat dominan, mentenagai sebagian besar situs web di dunia. Hal ini memunculkan pertanyaan mendasar: Apakah kita harus selalu beralih ke Python untuk menyisipkan kecerdasan buatan ke dalam website?

Secara historis, PHP didesain sebagai bahasa skrip untuk pengembangan web (*server-side scripting*), bukan untuk komputasi matematika berat. Namun, perkembangan PHP modern (versi 7 dan 8) telah membawa peningkatan performa yang signifikan dalam kecepatan eksekusi dan manajemen memori. Hal ini membuka peluang baru bagi implementasi algoritma ML sederhana secara *native*.

Terdapat pergeseran paradigma di mana komunitas pengembang mulai menyadari kebutuhan akan ML yang terintegrasi langsung (*Embedded* ML). Seringkali, sebuah aplikasi web hanya membutuhkan fitur prediksi sederhana seperti prediksi risiko diabetes berdasarkan gejala yang tidak memerlukan infrastruktur Python yang kompleks dan berat. Dalam skenario seperti ini,

menggunakan pustaka PHP murni (Native PHP Library) menjadi solusi yang lebih efisien dan ringan.

Beberapa tonggak perkembangan PHP dalam ranah ini meliputi:

1. Munculnya Pustaka Khusus: Kehadiran pustaka seperti PHP-ML (*PHP Machine Learning*) telah mendemokratisasi akses AI bagi pengembang web. Pustaka ini menyediakan fungsi siap pakai untuk klasifikasi (termasuk K-NN), regresi, dan klastering, memungkinkan pengembang fokus pada logika bisnis aplikasi tanpa harus membangun algoritma matematika dari nol.
2. Kemudahan Integrasi: Menggunakan PHP untuk ML menghilangkan hambatan integrasi. Pengembang tidak perlu mengelola dua lingkungan server berbeda (PHP untuk Web dan Python untuk AI). Data input dari formulir web dapat langsung diproses oleh algoritma K-NN di server yang sama, mempercepat waktu respons dan menyederhanakan arsitektur sistem.
3. Relevansi untuk Dataset Skala Menengah: Meskipun PHP mungkin belum secepat C++ atau Python (dengan NumPy) untuk mengolah Big Data jutaan baris, penelitian menunjukkan bahwa untuk dataset skala kecil hingga menengah (seperti ratusan atau ribuan data rekam medis), performa PHP sangat memadai dan dapat diandalkan.

Dengan demikian, buku ini mengambil pendekatan yang pragmatis: memanfaatkan teknologi web yang paling banyak digunakan (PHP) untuk memecahkan masalah kesehatan nyata, membuktikan bahwa solusi cerdas tidak selalu harus rumit atau mahal.

BAB 2 KARAKTERISTIK DAN MANAJEMEN DATA DALAM INFORMATIKA KESEHATAN

2.1. Kompleksitas Data Medis di Era Digital

Data kesehatan (*health data*) memiliki karakteristik yang unik dibandingkan dengan data dari sektor lain seperti keuangan atau *e-commerce*. Jika data transaksi keuangan bersifat pasti (deterministik), data kesehatan seringkali mengandung ketidakpastian (*uncertainty*), subjektivitas, dan variabilitas yang tinggi. Dalam konteks pengembangan sistem cerdas berbasis *Machine Learning*, pemahaman terhadap natur data ini adalah langkah pertama yang menentukan keberhasilan model prediksi (Eloranta & Boman, 2022).

Secara umum, data kesehatan yang digunakan dalam sistem prediksi penyakit bersumber dari dua entitas utama:

1. Data Klinis Objektif: Data yang diukur menggunakan alat medis dan memiliki satuan standar. Contohnya adalah kadar gula darah, tekanan darah, atau indeks massa tubuh (BMI).
2. Data Klinis Subjektif (Simtomatik): Data yang berasal dari keluhan pasien (anamnesis). Contohnya adalah rasa sering haus (polydipsia), sering buang air kecil (polyuria), atau penglihatan kabur (*visual blurring*).

Tantangan terbesar bagi seorang *data scientist* atau pengembang web kesehatan adalah menerjemahkan data subjektif—yang seringkali bersifat kualitatif seperti kalimat "Dok, saya merasa lemas"—menjadi data kuantitatif yang dapat diproses oleh algoritma komputasi seperti *K-Nearest Neighbor* (K-NN) (Tsaneva-Atanasova et al., 2025).

2.2. Taksonomi dan Tipe Atribut Data

Dalam terminologi Data Mining, variabel-variabel yang membentuk sebuah dataset disebut sebagai atribut atau fitur. Memahami tipe atribut sangat vital karena akan menentukan metode pra-pemrosesan (*preprocessing*) apa yang valid untuk diterapkan. Dalam dataset risiko penyakit, seperti pada kasus diabetes, kita umumnya menemukan campuran dari tipe-tipe data berikut:

2.2.1. Data Numerik (Rasio dan Interval)

Data numerik adalah data yang dapat diukur dan diurutkan. Dalam algoritma berbasis jarak seperti K-NN, data ini adalah yang paling bersahabat karena jarak antar nilainya nyata (Uddin et al., 2022).

Contoh: Usia (*Age*). Memiliki karakteristik yaitu rentang nilai yang lebar (misalnya 11 hingga 90 tahun). Implikasi pada K-NN yaitu: karena rentang nilainya bisa sangat besar (puluhan hingga ratusan), atribut numerik seringkali mendominasi perhitungan jarak Euclidean jika dibandingkan dengan atribut biner (0 atau 1). Oleh karena itu, atribut seperti usia mutlak memerlukan normalisasi agar kontribusinya setara dengan atribut lain.

2.2.2. Data Kategorikal (Nominal)

Data ini berfungsi sebagai label atau pengelompokan tanpa urutan tingkatan.

Contohnya Jenis Kelamin (*Gender*), Polifagia (Rasa lapar berlebih), Genital Thrush, atau Alopecia (Kerontokan rambut). Karakteristik nlainya berupa teks atau status, seperti Pria/Wanita atau Ya/Tidak (Yes/No). Implikasi pada K-NN yaitu komputer tidak bisa menghitung jarak antara kata Pria dan Wanita. Data ini harus dikonversi menjadi format numerik (0 dan 1) melalui proses *Label Encoding* atau *One-Hot Encoding* sebelum masuk ke tahap pemodelan.

2.2.3. Data Ordinal

Data kategori yang memiliki tingkatan atau urutan. Meskipun dalam dataset biner (Yes/No) urutan tidak terlihat, dalam konteks medis yang lebih luas, kita sering menemui data ordinal seperti stadium kanker (I, II, III, IV) atau skala nyeri (1-10).

Untuk memberikan gambaran yang lebih jelas mengenai bagaimana data gejala didefinisikan dalam sistem, berikut adalah ringkasan atribut yang digunakan dalam studi kasus prediksi diabetes ini:

Tabel 2.1. Atribut Dataset Risiko Diabetes

No.	Nama Atribut	Tipe Data	Deskripsi Medis / Nilai
1.	Age	Numerik	Usia pasien saat pemeriksaan, berkisar antara 20 hingga 65 tahun
2.	Sex	Kategorikal	Jenis kelamin biologis pasien (Laki-laki atau Perempuan).
3.	Polyuria	Kategorikal	Kondisi di mana pasien mengalami pengeluaran urin yang berlebihan atau abnormal.

4.	Polydipsia	Kategorikal	Rasa haus yang ekstrem dan terus-menerus, sering kali merupakan tanda awal hiperglikemia.
5.	Sudden Weight Loss	Kategorikal	Penurunan berat badan secara drastis dalam waktu singkat tanpa upaya yang disengaja.
6.	Weakness	Kategorikal	Perasaan lelah, letih, atau kekurangan energi yang sering dialami pasien diabetes.
7.	Polyphagia	Kategorikal	Peningkatan rasa lapar yang berlebihan akibat sel tubuh tidak mendapatkan asupan energi yang cukup.
8.	Genital Thrush	Kategorikal	Infeksi jamur pada area genital yang lebih rentan terjadi pada penderita kadar gula darah tinggi.
9.	Visual Blurring	Kategorikal	Gangguan penglihatan kabur yang disebabkan oleh perubahan kadar cairan dalam lensa mata.
10.	Itching	Kategorikal	Rasa gatal pada kulit yang bisa disebabkan oleh sirkulasi darah yang buruk atau infeksi kulit.
11.	Irritability	Kategorikal	Perubahan suasana hati atau perasaan mudah marah yang sering dikaitkan dengan fluktuasi gula darah.
12.	Delayed Healing	Kategorikal	Lambatnya proses penyembuhan luka atau infeksi pada tubuh pasien.
13.	Partial Paresis	Kategorikal	Kelemahan otot sebagian yang dapat mengindikasikan adanya komplikasi pada saraf (neuropati).
14.	Muscle Stiffness	Kategorikal	Kondisi otot yang terasa kaku atau tegang, sering dikaitkan dengan gangguan metabolik.
15.	Alopecia	Kategorikal	Kerontokan rambut yang tidak normal atau kebotakan pada area tertentu.
16.	Obesity	Kategorikal	Kondisi berat badan berlebih yang menjadi faktor risiko utama terjadinya diabetes tipe 2.

17.	Class	Label (Target)	Hasil klasifikasi akhir: Positive (Berisiko Diabetes) atau Negative (Tidak Berisiko).

Dalam konteks algoritma K-NN, atribut kategorikal di atas sering kali diperlakukan sebagai data biner (0 dan 1) dalam tahap pra-pemrosesan. Meskipun secara teknis bersifat nominal, pemberian nilai 0 untuk "No" (tidak ada gejala) dan 1 untuk "Yes" (ada gejala) memberikan tingkatan logis yang membantu algoritma menghitung jarak Euclidean.

Semakin banyak gejala yang bernilai 1 (Yes), maka posisi data tersebut dalam ruang multidimensi akan cenderung semakin dekat dengan klaster pasien yang berlabel Positive. Hal inilah yang mendasari mengapa pemahaman terhadap setiap atribut dalam Tabel 2.1 sangat krusial bagi akurasi sistem prediksi risiko diabetes yang dibangun.

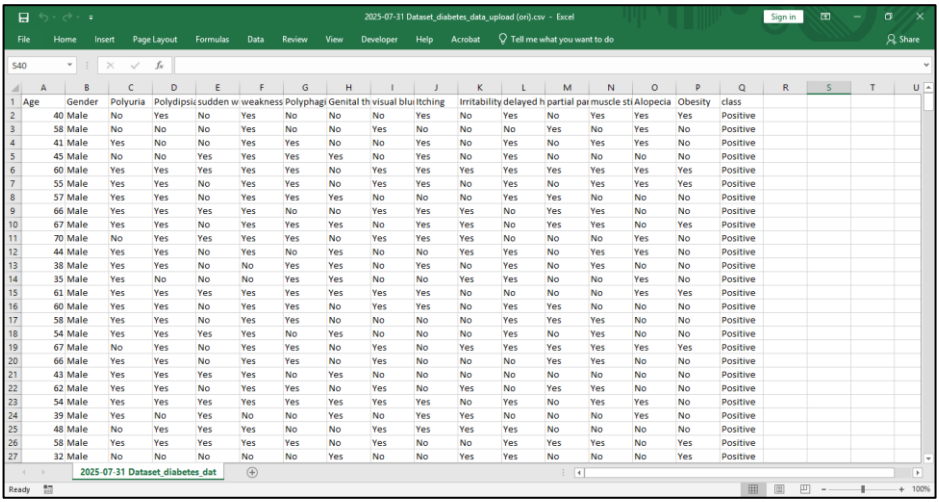
2.3. Prosedur Audit dan Validasi Teknis Data

Sebelum melangkah ke tahap pemodelan algoritma, seorang pengembang sistem cerdas harus melakukan audit teknis terhadap dataset yang akan digunakan. Prosedur ini bertujuan untuk memastikan bahwa data benar-benar bersih, lengkap, dan memiliki integritas struktural yang baik. Dalam pengembangan sistem prediksi risiko diabetes ini, audit dilakukan secara programatik untuk meminimalkan risiko kesalahan manusia (*human error*).

Berikut adalah tahapan audit teknis yang dilakukan sebelum data masuk ke tahap pra-pemrosesan:

2.3.1. Audit Dimensi dan Struktur Data (Data Shape)

Langkah pertama dalam audit teknis adalah melakukan pemeriksaan terhadap dimensi dataset. Hal ini dilakukan untuk memastikan bahwa jumlah sampel (baris) dan jumlah atribut (kolom) sesuai dengan spesifikasi awal penelitian. Prosedur: Menggunakan metode **data.shape** untuk mengekstraksi informasi dimensi. Hasil Audit: Berdasarkan pemeriksaan, dataset memiliki tepat 520 baris data pasien dan 17 kolom atribut. Konsistensi dimensi ini sangat penting agar tidak terjadi kehilangan informasi saat data dipecah menjadi data latih (*training*) dan data uji (*testing*).



Age	Gender	Polyuria	Polydipsia	sudden w	weakness	Polyphagi	Genital th	visual blu	itching	Irritability	delayed h	partial par	muscle sti	Alopecia	Obesity	class
40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive
55	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Positive
57	Male	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes	No	No	No	Positive
66	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No	No	Positive
67	Male	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Positive
70	Male	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	Positive
44	Male	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes	No	Yes	Yes	No	Positive
38	Male	Yes	Yes	No	No	Yes	Yes	No	Yes	No	Yes	No	Yes	No	No	Positive
35	Male	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No	Positive
61	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Positive
60	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	No	No	Positive
58	Male	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes	No	No	Positive
54	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	No	Yes	No	No	Positive
67	Male	No	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Positive
66	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes	Yes	No	No	Positive
43	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No	No	No	No	Positive
62	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	Positive
54	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	Positive
39	Male	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Positive
48	Male	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No	No	Positive
58	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	No	Yes	Positive
32	Male	No	No	No	No	No	Yes	No	No	Yes	Yes	No	No	No	Yes	Positive

Gambar 2.1. *Screenshot* Dataset Mentah Risiko Diabetes dalam Format Tabular.

2.3.2. Verifikasi Kelengkapan Data (*Null-Value Detection*)

Masalah utama dalam data medis dunia nyata adalah ketidakhadiran nilai (*missing values*) yang dapat merusak perhitungan jarak pada algoritma K-NN. Audit dilakukan secara mendalam pada setiap sel di dalam tabel dataset.

Analisis Tipe Data: Melalui metode **data.info()**, dilakukan pengecekan terhadap 520 entri pada setiap atribut. Hasil audit menunjukkan bahwa seluruh atribut memiliki status *non-null*, yang berarti tidak ada satupun sel data yang kosong.

Audit Persentase Kekosongan: Untuk memperkuat hasil, dilakukan perhitungan persentase menggunakan fungsi **data.isnull().sum()** dibagi dengan panjang total data.

Kesimpulan Teknis: Audit mengonfirmasi bahwa dataset berada pada kondisi 100% lengkap (nilai kekosongan 0% untuk seluruh kolom). Dengan kondisi data yang paripurna ini, sistem tidak memerlukan teknik imputasi data, sehingga originalitas informasi medis tetap terjaga.

2.3.3. Audit Integritas Kolom (*Feature Uniqueness*)

Integritas dataset juga ditentukan oleh keunikan setiap fitur. Adanya kolom ganda atau duplikasi atribut dapat menyebabkan konflik saat algoritma melakukan pemetaan data dalam ruang multidimensi (Dichenko & Finko, 2021).

Verifikasi Nama Kolom: Pemeriksaan dilakukan menggunakan fungsi **data.columns.unique()**.

Hasil Verifikasi: Audit menunjukkan bahwa seluruh nama kolom bersifat unik. Hal ini menjamin bahwa setiap variabel gejala (seperti Polyuria dan Polydipsia) berdiri sendiri sebagai fitur yang berbeda dan tidak akan menyebabkan redundansi perhitungan dalam algoritma K-NN.

2.3.4. Relevansi Audit terhadap Performa K-NN

Audit teknis ini bukan sekadar formalitas, melainkan syarat mutlak bagi performa algoritma berbasis jarak. K-NN sangat sensitif terhadap data yang tidak konsisten. Dengan melakukan prosedur audit di atas, pengembang dapat menjamin bahwa input yang masuk ke tahap pemodelan adalah data yang telah tervalidasi secara teknis, sehingga hasil prediksi akhir memiliki tingkat kepercayaan yang tinggi.

2.4. Matriks Konsistensi dan Validasi Range Atribut

Setelah melalui prosedur audit dimensi dan kelengkapan, tahap selanjutnya dalam manajemen data adalah memastikan konsistensi internal antara nilai-nilai atribut. Konsistensi data menjamin bahwa setiap entri informasi mengikuti aturan (*rules*) yang telah ditetapkan berdasarkan karakteristik fitur masing-masing. Tanpa validasi ini, algoritma K-NN dapat menghasilkan prediksi yang menyesatkan akibat adanya data pencilan (outliers) atau nilai yang tidak logis.

Dalam penelitian ini, matriks konsistensi disusun untuk melakukan verifikasi terhadap tiga parameter utama: keunikan nilai kategorikal, ketiadaan deviasi nilai, dan ketepatan rentang numerik.

2.4.1. Konsistensi Nilai Unik Atribut Kategorikal

Atribut kategorikal dalam dataset diabetes ini sebagian besar bersifat biner. Oleh karena itu, pemeriksaan dilakukan untuk memastikan tidak ada variasi penulisan yang dapat dianggap sebagai kategori baru oleh sistem.

Kriteria Pemeriksaan: Nilai pada kolom gejala hanya boleh berisi Yes atau No, kolom gender hanya Male atau Female, dan kolom label target hanya Positive atau Negative.

Hasil Verifikasi: Berdasarkan pemeriksaan terhadap seluruh fitur kategorikal, ditemukan bahwa semua nilai berada dalam kategori yang tepat tanpa adanya ambiguitas penulisan.

2.4.2. Validasi Range Atribut Numerik (Usia)

Atribut usia (Age) memiliki karakteristik yang berbeda karena bersifat numerik kontinu. Validasi rentang (range validation) diperlukan untuk menjamin bahwa data yang diproses adalah data populasi yang relevan secara medis.

Kriteria Pemeriksaan: Nilai usia harus berada dalam rentang minimal 11 tahun dan maksimal 100 tahun.

Hasil Verifikasi: Seluruh data usia (520 baris) dinyatakan valid karena berada dalam batas parameter yang telah ditentukan. Hal ini memastikan bahwa model K-NN tidak akan terdistorsi oleh data usia yang tidak masuk akal (misalnya nilai nol atau negatif).

2.4.3. Matriks Hasil Pemeriksaan Konsistensi

Secara sistematis, hasil dari prosedur validasi ini dirangkum ke dalam sebuah matriks konsistensi sebagai berikut:

Tabel 2.2. Matriks Konsistensi Nilai Dataset

No.	Parameter Pemeriksaan	Kriteria Validitas	Hasil	Status
1	Keunikan Nilai Kolom	Kategorikal harus Yes/No atau Male/Female	Valid	Sesuai
2	Deviasi Nilai	Tidak ada nilai di luar kategori yang ditentukan	Valid	Sesuai
3	Rentang Usia	Berada di antara 11 hingga 100 tahun	Valid	Sesuai

2.4.4. Signifikansi Validasi terhadap Integritas Model

Penerapan matriks konsistensi ini memberikan jaminan bahwa data memiliki reliabilitas tinggi. Dalam konteks medis, validasi rentang usia dan konsistensi gejala adalah bentuk dari penyelarasan antara data teknis dengan fakta klinis. Data yang telah tervalidasi ini menjadi fondasi utama bagi tahap normalisasi dan pemodelan K-NN agar dapat menghasilkan akurasi prediksi yang optimal di tahap selanjutnya.

2.5. Interpretasi Klinis Fitur Prediksi

Sebuah sistem prediksi yang baik tidak hanya akurat secara matematis, tetapi juga relevan secara medis (*clinically relevant*). Pengembang sistem harus memahami konteks medis dari setiap fitur yang digunakan.

Mengambil studi kasus pada deteksi dini diabetes, relevansi fitur dapat dijelaskan sebagai berikut:

1. Poliuria dan Polidipsia: Kedua gejala ini sering disebut sebagai gejala klasik diabetes. Tubuh berusaha membuang kelebihan glukosa melalui urin

(poliuria), yang menyebabkan dehidrasi sehingga pasien merasa sangat haus (polidipsia).

2. Penurunan Berat Badan Tiba-tiba: Meskipun pasien makan banyak (polifagia), sel tubuh tidak mendapatkan energi dari glukosa karena kekurangan insulin, sehingga tubuh memecah cadangan lemak dan otot.
3. Kelemahan Parsial (*Partial Paresis*): Menggambarkan kelemahan otot sebagian yang bisa menjadi indikasi komplikasi neuropati diabetes.

Memahami konteks ini penting bagi pengembang sistem untuk melakukan *Feature Selection* (seleksi fitur). Tidak semua data relevan; namun dalam kasus dataset standar seperti *Early Stage Diabetes Risk Prediction Dataset* dari Kaggle, ke-16 atribut gejala yang ada telah dikurasi dan terbukti relevan untuk klasifikasi.

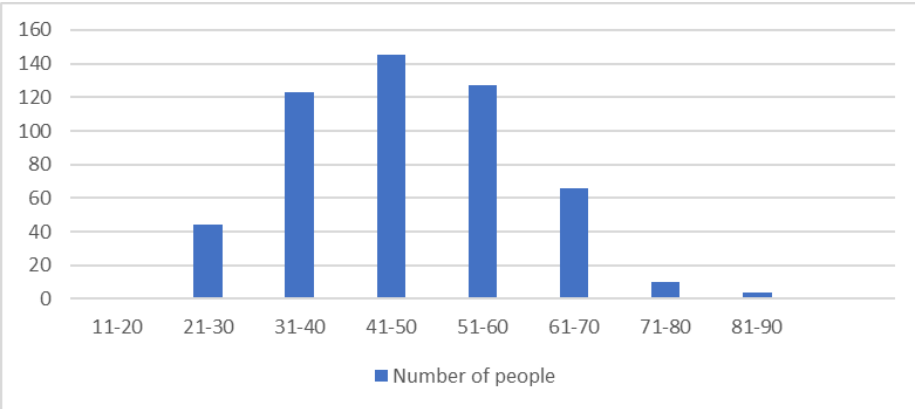
2.6. Analisis Distribusi Frekuensi Data

Setelah data divalidasi dan dibersihkan dari anomali, langkah analitik pertama yang wajib dilakukan sebelum pemodelan adalah memahami profil populasi dataset tersebut. Salah satu alat statistik deskriptif yang paling esensial dalam tahap ini adalah Analisis Distribusi Frekuensi.

Distribusi frekuensi memberikan gambaran makro mengenai bagaimana nilai-nilai dalam suatu atribut tersebar. Dalam informatika kesehatan, analisis ini bukan sekadar rutinitas statistik, melainkan jendela untuk melihat apakah dataset yang kita gunakan benar-benar merepresentasikan kondisi demografis dan epidemiologis di dunia nyata.

2.6.1. Profil Demografi Usia (*Age Distribution*)

Sebagai ilustrasi kasus, kita dapat mengamati atribut usia (Age) pada dataset prediksi risiko diabetes tahap awal (*Early Stage Diabetes Risk Prediction*). Usia pasien dalam dataset tidak dibiarkan sebagai angka acak, melainkan didiskritisasi ke dalam rentang interval 10 tahunan, mulai dari kelompok 11-20 tahun hingga 91-100 tahun.



Gambar 2.2. Grafik Histogram Distribusi Frekuensi Usia Pasien (Sumber: Data Latih Kaggle)

Jika kita memvisualisasikan data ini ke dalam bentuk histogram atau diagram batang, kita akan melihat sebuah pola distribusi yang spesifik. Berdasarkan observasi pada dataset studi kasus:

- Puncak Frekuensi (Modus): Konsentrasi pasien tertinggi berada pada rentang usia 41-50 tahun.
- Kelompok Mayoritas: Jika diperluas, mayoritas data menumpuk pada kelompok usia produktif dan paruh baya, yakni antara 31 hingga 60 tahun.
- Kelompok Minoritas: Frekuensi pasien pada usia sangat muda (11-20 tahun) dan usia sangat tua (di atas 71 tahun) memiliki jumlah yang sangat sedikit.

2.6.2. Interpretasi Klinis dari Pola Distribusi

Dari kacamata medis, distribusi yang condong atau menumpuk di usia 40-an hingga 50-an ini sangat logis dan sejalan dengan literatur kedokteran mengenai Diabetes Melitus Tipe 2. Penyakit ini umumnya mulai berkembang dan terdiagnosis pada usia paruh baya akibat akumulasi gaya hidup yang kurang sehat, penurunan fungsi metabolisme, dan resistensi insulin yang terjadi seiring bertambahnya usia.

Oleh karena itu, dataset ini dapat dikatakan valid secara epidemiologis karena distribusinya mencerminkan prevalensi diabetes di dunia nyata, bukan sekadar data buatan yang terdistribusi merata (seragam) secara tidak wajar.

2.6.3. Implikasi Distribusi terhadap Kinerja Algoritma K-NN

Memahami distribusi frekuensi sangat krusial karena berdampak langsung pada cara algoritma *K-Nearest Neighbor* (K-NN) membuat keputusan. K-NN adalah

algoritma yang mencari teman. Ia memprediksi nasib sebuah data baru berdasarkan mayoritas tetangga terdekatnya.

Distribusi usia yang tidak merata (menumpuk di tengah) membawa dua implikasi matematis:

- Akurasi Tinggi pada Kelas Mayoritas: Jika sistem menerima input pasien berusia 45 tahun, K-NN akan bekerja dengan sangat optimal. Ruang dimensi di sekitar usia 45 tahun sangat padat dengan data riwayat pasien lain. K-NN memiliki banyak referensi tetangga untuk menghasilkan prediksi yang sangat akurat.
- Ketidakpastian pada *Outliers* (Kelas Minoritas): Sebaliknya, jika sistem menerima input pasien berusia 15 tahun atau 85 tahun, model K-NN mungkin akan sedikit kesulitan. Karena jumlah sampel di rentang usia tersebut sangat jarang, jarak (Euclidean distance) untuk menemukan K tetangga terdekat akan menjadi lebih jauh. Tetangga terdekat yang ditemukan mungkin tidak memiliki kemiripan karakteristik yang identik, yang dapat menurunkan tingkat keyakinan (*confidence level*) dari prediksi tersebut.

2.6.4. Visualisasi sebagai Alat Komunikasi

Dalam pengembangan perangkat lunak klinis, analisis distribusi frekuensi harus selalu divisualisasikan. Diagram batang (*Bar Chart*) atau Histogram berfungsi sebagai alat komunikasi yang transparan antara data scientist dan pakar medis (dokter). Dengan melihat grafik frekuensi, pakar medis dapat segera menilai apakah data latih (training data) yang digunakan oleh sistem sudah cukup komprehensif atau masih memiliki bias demografis yang perlu diperbaiki di masa mendatang.

2.7. Tantangan Kualitas Data (*Data Quality*)

Dalam teori *Knowledge Discovery in Database* (KDD), kualitas data input sangat menentukan kualitas output model prediksi. Beberapa masalah umum pada data kesehatan meliputi:

2.7.1. Kelengkapan Data (*Missing Values*)

Data medis seringkali tidak lengkap karena pasien lupa menjawab pertanyaan atau alat ukur tidak tersedia.

a. Dampak: Algoritma K-NN tidak dapat menghitung jarak Euclidean jika ada koordinat (nilai) yang kosong.

b. Solusi: Teknik imputasi (pengisian nilai kosong dengan rata-rata/median) atau penghapusan data (Singh et al., 2026).
c. Studi Kasus: Dalam dataset ideal (seperti data Kaggle yang divalidasi), mungkin tidak ditemukan *missing values* (0% null). Namun, dalam implementasi nyata di rumah sakit, sistem harus disiapkan untuk menangani error jika pengguna mengosongkan formulir input.

2.7.2. Konsistensi Data

Inkonsistensi terjadi ketika format data berbeda-beda. Misalnya, penulisan kategori Laki-laki, Pria, Male, atau L dalam satu kolom yang sama.

a. Verifikasi: Sebelum pemodelan, perlu dilakukan pengecekan nilai unik (*unique values*) pada setiap kolom untuk memastikan tidak ada duplikasi kategori yang membingungkan algoritma.
b. Validasi Range: Untuk data numerik seperti usia, sistem harus memastikan nilai berada dalam rentang logis manusia (misal: 11-100 tahun). Data di luar rentang ini (outliers ekstrem) harus diinvestigasi validitasnya (Van Der Loo & De Jonge, 2021).

2.7.3. Keseimbangan Kelas (*Class Imbalance*)

Keseimbangan antara jumlah data kelas positif (sakit) dan negatif (sehat) sangat mempengaruhi performa K-NN.

Jika data latih didominasi oleh kelas Sehat (misal 90% sehat, 10% sakit), algoritma K-NN akan bias memprediksi data baru sebagai Sehat karena mayoritas tetangganya pasti sehat.

Kondisi ideal ialah algoritma K-NN bekerja secara optimal pada dataset dengan distribusi kelas yang relatif seimbang (Kim, 2021).

2.8. Struktur Data dalam Pemrograman Web (PHP Arrays)

Setelah data rekam medis melalui seluruh proses transformasi dan normalisasi, data tersebut tidak lagi berbentuk teks naratif atau tabel Excel, melainkan telah menjadi sekumpulan angka murni. Tantangan selanjutnya dalam pengembangan sistem berbasis web adalah: Bagaimana cara menyimpan dan menyajikan data numerik tersebut ke dalam memori server agar dapat diproses oleh algoritma Kecerdasan Buatan?

Dalam ekosistem pengembangan web berbasis PHP, struktur data yang paling fundamental dan efisien untuk menangani dataset *Machine Learning* adalah Array Multidimensi.

Dalam matematika matriks, sebuah dataset direpresentasikan sebagai matriks baris dan kolom. Konsep ini diterjemahkan ke dalam bahasa pemrograman PHP menggunakan struktur *Array of Arrays* (Array di dalam Array). Untuk memfasilitasi proses pelatihan model klasifikasi (model training), dataset wajib dipisahkan menjadi dua variabel array yang berbeda secara logis (Awad, 2024).

2.8.1. Matriks Fitur (Variabel $samples)

Variabel pertama sering disebut sebagai *Feature Matrix* atau matriks variabel independen (X). Dalam sistem ini, matriks tersebut direpresentasikan oleh variabel $samples.

Variabel **$samples** bertugas menampung seluruh atribut gejala pasien, mulai dari usia yang telah dinormalisasi hingga atribut obesitas. Struktur logikanya adalah array dua dimensi di mana: Baris (*Row*): Merepresentasikan satu entitas data pasien individu. Kolom (*Column*): Merepresentasikan nilai dari satu fitur atau gejala spesifik dari pasien tersebut.

Sebagai ilustrasi, jika kita memiliki 520 data pasien dan 16 fitur gejala, maka variabel $samples akan memiliki 520 elemen array utama, di mana setiap elemennya berisi array turunan yang masing-masing berjumlah 16 angka numerik.

Gambar 2.3. Array Dua Dimensi $samples

2.8.2. Vektor Target (Variabel $labels)

Variabel kedua dikenal sebagai Target Vector atau variabel dependen (Y). Dalam pemrograman sistem prediksi ini, digunakan variabel **$labels**.

Berbeda dengan **$samples** yang berupa array dua dimensi, **$labels** adalah array satu dimensi yang hanya berisi label kelas sasaran (status diagnosis akhir) untuk setiap pasien. Elemen di dalam array ini saling berkorespondensi secara eksak dengan indeks pada array **$samples**.

Artinya, nilai **$labels[0]** (misalnya bernilai 1 atau Positive) adalah hasil diagnosis mutlak untuk kombinasi gejala yang terdapat pada **$samples[0]**.

Gambar 2.4. Array $labels

2.8.3. Signifikansi Pemisahan Logika Data

Pemisahan dataset tunggal menjadi dua entitas memori (Fitur dan Target) bukanlah sebuah kebetulan, melainkan standar arsitektur dalam implementasi *Supervised Learning*.

Pemisahan ini dilakukan secara sengaja untuk memfasilitasi algoritma (seperti K-NN yang disediakan oleh library PHP-ML) agar dapat mempelajari pola hubungan sebab-akibat. Algoritma memerlukan variabel input **($samples**) untuk dikalkulasi jarak kedekatannya, dan memerlukan variabel output **($labels**) sebagai kunci jawaban (*ground truth*) untuk memvalidasi dan memetakan prediksi akhir.

Dengan mengubah bahasa pasien dari rekam medis kertas menjadi struktur Array PHP yang rapi dan terukur ini, data telah sepenuhnya siap untuk dieksekusi oleh mesin kalkulasi matematika algoritma kecerdasan buatan.

2.9. Persiapan Data untuk Algoritma Berbasis Jarak (*Distance-Based Algorithms*)

Algoritma *K-Nearest Neighbor* bekerja dengan prinsip geometri: menghitung jarak antara titik data baru dengan titik data lama dalam ruang multidimensi. Oleh karena itu, data harus dipersiapkan (dimasak) agar layak secara geometris.

2.9.1. Transformasi Numerik

Karena komputer hanya mengerti angka, langkah fundamental adalah mengubah seluruh atribut kategorikal menjadi numerik.

Binary Encoding ialah mengubah No menjadi 0 dan Yes menjadi 1.

Gender Encoding: Mengubah Male menjadi 0 dan Female menjadi 1 (atau sebaliknya). Proses ini memungkinkan atribut kualitatif dihitung dalam rumus matematika.

2.9.2. Penskalaan Fitur (*Feature Scaling*)

Ini adalah aspek terpenting dalam penggunaan K-NN pada data campuran (numerik dan biner). Bayangkan kita menghitung jarak antara dua pasien:

- Atribut A (Usia): Selisih 20 tahun.

- Atribut B (Gejala Haus): Selisih 1 (Ya vs Tidak).

Tanpa penskalaan, selisih usia (20) akan menenggelamkan selisih gejala (1). Jarak total akan didominasi oleh usia. Untuk mencegah bias ini, teknik *Min-Max* Scaling diterapkan untuk memaksa semua atribut—termasuk usia—berada dalam rentang yang seragam, biasanya antara 0 hingga 1. Hal ini menjamin bahwa setiap gejala memiliki kontribusi yang setara (*fair contribution*) dalam penentuan diagnosis.

Untuk menyamakan kontribusi setiap fitur dalam perhitungan jarak Euclidean, kita menggunakan metode *Min-Max Scaling*. Rumus dasar untuk mentransformasikan nilai asli n_i menjadi nilai ternormalisasi n_i^1 dalam rentang $[0, 1]$ adalah sebagai berikut:

$$n_i^1 = \frac{n_i - min_A}{max_A - min_A}$$

Ilustrasi Perhitungan Manual:
Misalkan kita memiliki atribut Age yang telah didiskritisasi ke dalam nilai range 1 hingga 9 ($min = 1, maks = 9$). Jika seorang pasien berada pada kategori usia 41-50 tahun (Nilai asli $n_i = 4$), maka proses normalisasinya adalah:

- Identifikasi nilai: $n_i = 4, min = 1, maks = 9$.
- Substitusi ke rumus:

$$n_i^1 = \frac{4 - 1}{9 - 1} = \frac{3}{8}$$

- Hasil Normalisasi: 0,375.

Dengan hasil ini, variabel usia yang tadinya bernilai 4 kini memiliki skala yang sama dengan variabel biner lainnya (0 atau 1), sehingga tidak mendominasi perhitungan jarak pada algoritma K-NN.

2.9.3. Diskritisasi (*Discretization*)

Dalam beberapa kasus, mengubah data numerik kontinu menjadi data kategori (rentang) dapat membantu menyederhanakan pola, terutama jika variasi data terlalu tinggi namun trennya berkelompok.

Contoh: Mengelompokkan usia menjadi Remaja, Dewasa, Lansia atau rentang 10 tahunan (21-30, 31-40, dst). Manfaatnya dapat menyederhanakan input model dan mengurangi dampak *noise* dari fluktuasi umur yang kecil. Namun, perlu diingat bahwa diskritisasi juga menghilangkan detail informasi presisi.

Tabel 2.3. Data sebelum *pre-processing*

Age	Gender	Polyuria	Polydipsia	…	Class
16	Male	Yes	No	…	Positive
25	Female	No	No	…	Positive
25	Male	Yes	Yes	…	Positive
26	Male	No	No	…	Negative
…	…	…	…	…	…
90	Female	No	Yes	…	Positive
90	Female	No	Yes	…	Positive

Tabel 2.4. Data Setelah *pre-processing*

Age	Gender	Polyuria	Polydipsia	…	Class
1	0	1	0	…	1
2	1	0	0	…	1
2	0	1	1	…	1
2	0	0	0	…	0
…	…	…	…	…	…
8	1	0	1	…	1
8	1	0	1	…	1

2.10. Rangkuman

Bab ini telah mengulas bahwa data kesehatan adalah entitas yang kompleks yang memerlukan penanganan khusus. Dari identifikasi tipe data hingga proses normalisasi, setiap langkah data preparation bertujuan untuk mengubah data mentah menjadi format yang dapat dicerna oleh algoritma K-NN.

Data yang bersih, lengkap, dan memiliki skala yang seragam adalah prasyarat mutlak. Kegagalan dalam tahap ini akan menyebabkan model prediksi yang bias, tidak peduli seberapa canggih algoritma yang digunakan.

Pada bab selanjutnya, kita akan melangkah dari pembahasan Data menuju pembahasan Algoritma. Kita akan membedah secara matematis bagaimana algoritma *K-Nearest Neighbor* (K-NN) memanfaatkan data yang telah kita persiapkan di bab ini untuk melakukan klasifikasi risiko penyakit.

BAB 3 KONSEP DASAR DAN ANATOMI MATEMATIS *K-NEAREST NEIGHBOR* (K-NN)

3.1. Pengantar *Machine Learning* dan *Supervised Learning*

Sebelum kita menyelami mekanisme spesifik dari algoritma *K-Nearest Neighbor* (K-NN), penting untuk meletakkan fondasi pemahaman mengenai ekosistem di mana algoritma ini beroperasi. K-NN adalah bagian dari cabang ilmu Kecerdasan Buatan (*Artificial Intelligence*) yang dikenal sebagai *Machine Learning* (Pembelajaran Mesin). Secara esensial, *Machine Learning* adalah studi tentang algoritma komputer yang dapat belajar dan berkembang dari pengalaman (data) tanpa harus diprogram secara eksplisit untuk setiap skenario yang mungkin terjadi.

Dalam taksonomi *Machine Learning*, algoritma secara umum dibagi ke dalam beberapa paradigma pembelajaran. K-NN masuk ke dalam kategori Pembelajaran Terarah (*Supervised Learning*).

3.1.1. Definisi *Supervised Learning*

Supervised Learning adalah pendekatan pembelajaran mesin di mana model dilatih menggunakan himpunan data yang sudah memiliki label kebenaran absolut (*ground truth*). Dalam pendekatan ini, setiap sampel data latih terdiri dari pasangan input (sering disebut sebagai fitur atau atribut) dan output yang diinginkan (disebut sebagai label atau kelas).

Tujuan utama dari *Supervised Learning* adalah memetakan fungsi dari input ke output berdasarkan contoh pasangan input-output yang diberikan. Jika kita membayangkan sebuah sistem pendidikan, *Supervised Learning* ibarat seorang siswa yang belajar memecahkan soal matematika dengan bimbingan seorang guru. Guru memberikan soal (fitur) beserta kunci jawabannya (label). Setelah siswa mempelajari berbagai pola soal dan kunci jawaban tersebut, ia diuji dengan soal baru yang belum pernah ia lihat sebelumnya. Diharapkan, siswa tersebut dapat memberikan jawaban yang benar berdasarkan pola yang telah ia pelajari (Sen et al., 2020).

3.1.2. Klasifikasi vs Regresi

Dalam payung *Supervised Learning*, terdapat dua jenis permasalahan utama yang dapat diselesaikan:

a. Regresi: Jika output yang diprediksi berupa nilai numerik yang kontinu (Saharan et al., 2021). Contoh: memprediksi harga rumah berdasarkan luas tanah, atau memprediksi kadar gula darah spesifik seseorang.

b. Klasifikasi: Jika output yang diprediksi berupa kategori atau kelas yang diskrit (An et al., 2023). Contoh: memprediksi apakah sebuah email adalah Spam atau Bukan Spam, atau dalam konteks buku ini, memprediksi apakah seorang pasien Positif Berisiko Diabetes atau Negatif.

Meskipun K-NN dapat digunakan untuk regresi, dalam ranah informatika kesehatan dan diagnostik medis, K-NN paling sering dan paling kuat digunakan sebagai algoritma Klasifikasi. Algoritma ini akan mengkategorikan pasien baru ke dalam kelompok risiko tertentu berdasarkan data rekam medis historis yang telah dilabeli sebelumnya.

3.2. Filosofi dan Intuisi di Balik *K-Nearest Neighbor*

Sebagian besar algoritma *Machine Learning* modern—seperti *Neural Networks* atau *Support Vector Machines*—melibatkan proses matematika yang sangat kompleks dan seringkali dianggap sebagai kotak hitam (*black box*). Sebaliknya, *K-Nearest Neighbor* adalah salah satu algoritma yang paling transparan, intuitif, dan selaras dengan cara berpikir alami manusia.

3.2.1. Analogi Kedekatan Sosial

Intuisi dasar dari K-NN dapat diringkas dalam sebuah pepatah lama yang sangat terkenal:

"*Tell me who your friends are, and I will tell you who you are.*" (Beri tahu saya siapa teman-temanmu, dan saya akan memberi tahu siapa dirimu).

Dalam kehidupan sosial, manusia cenderung berkumpul dan membentuk kelompok dengan individu-individu lain yang memiliki minat, latar belakang, atau karakteristik yang serupa. Jika kita melihat seseorang yang dikelilingi oleh orang-orang yang gemar membaca buku, gemar berdiskusi tentang sains, dan sering menghabiskan waktu di perpustakaan, kita secara intuitif akan menyimpulkan atau memprediksi bahwa orang tersebut kemungkinan besar adalah seorang akademisi atau kutu buku.Kita membuat kesimpulan tersebut bukan dengan menghitung rumus matematika yang rumit di dalam kepala kita, melainkan dengan melihat mayoritas karakteristik dari tetangga terdekat (*nearest neighbors*) di sekitar orang tersebut (Cunningham & Delany, 2022).

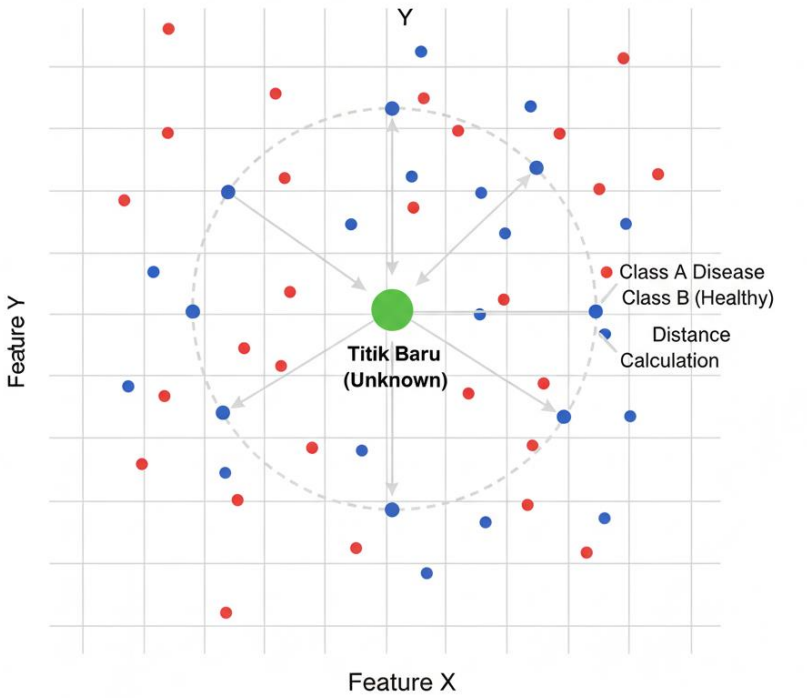
3.2.2. Penerapan Intuisi pada Data Medis

Filosofi ini diterjemahkan secara harfiah ke dalam dunia medis oleh algoritma K-NN. Dalam diagnosis klinis, penyakit memanifestasikan dirinya

melalui serangkaian gejala yang khas. Dua orang pasien yang menderita penyakit yang sama cenderung akan menunjukkan pola gejala yang mirip.

Misalnya, kita memiliki pasien baru yang mengeluhkan rasa haus ekstrem (Polydipsia), sering buang air kecil (Polyuria), dan kelelahan. Bagaimana K-NN memprediksi risikonya?

Algoritma ini tidak berusaha membuat aturan medis baku. Sebaliknya, ia akan melihat ke dalam pangkalan data (database) riwayat rekam medis pasien-pasien terdahulu. Ia akan mencari sejumlah *K* pasien masa lalu yang pola gejalanya paling mirip (atau posisinya paling dekat) dengan pasien baru tersebut.



Gambar 3.1. Analogi Algoritma *K-Nearest Neighbor* pada Ruang Koordinat

Jika mayoritas dari *K* pasien terdahulu tersebut ternyata adalah penderita diabetes, maka algoritma akan mengambil kesimpulan logis: pasien baru ini juga memiliki risiko diabetes. Inilah mengapa algoritma ini disebut sebagai *K-Nearest Neighbor* (K-Tetangga Terdekat).

3.3. Anatomi Matematis K-NN: Metrik Jarak (*Distance Metrics*)

Meskipun konsepnya sederhana secara intuitif, agar komputer dapat mengenali siapa tetangga terdekat tersebut, kita harus menerjemahkan konsep kemiripan ke dalam bahasa matematika kuantitatif. Dalam geometri dan aljabar linear, kemiripan antara dua titik data diukur menggunakan konsep Jarak (Distance). Semakin kecil jarak antara dua titik, semakin tinggi kemiripan di antara keduanya.

Setiap pasien direpresentasikan sebagai sebuah titik koordinat dalam ruang multidimensi. Jika kita menggunakan 16 gejala medis sebagai fitur prediktor (seperti usia, jenis kelamin, poliuria, polifagia, dll.), maka setiap pasien hidup dalam ruang berdimensi 16. Tugas metrik jarak adalah menghitung seberapa jauh satu pasien dengan pasien lainnya di dalam ruang dimensi tinggi tersebut.

Terdapat berbagai variasi rumus perhitungan jarak, di mana setiap rumus memiliki karakteristik dan peruntukannya masing-masing.

3.3.1. *Euclidean Distance* (Jarak Garis Lurus)

Euclidean distance adalah metrik jarak yang paling populer, paling standar, dan paling sering diimplementasikan dalam berbagai pustaka *Machine Learning*, termasuk dalam algoritma K-NN standar. Konsep ini merupakan generalisasi multidimensi dari Teorema Pythagoras yang kita pelajari di sekolah dasar.

Jarak Euclidean mengukur panjang lintasan garis lurus terpendek yang menghubungkan dua titik dalam ruang Euclidean. Jika kita memiliki data pasien baru (diwakili oleh titik *q*) dan data pasien riwayat dari database (diwakili oleh titik *p*), serta keduanya memiliki *n* buah fitur (gejala), maka rumus jarak Euclidean didefinisikan secara formal sebagai berikut:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

Atau dalam notasi sigma yang lebih ringkas:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Penjelasan Komponen:

- $d(p,q)$: Jarak *Euclidean* antara pasien *p* dan pasien *q*.
- n : Jumlah total fitur atau atribut (dalam kasus prediksi diabetes ini, $n = 16$).
- q_i : Nilai fitur ke-*i* dari pasien uji (data baru).

- p_i : Nilai fitur ke-*i* dari pasien latih (data historis di database).

Penggunaan kuadrat dalam rumus ini bertujuan untuk dua hal: pertama, menghilangkan nilai negatif sehingga perbedaan arah tidak membatalkan perbedaan magnitudo; kedua, memberikan penalti yang lebih besar (pembobotan lebih) pada fitur-fitur yang memiliki perbedaan sangat ekstrem antar dua pasien. Inilah alasan mengapa tahap normalisasi (seperti Min-Max Scaling) sangat kritis sebelum menggunakan Euclidean Distance; tanpa normalisasi, satu fitur dengan skala besar (misal: gaji jutaan rupiah, atau usia 80 tahun) akan sepenuhnya mendominasi nilai kuadrat ini dan mengabaikan fitur biner lainnya (seperti Ya/Tidak) (Mukherjee et al., 2024).

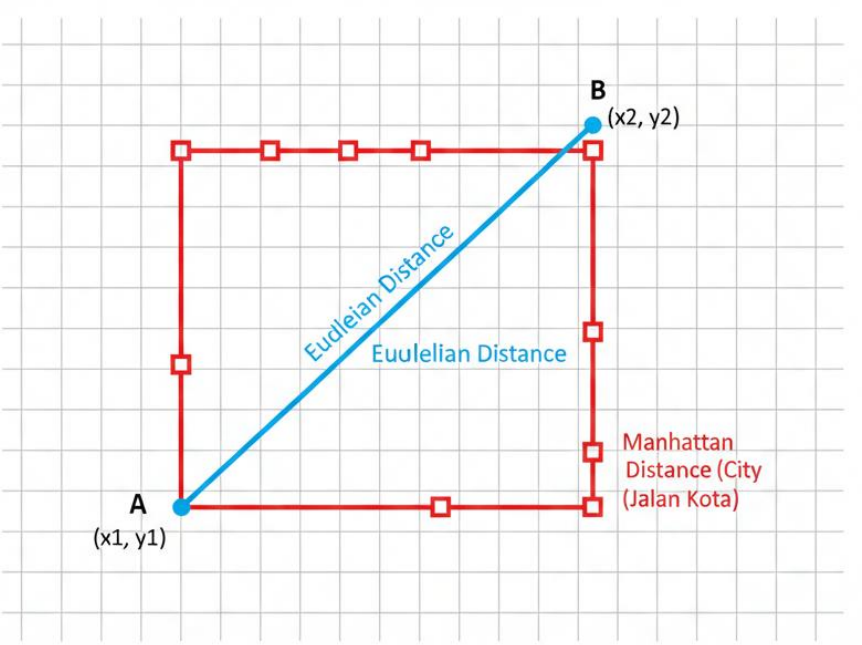
3.3.2. *Manhattan Distance* (Jarak Blok Kota)

Selain Euclidean, terdapat metrik alternatif yang dikenal sebagai Manhattan Distance atau sering juga disebut *City Block Distance* atau *L1-Norm*. Nama ini terinspirasi dari tata letak jalanan di pulau Manhattan, New York, yang berbentuk kisi-kisi (grid) tegak lurus.

Berbeda dengan Euclidean yang memotong jalan dalam garis lurus, *Manhattan distance* menghitung jarak berdasarkan pergerakan vertikal dan horizontal murni (tidak ada garis diagonal). Rumus matematikanya adalah akumulasi dari selisih absolut antar koordinat:

$$d(p,q) = \sum_{i=1}^{n} |q_i - p_i|$$

Manhattan distance seringkali lebih disukai daripada Euclidean ketika kita berhadapan dengan data yang memiliki banyak *outlier* (pencilan). Karena Manhattan tidak mengkuadratkan selisih nilai, ia tidak memberikan penalti eksponensial terhadap perbedaan yang ekstrem, sehingga model menjadi lebih *robust* (tangguh) terhadap data *noise*. Selain itu, pada dataset yang memiliki dimensi sangat tinggi (*high-dimensional space*), Manhattan kadang-kadang mampu memitigasi efek dari kutukan dimensi (*Curse of Dimensionality*) lebih baik daripada Euclidean(Rustam & Usman, 2025).



Gambar 3.2. Perbandingan Euclidean vs Manhattan

3.3.3. *Minkowski Distance* (Generalisasi Jarak)

Minkowski distance sebenarnya bukanlah sebuah metrik tunggal, melainkan formula matematis umum (generalisasi) yang merangkum baik Euclidean maupun Manhattan distance ke dalam satu persamaan parametrik.

Rumus *Minkowski Distance* atau *L-p Norm* didefinisikan sebagai:

$$d(p,q) = \left(\sum_{i=1}^{n} |q_i - p_i|^p\right)^{\frac{1}{p}}$$

Parameter p adalah sebuah bilangan riil ($p \geq 1$) yang menentukan ordo dari metrik jarak tersebut. Jika $p = 1$, rumus Minkowski akan mereduksi (berubah) persis menjadi rumus Manhattan Distance. Jika $p = 2$, rumus Minkowski akan mereduksi (berubah) persis menjadi rumus Euclidean Distance (karena pangkat $1/2$ sama dengan akar kuadrat).

Pengenalan berbagai variasi jarak ini penting bagi seorang praktisi Data Science. Memilih metrik jarak yang tepat adalah seni tersendiri dan sangat

bergantung pada karakteristik distribusi data medis yang sedang ditangani (Mailagaha Kumbure & Luukka, 2021).

3.4. Mekanisme Klasifikasi dan Majority Voting

Setelah jarak antara pasien baru dengan seluruh pasien di dalam pangkalan data berhasil dihitung (misalnya menggunakan fungsi Euclidean), langkah selanjutnya adalah proses klasifikasi sesungguhnya. Proses ini terdiri dari dua tahap utama: Pemilahan (*Sorting*) dan Pemilihan (*Voting*).

3.4.1. Mengidentifikasi Tetangga Terdekat

Algoritma akan mengumpulkan seluruh nilai jarak yang telah dihitung, kemudian mengurutkannya secara *ascending* (dari nilai jarak terkecil hingga terbesar). Jarak terkecil merepresentasikan tingkat kemiripan tertinggi.

Komputer kemudian akan mengambil *K* buah data dengan jarak terkecil tersebut. Himpunan *K* data inilah yang resmi menyandang status sebagai Tetangga Terdekat (*Nearest Neighbors*) dari pasien uji.

3.4.2. Aturan Keputusan (*Decision Rule*): *Simple Majority Voting*

Metode standar yang digunakan oleh K-NN untuk menentukan kelas akhir dari pasien baru adalah melalui pemungutan suara mayoritas atau *Majority Voting*. Sistem akan melihat label kelas apa yang paling banyak muncul di antara himpunan *K* tetangga tersebut.

Sebagai ilustrasi teoretis, misalkan kita menetapkan parameter $K = 5$. Setelah menghitung jarak, ditemukan 5 pasien dengan jarak terdekat dari pasien baru. Dari kelima pasien historis tersebut, kita periksa label diagnosis asli mereka: Tetangga 1: Positif Diabetes, Tetangga 2: Positif Diabetes, Tetangga 3: Negatif, Tetangga 4: Positif Diabetes, Tetangga 5: Negatif.

Sistem akan melakukan pemungutan suara: skor untuk kelas Positif Diabetes adalah 3, sedangkan skor untuk Negatif adalah 2. Berdasarkan prinsip *Majority Voting*, sistem akan memprediksi dan menyimpulkan bahwa pasien baru tersebut masuk ke dalam kelas Positif Diabetes.

Catatan Teoretis: Inilah alasan mengapa dalam kasus klasifikasi biner atau dua kelas, para ahli sangat menyarankan untuk menggunakan nilai *K* yang ganjil (seperti 1, 3, 5, 7) untuk menghindari kondisi seri atau tie.

3.5. Kompleksitas Model dan Teori Pemilihan Nilai *K*

Satu-satunya parameter utama yang harus disetel (tuning) oleh pengembang dalam algoritma K-NN adalah nilai *K* itu sendiri. Huruf *K* merepresentasikan konstanta bilangan bulat positif yang menentukan jumlah tetangga yang akan dipertimbangkan dalam proses pengambilan keputusan.

Pertanyaan mendasar yang selalu muncul dalam implementasi K-NN adalah: Berapakah nilai K yang paling optimal?. Secara matematis dan teoretis, tidak ada satu nilai *K* ajaib yang berlaku untuk semua jenis dataset. Nilai *K* mengendalikan keseimbangan antara Bias (Kesalahan akibat asumsi model yang terlalu sederhana) dan Variance (Kesalahan akibat model yang terlalu sensitif terhadap fluktuasi data).

Pemilihan nilai *K* akan menghasilkan dua ekstrem yang harus dihindari: *Overfitting* dan *Underfitting*.

3.5.1. Dampak *K* Terlalu Kecil: Risiko *Overfitting*

Jika kita mengatur nilai *K* sangat kecil (misalnya, nilai minimum ekstrem *K*=1), sistem hanya akan melihat satu tetangga terdekat tunggal untuk mengambil keputusan.

Secara teoretis, model dengan *K* kecil akan menghasilkan batas keputusan (*decision boundary*) yang sangat rumit, bergelombang, dan bergerigi. Model ini memiliki Bias yang sangat rendah karena ia menyesuaikan diri dengan sangat ketat terhadap data latih, namun ia memiliki Variance yang sangat tinggi.

Kondisi ini disebut *Overfitting*. Kelemahan utamanya adalah model menjadi sangat sensitif terhadap *noise* (derau) atau anomali data. Jika kebetulan satu tetangga terdekat tersebut adalah pasien dengan data yang salah catat (*outlier*), maka pasien baru akan langsung didiagnosis secara keliru karena sistem buta terhadap tren mayoritas di sekitarnya. Model gagal menggeneralisasi pola dan hanya menghafal data latih.

3.5.2. Dampak *K* Terlalu Besar: Risiko Underfitting

Di sisi lain spektrum, apa yang terjadi jika kita menetapkan nilai *K* yang sangat besar? (Misalnya *K*=100 pada dataset berisi 500 pasien).

Dalam skenario ini, sistem mempertimbangkan terlalu banyak tetangga. Batas keputusan (*decision boundary*) akan menjadi sangat halus dan cenderung mengabaikan pola lokal yang spesifik. Model ini memiliki Variance yang rendah namun Bias yang tinggi, sebuah kondisi yang dikenal sebagai Underfitting.

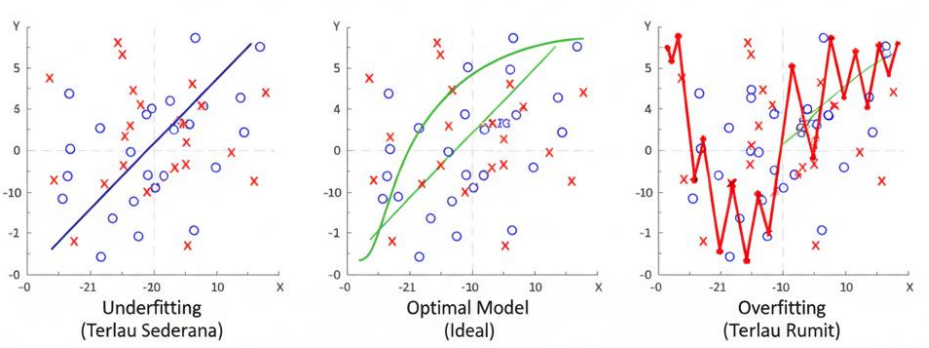
Kelemahan fatal dari *K* yang terlalu besar adalah model akan selalu didominasi oleh kelas mayoritas di dalam dataset. Jika di dalam pangkalan data terdapat 80% pasien diabetes dan 20% sehat, maka siapapun pasien baru yang

masuk, hasil pemungutan suara kemungkinan besar akan selalu dimenangkan oleh kelas diabetes, karena tetangga yang dipertimbangkan mencakup sebagian besar populasi. Model kehilangan kemampuannya untuk membedakan detail fitur yang subtil.

3.5.3. Strategi Penentuan *K* Optimal (*Cross-Validation*)

Karena nilai *K* tidak bisa ditebak secara acak, ia harus ditemukan secara empiris (trial and error yang terukur). Salah satu metode teoretis terbaik untuk mencari *K* yang optimal adalah melalui *K-Fold Cross-Validation*.

Dalam teknik ini, dataset tidak hanya dibagi sekali menjadi data latih dan data uji. Alih-alih, data dibagi menjadi lipatan-lipatan (*folds*) yang dievaluasi secara bergantian. Pengembang akan menjalankan algoritma K-NN berulang kali dengan nilai *K*=1, *K*=2, *K*=3, dan seterusnya. Nilai *K* yang menghasilkan rata-rata akurasi evaluasi tertinggi dengan tingkat kesalahan terkecil pada pengujian lintas-lipatan (*cross-validation*) itulah yang akhirnya ditetapkan sebagai parameter permanen untuk sistem produksi.



Gambar 3.3. Perbandingan Visual Model *Underfitting*, Optimal, dan *Overfitting* pada Klasifikasi

3.6. Karakteristik K-NN: Paradigma *Lazy Learning*

Sebagai penutup pembedahan teori K-NN, penting bagi pembaca untuk memahami bahwa K-NN memiliki paradigma komputasi yang unik dibandingkan algoritma *Machine Learning* lainnya. K-NN diklasifikasikan sebagai tipe algoritma *Lazy Learning* (Pembelajaran Malas) atau *Instance-based Learning*.

Istilah malas di sini bukan bermakna konotasi negatif, melainkan menggambarkan efisiensi di fase awal. Sebagian besar algoritma cerdas (seperti *Neural Network* atau *Decision Tree*) adalah *Eager Learners*; mereka menghabiskan

waktu berjam-jam atau berhari-hari untuk melakukan fase komputasi pelatihan model (*training phase*) di mana mereka membangun formula atau fungsi pemisah sebelum memproses data baru.

Sebaliknya, K-NN tidak memiliki fase pelatihan yang eksplisit. Saat data rekam medis pasien (seperti variabel **$samples** dan **$labels**) dimasukkan ke dalam memori, K-NN tidak melakukan perhitungan matematika apapun. Ia hanya menghafal dan menyimpan data tersebut di dalam ruang memori.

Seluruh komputasi matematika yang berat (menghitung rumus Euclidean, mensortir array, dan menghitung hasil voting) ditunda dan baru dieksekusi secara instan ketika ada permintaan (query) pasien baru yang masuk untuk diprediksi.

Sifat Lazy Learning ini menjadikan K-NN sangat ideal dan adaptif untuk diimplementasikan ke dalam aplikasi berbasis Web (menggunakan PHP). Ketika ada data pasien baru yang ditambahkan ke database (misalnya oleh pihak admin), sistem tidak perlu melakukan proses training ulang dari nol. Pengetahuan K-NN secara otomatis terbarui secara real-time. Namun, *trade-off* dari sifat ini adalah beban komputasi server akan meningkat pada saat fase testing atau fase prediksi, terutama jika jumlah data historis di dalam database mencapai ratusan ribu baris (Zhang & Li, 2023).

3.7. Rangkuman Konseptual

Bab ini telah mengartikulasikan dasar filosofis dan fondasi matematis dari algoritma *K-Nearest Neighbor* (K-NN). Kita memulai dari intuisi kedekatan sosial, lalu menerjemahkannya ke dalam metrik pengukuran jarak objektif seperti rumus Euclidean Distance. Kita juga telah membedah anatomi pengambilan keputusan melalui prinsip *Majority Voting* dan teori kritis dalam menyeimbangkan parameter *K* agar terhindar dari bias *Underfitting* maupun *Overfitting*.

Konsep *Lazy Learning* membuktikan bahwa kecerdasan sistem prediksi tidak selalu berasal dari persamaan matematis yang rumit, melainkan dari pemanfaatan secara cerdas atas data historis yang telah terakumulasi. Berbekal pemahaman teoritis tentang anatomi perhitungan jarak ini, kita kini memiliki landasan akademis yang kokoh untuk melangkah ke bab-bab teknis berikutnya.

Pada bab selanjutnya, kita akan mulai mendaratkan konsep matematika ini ke dalam realita data medis. Kita akan membedah secara langsung karakteristik dataset risiko diabetes tahap awal dan menelaah taktik preprocessing yang diwajibkan oleh formula-formula di bab ini sebelum sistem dihidupkan melalui baris-baris kode pemrograman.

BAB 4 ARSITEKTUR DAN PEMODELAN SISTEM CERDAS KESEHATAN

4.1. Rekayasa Perangkat Lunak dalam Informatika Kesehatan

Transformasi ide dan teori matematika murni—seperti yang telah dibahas pada bab-bab sebelumnya mengenai algoritma *K-Nearest Neighbor* (K-NN)—menjadi sebuah produk teknologi yang dapat digunakan oleh masyarakat luas memerlukan disiplin ilmu tersendiri. Disiplin tersebut adalah Rekayasa Perangkat Lunak (*Software Engineering*). Dalam domain informatika kesehatan (*Health Informatics*), pendekatan coba-coba (*trial and error*) dalam penulisan kode program sangat tidak disarankan karena sistem ini berkaitan langsung dengan data medis dan potensi keselamatan pengguna.

Oleh karena itu, pembangunan sistem prediksi risiko penyakit berbasis web harus dipandu oleh kerangka kerja yang terstruktur, metodologis, dan dapat diaudit pada setiap fase pengembangannya.

4.1.1. Pendekatan *Software Development Life Cycle* (SDLC)

Software Development Life Cycle (SDLC) atau Siklus Hidup Pengembangan Perangkat Lunak adalah kerangka kerja konseptual yang mendeskripsikan tahapan-tahapan yang terlibat dalam proses rekayasa sistem informasi, mulai dari studi kelayakan awal hingga pemeliharaan aplikasi pasca-peluncuran. SDLC menyediakan cetak biru bagi pengembang untuk memastikan bahwa perangkat lunak yang dihasilkan memenuhi standar kualitas, efisiensi waktu, dan keakuratan fungsional.

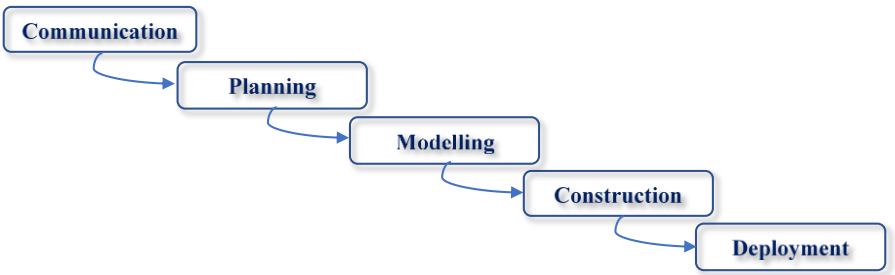
Dalam konteks pengembangan sistem cerdas prediktif, SDLC memastikan bahwa tidak ada kesenjangan antara kemampuan algoritma *Machine Learning* di sisi *backend* (server) dengan kebutuhan pengguna akhir (pasien atau staf medis) di sisi frontend (antarmuka) (Chahar & Singh, 2024).

4.1.2. Justifikasi Penggunaan Model Waterfall

Terdapat berbagai model SDLC modern seperti Agile, Scrum, atau Iterative. Namun, untuk pengembangan purwarupa sistem pakar medis atau sistem prediksi penyakit tahap awal yang memiliki batasan masalah dan parameter dataset yang sangat jelas sejak awal (seperti dataset gejala diabetes dengan 16 atribut pasti), Model Waterfall seringkali menjadi pilihan arsitektural yang sangat solid dan relevan (Saravanos & Curinga, 2023).

Model Waterfall mengedepankan alur kerja yang sekuensial (berurutan) dan sistematis. Setiap tahapan harus diselesaikan secara definitif dan divalidasi sebelum aliran proses jatuh ke tahap berikutnya. Karakteristik ini sangat sejalan dengan penanganan data medis, di mana tahapan pra-pemrosesan data tidak boleh diubah-ubah secara mendadak saat algoritma sudah mulai diprogram.

Pendekatan sekuensial ini menjamin bahwa setiap tahapan pengembangan sistem terdokumentasi dengan baik, di mana alur metodologi *Waterfall* yang diadopsi dalam pengembangan sistem ini dapat dilihat pada Gambar 4.1.



Gambar 4.1. Tahapan Metodologi Pengembangan Sistem dengan Model *Waterfall*

Dalam pengembangan sistem deteksi dini berbasis K-NN, tahapan Waterfall diterjemahkan ke dalam formulasi operasional sebagai berikut:

1. *Communication* (Komunikasi dan Analisis Kebutuhan): Fase ini adalah fondasi awal. Pengembang harus memahami domain medis dari masalah yang dipecahkan. Pada tahap ini dilakukan evaluasi ekstensif terhadap sumber data klinis. Misalnya, memvalidasi kelengkapan Early Stage Diabetes Risk Prediction Dataset, mengidentifikasi 17 atribut secara rinci, serta memahami konteks klinis dari gejala seperti poliuria dan polifagia. Kesalahan dalam memahami data pada tahap ini akan berakibat fatal pada keseluruhan sistem prediktif.
2. *Planning* (Perancangan Sistem): Berbekal pemahaman data, pengembang mulai menyusun arsitektur sistem. Fase ini berfokus pada pemilihan tumpukan teknologi (*tech stack*), yang dalam hal ini merujuk pada ekosistem web berbasis native PHP dan manajemen basis data MySQL. Seluruh komponen direpresentasikan dalam bentuk visual seperti *Context Diagram* dan *Entity Relationship Diagram* (ERD).
3. *Modelling* (Pemodelan Data dan Algoritma): Fase ini adalah inti dari kecerdasan sistem. Berbeda dengan aplikasi web biasa, sistem cerdas

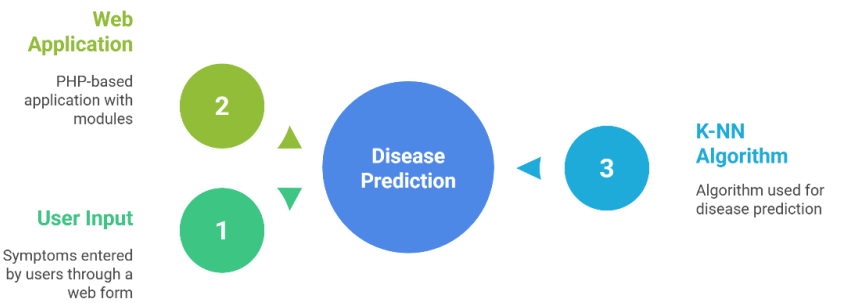
memerlukan tahap pemodelan di mana data mentah dikonversi menjadi format array matematis. Di sinilah algoritma K-NN didefinisikan, parameter *K* dioptimasi (misalnya melalui komparasi *K*=1, 3, 5), dan skenario normalisasi seperti *Min-Max Scaling* dirumuskan secara logis.

4. *Construction* (Konstruksi dan Penulisan Kode): Fase penerjemahan model menjadi baris-baris kode fungsional. Konstruksi mencakup pembuatan modul input (formulir kesehatan), pembuatan modul klasifikasi yang menghitung jarak Euclidean, hingga antarmuka hasil prediksi yang akan dibaca oleh pengguna.
5. *Deployment* (Penyebaran dan Pengujian Operasional): Aplikasi yang telah selesai dikonstruksi kemudian diunggah ke Virtual Private Server (VPS) agar dapat diakses secara publik. Tahap ini juga mencakup pengujian stabilitas lingkungan produksi (konfigurasi web server seperti Apache/Nginx, serta PHP 8.2) dan validasi apakah prediksi sistem di web cocok dengan teori perhitungan manual.

4.2. Arsitektur Sistem Terintegrasi (*Three-Tier Architecture*)

Sistem prediksi risiko kesehatan modern tidak dapat lagi dibangun dalam bentuk aplikasi monolitik yang kaku. Agar sebuah sistem dapat melayani ratusan atau ribuan pasien secara komputasional dengan lancar, ia harus dipecah ke dalam lapisan-lapisan logis yang independen. Arsitektur yang paling lazim dan kokoh untuk mengimplementasikan Machine Learning di lingkungan web adalah Arsitektur Tiga Lapis (*Three-Tier Architecture*).

Arsitektur ini memisahkan aplikasi menjadi tiga entitas yang saling berkomunikasi namun memiliki tanggung jawab komputasi yang terisolasi (GabAllah et al., 2023).



Gambar 4.2. Arsitektur Tiga Lapis (*Three-Tier Architecture*) pada Sistem Prediksi Kesehatan.

Untuk menjaga skalabilitas dan keamanan data medis, sistem ini dirancang menggunakan arsitektur tiga lapis yang memisahkan antara antarmuka pengguna, logika pemrosesan, dan penyimpanan data, sebagaimana diilustrasikan pada Gambar 4.2 di atas.

4.2.1. Lapisan Presentasi (*Presentation Tier / Client-Side*)

Lapisan ini adalah garda terdepan sistem—satu-satunya bagian yang berinteraksi langsung dengan panca indera pengguna (pasien atau tenaga kesehatan).

1. Fungsi: Mengumpulkan input data gejala medis dari pengguna melalui elemen-elemen antarmuka seperti dropdown, radio button, dan kolom input text. Selain itu, lapisan ini bertugas menampilkan vonis hasil prediksi (misal: Berisiko Tinggi Diabetes atau Risiko Rendah) dengan cara yang visual dan mudah dipahami.
2. Karakteristik dalam E-Health: Desain antarmuka pada lapisan presentasi kesehatan harus mengikuti prinsip *Usability* dan *Accessibility*. Pengguna mungkin adalah orang awam yang tidak paham bahasa pemrograman; oleh karena itu, form input harus menggunakan bahasa yang ramah (misalnya, menjelaskan Polyuria sebagai Sering buang air kecil).
3. Teknologi: Umumnya dibangun menggunakan kerangka HTML, CSS, dan perenderan antarmuka di peramban web (*browser*).

4.2.2. Lapisan Logika Bisnis dan Komputasi (*Logic Tier / Server-Side*)

Ini adalah otak dari keseluruhan arsitektur. Di sinilah letak kecerdasan buatan dan algoritma K-NN dieksekusi.

1. Fungsi Utama: Menerima data gejala yang dikirimkan oleh lapisan presentasi, melakukan validasi keamanan data, mengubah teks menjadi representasi numerik (0 dan 1), serta menormalisasi data rentang umur.
2. Eksekusi Algoritma: Pada lapisan inilah kalkulasi matematika yang berat (seperti menghitung akar kuadrat dari Euclidean Distance terhadap 520 data latih secara simultan) terjadi.
3. Keunggulan Kinerja: Dengan memusatkan komputasi berat di server (menggunakan engine PHP), sistem tidak akan membebani RAM atau CPU dari gawai (misal: smartphone atau laptop spesifikasi rendah) milik pengguna. Pengguna hanya perlu mengirimkan request dan menerima response hasil akhir yang ringan.

4.2.3. Lapisan Basis Data (*Data Tier / Database Server*)

Sebuah algoritma K-NN tipe *Lazy Learner* sangat bergantung pada kualitas dan ketersediaan data historis yang disimpan. Lapisan ini bertindak sebagai memori jangka panjang sistem.

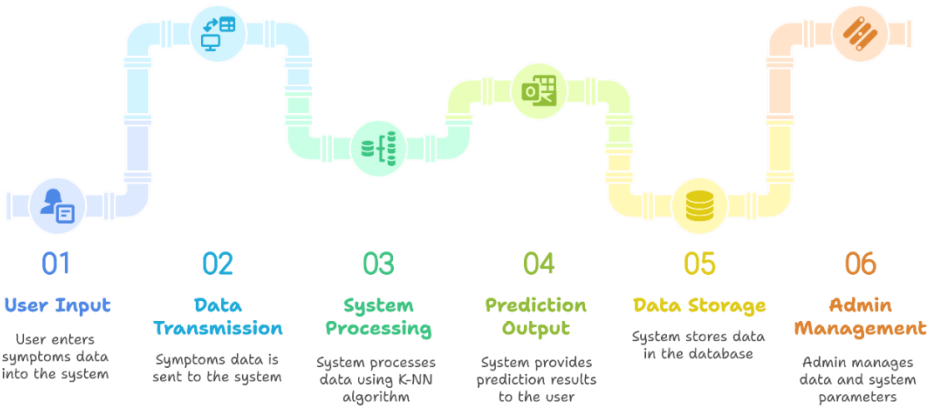
1. Fungsi Penyimpanan Statis: Menyimpan dataset latih (seperti 520 rekam jejak pasien dari Kaggle) yang menjadi referensi perhitungan.
2. Fungsi Penyimpanan Dinamis: Menyimpan log atau riwayat pengujian pasien baru yang menggunakan sistem web tersebut. Riwayat ini sangat berharga untuk audit medis di masa depan.
3. Teknologi: RDBMS (*Relational Database Management System*) seperti MySQL umumnya menjadi pilihan ideal karena kompatibilitasnya yang tanpa batas dengan PHP, serta kemampuannya menangani struktur tabel baris-kolom yang menjadi representasi murni dari dataset matriks.

4.3. Pemodelan Alur Kerja (Workflow) dan Aliran Data

Sebuah arsitektur yang baik memerlukan pemetaan alur kerja visual agar tim pengembang memiliki pemahaman sintaksis yang seragam mengenai ke mana data mengalir dari awal hingga akhir. Dalam rekayasa perangkat lunak, hal ini direpresentasikan melalui Context Diagram dan Flowchart proses algoritma.

4.3.1. Diagram Konteks (*Context Diagram*)

Context Diagram merepresentasikan pandangan level tertinggi (Level 0) dari sistem, menunjukkan batasan sistem (*system boundary*) serta interaksinya dengan entitas-entitas eksternal.



Gambar 4.3. Diagram Konteks Sistem Prediksi Risiko Diabetes Berbasis Web.

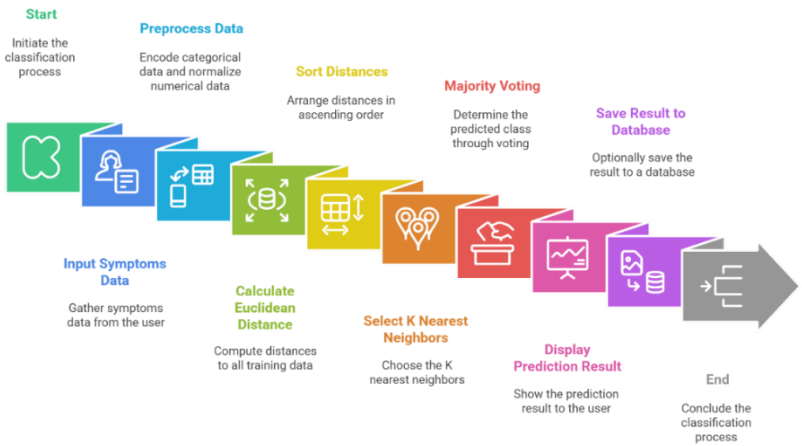
Secara makro, interaksi antara pengguna, administrator, dan proses utama dalam sistem klasifikasi ini dipetakan melalui diagram konteks yang disajikan pada Gambar 4.3 di atas.

Dalam sistem prediksi cerdas berbasis web, interaksi entitas dipetakan secara sistematis:

1. Entitas Pengguna (*User*): Bertindak sebagai sumber data dinamis. Pengguna menavigasi ke aplikasi web, berinteraksi dengan form (*User Input*), lalu memicu transmisi data gejala. Sistem membalas dengan memberikan informasi diagnostik awal (Output Klasifikasi) yang tampil di layar.
2. Entitas Sistem Prediksi (*Main Process*): Ini adalah aplikasi web utama (PHP) yang membungkus keseluruhan modul. Ia menerima keluhan pasien, menjalankan model komputasi K-NN di dapur pacunya, lalu memutuskan kelas prediksi.
3. Entitas Administrator (Opsional/Pengembangan): Bertugas melakukan manajemen pangkalan data, memperbarui nilai dataset latih jika terdapat riset medis terbaru, dan memelihara server web (VPS).

4.3.2. Dekonstruksi Flowchart Klasifikasi K-NN

Setelah memahami gambaran makro, kita harus melakukan *zoom-in* ke dalam jantung *Logic Tier* untuk melihat proses Data Flow (aliran data) pada tahap klasifikasi. Proses logika di dalam *logic tier* saat menangkap input gejala hingga menghasilkan keputusan diagnosis akhir melalui perhitungan K-NN dapat ditelusuri melalui alur kerja yang digambarkan pada Gambar 4.4.



Gambar 4.4. Flowchart Alur Kerja Algoritma *K-Nearest Neighbor* dalam Sistem Klasifikasi

Pemodelan alur algoritma K-NN digambarkan melalui tahapan linier berikut:

1. Inisiasi dan Penangkapan Input: Sistem menangkap array berisi keluhan pasien dari variabel POST atau GET web.
2. Proses Pra-pemrosesan (*Preprocessing Data*): Aliran tidak langsung dihitung. Data harus melewati gerbang konversi. Teks Laki-laki diubah menjadi 0 atau 1. Nilai umur murni (misal 45 tahun) di-binning ke rentang diskrit, lalu dimasukkan ke dalam rumus Min-Max Scaling agar tidak membiaskan perhitungan.
3. Iterasi Penghitungan Jarak: Sistem membuka gerbang koneksi ke dataset latih. Algoritma melakukan perulangan (*looping*) dari baris data ke-1 hingga baris ke-520. Pada setiap iterasi, nilai jarak Euclidean antara pasien uji dan pasien historis dihitung dan disimpan ke dalam struktur array sementara beserta index rujukannya.
4. Pengurutan Nilai (*Sorting Distances*): Array jarak tersebut kemudian diurutkan dari magnitudo terkecil (paling mirip) ke magnitudo terbesar (paling berbeda).
5. Pemotongan K-Nearest: Algoritma menghentikan pengurutan, dan hanya mengisolasi *K* data pada peringkat teratas (misalnya, mengambil 1, 3, atau 5 tetangga terdekat sesuai dengan parameter optmasi).
6. Pengambilan Keputusan (*Majority Voting*): Algoritma menengok ke kolom target (class label) dari *K* tetangga tersebut. Kelas (*Positive* atau *Negative*) yang mendominasi kelompok akan dideklarasikan sebagai pemenang.
7. Terminal Output: Hasil diserahkan kembali ke interface peramban, dan data sesi disimpan ke dalam pangkalan data MySQL.

4.4. Perancangan Skema Basis Data (MySQL)

Sistem pendukung keputusan medis bukanlah sistem komputasi sementara yang melupakan data seketika setelah tab peramban ditutup. Rekam jejak elektronik (*Electronic Health Records*) mewajibkan integritas penyimpanan permanen. Dalam kerangka kerja Three-Tier, basis data MySQL bertindak sebagai jangkar persistensi.

Penerjemahan atribut medis ke dalam desain Relational Database memerlukan strategi perancangan tabel yang presisi.

4.4.1. Filosofi Desain Tabel Berorientasi Atribut

Berbeda dengan sistem *e-commerce* yang memiliki relasi kompleks antar banyak tabel, perancangan database untuk klasifikasi berbasis gejala (seperti dataset prediksi diabetes 16 atribut) umumnya berfokus pada desain struktur tabel datar (*flat table structure*) yang sangat efisien dan optimal untuk proses kueri pembacaan cepat (*fast reading query*).

Setiap pasien yang tersimpan direpresentasikan sebagai satu tupel tunggal (satu baris data), di mana setiap fiturnya dialokasikan ke dalam satu kolom spesifik. Desain yang tepat akan mencegah timbulnya anomali data (seperti redundansi atau integritas referensial yang rusak).

4.4.2. Struktur Tabel Data Gejala Latih (*Training Data Storage*)

Dataset awal yang bersumber dari pangkalan data penelitian (seperti Kaggle) dapat disimpan secara statis dalam file CSV atau, untuk performa aplikasi web dinamis yang lebih tinggi, dimasukkan ke dalam tabel inti di basis data MySQL.

Struktur konseptual tabel ini terdiri dari:

1. Kolom Identitas (*Primary Key*): Berupa ID Pasien atau Nomor Rekam Medis (bertipe Integer dengan *Auto Increment*).
2. Kolom Demografi Dasar: Memuat atribut Age (bertipe Integer untuk rentang nilai asli, atau Float jika menyimpan nilai yang sudah ternormalisasi antara 0 hingga 1) dan tipe kelamin (Gender / Sex).
3. Kolom Fitur Simtomatik: Terdapat 14 kolom biner (bertipe TINYINT atau BOOLEAN) untuk menampung kondisi ya/tidak dari setiap gejala, antara lain: Polyuria, Polydipsia, Sudden Weight Loss, Weakness, Polyphagia, Genital Thrush, Visual Blurring, Itching, Irritability, Delayed Healing, Partial Paresis, Muscle Stiffness, Alopecia, dan Obesity.
4. Kolom Label Kelas (Class Attribute): Kolom esensial yang bertindak sebagai landasan kebenaran absolut (ground truth), di mana nilai 1 merepresentasikan status Positif Berisiko Diabetes dan 0 merepresentasikan Negatif.

Dengan menanamkan data latih ke dalam MySQL, pemutakhiran dataset (menambah data pasien baru di masa depan) dapat dilakukan melalui perintah antarmuka Dashboard Admin biasa, tanpa harus menyentuh ulang struktur kode (source code) aplikasi.

4.4.3. Tabel Riwayat Prediksi Pengguna (*Prediction Logs*)

Sebagai fungsi rekam medis elektronik (E-Health), aplikasi web yang baik harus mampu mengingat riwayat screening pengguna. Oleh karena itu, dirancang pula sebuah tabel riwayat yang saling berelasi.

Setiap kali modul klasifikasi K-NN selesai mengeksekusi perhitungan dan memvonis hasil, aplikasi (melalui metode PDO PHP) akan mengirimkan kueri INSERT ke dalam tabel logs. Tabel ini menampung tanggal/waktu eksekusi (timestamp), seluruh input 16 gejala dari pengguna saat itu, nilai probabilitas (confidence level jika diperlukan), dan hasil keputusan akhir algoritma.

Desain penyimpanan ini sangat krusial, karena di masa depan, kumpulan data di tabel logs ini dapat dimanfaatkan ulang (*re-purposed*) sebagai tambahan bahan pelatihan baru (memperbesar volume data latih), menciptakan ekosistem pembelajaran mesin yang berkelanjutan dan terus tumbuh (*continuous machine learning ecosystem*).

4.5. Ringkasan Bab

Bab ini telah menjembatani konsep teoretis dengan praktik rekayasa teknis. Penggunaan metodologi siklus hidup Waterfall memberikan jaminan bahwa pengembangan sistem kesehatan digital bergerak dalam koridor yang terencana dan presisi.

Dengan mengadopsi Arsitektur Tiga Tingkat (*Three-Tier Architecture*), kita berhasil memisahkan kompleksitas interaksi manusia di sisi frontend, kecanggihan otak kalkulasi K-NN di sisi backend, dan ketahanan penyimpanan data di MySQL. Pemodelan alur kerja melalui diagram visual secara esensial membuktikan bahwa kecerdasan buatan dapat diintegrasikan secara elegan ke dalam struktur aplikasi web yang konvensional sekalipun.

Setelah memahami cetak biru dan arsitektur fondasional ini, bab selanjutnya akan memasuki tahap yang paling menantang: penerjemahan desain ini ke dalam ranah kode komputasi. Kita akan membedah secara spesifik bagaimana lingkungan eksekusi PHP dikonfigurasi dan bagaimana pustaka ML (*Machine Learning Library*) dimanfaatkan untuk menghidupkan algoritma K-NN di server produksi.

BAB 5 INTEGRASI *MACHINE LEARNING* PADA LINGKUNGAN WEB (PHP)

5.1. Ekosistem Pengembangan Perangkat Lunak Modern

Setelah merumuskan arsitektur sistem dan memodelkan alur kerja secara konseptual pada bab sebelumnya, fase krusial berikutnya adalah mendaratkan konsep tersebut ke dalam ekosistem teknologi yang nyata. Pemilihan teknologi (*tech stack*) tidak boleh dilakukan secara serampangan; ia harus didasarkan pada kebutuhan skalabilitas, kemudahan pemeliharaan, dan kapabilitas lingkungan peladen (*server-side environment*) untuk menangani beban komputasi matematika.

Dalam perancangan sistem prediksi risiko kesehatan ini, ekosistem yang dibangun bertumpu pada bahasa pemrograman PHP versi modern (PHP 8.2) dan sistem manajemen basis data relasional MySQL 8.0, yang kemudian di- deploy (disebarkan) pada lingkungan *Virtual Private Server* (VPS).

5.1.1. Evolusi PHP 8.2 sebagai Mesin Komputasi Algoritma

Secara historis, PHP (PHP: *Hypertext Preprocessor*) diciptakan sebagai bahasa skrip yang disisipkan pada HTML untuk membuat halaman web dinamis yang sederhana. Namun, selama lebih dari dua dekade, PHP telah berevolusi menjadi bahasa pemrograman tingkat server yang sangat tangguh (*robust*).

Penggunaan PHP versi 8.2 dalam sistem klasifikasi *Machine Learning* bukanlah tanpa alasan. Versi modern PHP membawa perombakan arsitektur di bawah kap mesinnya (*under the hood*), khususnya dengan kehadiran *Just-In-Ti*me (JIT) *Compiler* yang diperkenalkan sejak versi 8.0. JIT mengubah paradigma bagaimana skrip PHP dieksekusi: alih-alih mengompilasi kode menjadi opcode secara *real-time* berulang kali pada setiap eksekusi, JIT menerjemahkan bagian-bagian kode yang sering dipanggil langsung menjadi instruksi bahasa mesin (*machine code*).

Implikasi dari fitur ini sangat masif bagi algoritma berbasis jarak seperti K-NN. Algoritma K-NN mengharuskan iterasi (perulangan) kalkulasi jarak Euclidean dari satu pasien uji terhadap ratusan data pasien historis secara simultan. Dengan mesin PHP modern, kalkulasi akar kuadrat dan pengurangan matriks yang intensif ini dapat dieksekusi dalam orde milidetik, menjadikan PHP platform yang memadai untuk implementasi kecerdasan buatan skala menengah, dan menghapuskan stigma lama bahwa PHP tidak cocok untuk *Data Science*.

5.1.2. Reliabilitas Basis Data MySQL 8.0

Basis data adalah fondasi persistensi sistem. MySQL versi 8.0 dipilih karena keandalannya yang telah teruji industri, dukungannya terhadap integritas transaksional (ACID *compliance*), dan optimasi mesin kueri (*query engine*).

Dalam konteks algoritma tipe *Lazy Learning* seperti K-NN, kecepatan akses data sangat krusial. Seperti yang telah dibahas, K-NN tidak membangun model statis di awal; ia harus menarik data latih (variabel gejala pasien historis) langsung dari pangkalan data atau media penyimpanan internal setiap kali ada permintaan prediksi baru. MySQL 8.0 menyediakan manajemen *indexing* yang superior, sehingga pencarian dan pengambilan set data pelatihan berkapasitas besar dapat dilakukan dengan latensi yang nyaris tidak terasa oleh pengguna akhir.

5.1.3. Lingkungan *Virtual Private Server* (VPS)

Implementasi algoritma AI yang responsif menuntut lingkungan hosting yang memiliki sumber daya terdedikasi, bukan sekadar *Shared Hosting* konvensional. Sistem dideploy pada *Virtual Private Server* (VPS) yang beroperasi di atas sistem operasi Linux (seperti Ubuntu atau CentOS).

VPS memberikan kendali penuh (*root access*) atas alokasi memori (RAM) dan penggunaan inti prosesor (CPU). Mengingat proses sorting atau pengurutan nilai jarak Euclidean akan memakan alokasi memori yang linear dengan jumlah data latih, arsitektur VPS memastikan bahwa proses komputasi *Machine Learning* di sisi backend tidak akan terhenti secara tiba-tiba akibat pembatasan eksekusi sumber daya (*resource limitation timeout*) yang sering terjadi pada server berbagi (*shared servers*). Konfigurasi web server seperti Nginx atau Apache bertindak sebagai reverse proxy yang andal untuk meneruskan permintaan pasien (melalui peramban) ke mesin pemroses PHP secara asinkron.

5.2. Pustaka PHP *Machine Learning* (PHP-ML)

Dalam ranah rekayasa perangkat lunak modern, ada sebuah prinsip fundamental yang berbunyi: "Don't reinvent the wheel" (Jangan menemukan kembali roda). Mengkodekan algoritma matematika dari awal (*from scratch*) tentu memiliki nilai edukasi, namun dalam skala produksi perangkat lunak profesional, praktik tersebut rawan akan bug, inefisiensi komputasi, dan celah keamanan.

Di sinilah konsep Pustaka (Library) masuk. Pustaka adalah kumpulan kode terkompilasi yang telah ditulis, diuji secara ketat, dan dioptimasi oleh komunitas pengembang global, yang dapat langsung digunakan untuk menjalankan tugas

spesifik. Dalam sistem prediksi kesehatan ini, kekuatan K-NN digerakkan oleh pustaka PHP-ML (PHP *Machine Learning*).

5.2.1. Rasionalisasi Pemilihan PHP-ML

Ekosistem bahasa pemrograman Python memiliki pustaka Scikit-Learn (sklearn) yang mendominasi industri AI. Lalu, mengapa mengembangkan sistem menggunakan PHP-ML?

Jawabannya terletak pada keringanan integrasi web (*lightweight web integration*). Membangun antarmuka pengguna berbasis web dengan HTML/CSS dan menghubungkannya dengan kecerdasan buatan berbasis Python biasanya memaksa pengembang membangun arsitektur API (*Application Programming Interface*) ganda. Sistem PHP harus memanggil sistem Python, lalu menunggu jawaban, yang berpotensi menimbulkan bottleneck (leher botol) aliran data.

Dengan mengadopsi PHP-ML, jembatan antara antarmuka web, koneksi basis data MySQL, dan mesin kecerdasan buatan dipersatukan ke dalam satu bahasa dan satu lingkungan eksekusi. Hal ini menyederhanakan arsitektur perangkat lunak secara drastis, mengurangi jumlah lapisan yang dapat mengalami kegagalan (*points of failure*), serta memungkinkan pemrosesan yang mulus dari penangkapan input pengguna (*form submission*) langsung menuju pemetaan ruang multidimensi K-NN.

5.2.2. Manajemen Dependensi dengan Composer

Pustaka PHP-ML tidak didistribusikan secara manual, melainkan dikelola secara elegan menggunakan Composer, yaitu manajer dependensi (*dependency manager*) standar de-facto untuk PHP. Composer memastikan bahwa sistem secara otomatis mengunduh pustaka utama PHP-ML beserta pustaka-pustaka turunan yang dibutuhkan untuk operasi matematis (seperti pustaka manipulasi matriks tingkat lanjut).

Penggunaan Composer (melalui perintah terminal **composer require php-ai/php-ml**) memastikan bahwa kode sumber (*source code*) yang berada di VPS akan selalu terikat pada versi PHP-ML yang stabil dan bebas dari konflik kode.

5.2.3. Anatomi Internal Pustaka PHP-ML

Pustaka PHP-ML dibangun dengan arsitektur Pemrograman Berorientasi Objek (*Object-Oriented Programming* / OOP) yang sangat ketat, mengikuti standar penulisan kode PHP global (PSR). Struktur internalnya diklasifikasikan berdasarkan domain kerja *Machine Learning*.

Bagi algoritma K-NN, kita berinteraksi langsung dengan klasifikasi namespace **Phpml\Classification\KNearestNeighbors**. Pustaka ini secara elegan telah memecah tanggung jawab fungsional menjadi beberapa interface kunci:

1. *Estimator Interface*: Interface utama yang mewajibkan seluruh algoritma klasifikasi untuk memiliki dua metode (method) esensial, yaitu metode **train()** untuk memuat data latih, dan metode **predict()** untuk mengevaluasi data baru.
2. *Distance Interface*: Antarmuka yang secara khusus menangani abstraksi kalkulasi metrik. Secara bawaan (default), objek K-NN di dalam PHP-ML akan menginjeksikan metode Euclidean(), yang mana rumusnya persis dengan apa yang telah dikonseptualisasikan pada Bab 3.

5.3. Pemodelan Klasifikasi K-NN di Sisi Backend

Sub-bab ini akan membahas bagaimana teori algoritma *K-Nearest Neighbor* diterjemahkan menjadi alur komputasi di backend server. Transformasi ini mengubah logika konseptual menjadi logika sintaksis.

5.3.1. Penyiapan Data Pelatihan (*Fase Training*)

Sesuai dengan sifat K-NN sebagai *Lazy Learner*, fase pelatihan pada pustaka PHP-ML sebenarnya merupakan proses penyusunan ulang memori (*memory allocation*).

Sistem akan memanggil variabel **$samples** yang merupakan matriks dua dimensi berisi seluruh fitur rekam medis (dari usia hingga kondisi obesitas pasien historis), beserta variabel **$labels** yang merupakan representasi diagnosis akhir (Positive atau Negative).

Sintaks secara operasional memanggil kelas utama:

```
$classifier = new KNearestNeighbors($k = 1);
$classifier->train($samples, $labels);
```

Dalam tahapan ini, parameter *K* yang merupakan hasil optimasi akurasi (*cross-validation*) diinjeksikan secara eksplisit. Pemilihan k = 1 memastikan bahwa pustaka ini akan diprogram untuk mengeksekusi Euclidean distance, lalu berhenti dan mengambil satu tetangga absolut terdekat untuk melakukan penarikan kesimpulan.

Perintah **train()** pada PHP-ML tidak melakukan pembobotan bobot (*weight tuning*) layaknya jaringan saraf tiruan (*neural network*), melainkan sekadar menempatkan seluruh array multidimensi tersebut ke dalam atribut objek (object property) di RAM, menjadikannya siap pakai sewaktu-waktu dibutuhkan.

5.3.2. Penangkapan dan Pemetaan Input Tak Dikenal (*Unknown Sample*)

Saat seorang pengguna memasukkan data melalui formulir web, data tersebut berwujud paket teks (String). Agar dapat dikonsumsi oleh **$classifier**, sistem backend harus melakukan pemetaan (mapping) menjadi array berdimensi tunggal yang struktur panjangnya identik dengan array data latih.

Sebagai contoh, 16 input dari form HTML akan dikonversi menggunakan logika bersyarat (if-else atau ternary operator). Jika pasien mencentang Polyuria sebagai Yes, maka nilainya di-*push* ke dalam array sebagai integer 1. Untuk atribut Age (Usia), nilai mentah (misalnya 45 tahun) secara otomatis akan diproses oleh sub-rutin normalisasi Min-Max sehingga nilainya menyusut menjadi angka proporsional di antara 0 dan 1 (seperti 0,375).

Hasil akhir dari proses ini adalah sebuah vektor (array linear):

$unknownSample = [0.375, 1, 1, 0, 1, 0 ...];

5.3.3. Eksekusi Prediksi (*Prediction Phase*)

Vektor input ini kemudian dilemparkan ke dalam fungsi pamungkas dari PHP-ML:

```
$hasilPrediksi = $classifier->predict($unknownSample);
```

Di balik satu baris kode **predict()** yang sederhana ini, PHP-ML mengeksekusi operasi matematika yang masif:

1. Melakukan perulangan otomatis terhadap ratusan baris data di variabel memori **$samples**.
2. Mengkalkulasi selisih jarak antara setiap elemen **$unknownSample** dengan elemen **$samples**.
3. Mengurutkan hasil dari yang berjarak 0.000 (sangat identik) hingga yang terjauh menggunakan algoritma sorting bawaan mesin PHP.
4. Mengekstrak satu label (karena *K*=1) yang terkait dengan elemen dengan jarak terendah tersebut.

5. Mengembalikan nilai 1 (Berisiko/Positif) atau 0 (Negatif/Aman).Logika internal inilah yang memberikan kecerdasan pada aplikasi web, menjadikannya bukan sekadar tempat penyimpanan data statis, melainkan mesin *decision-support* (pendukung keputusan).

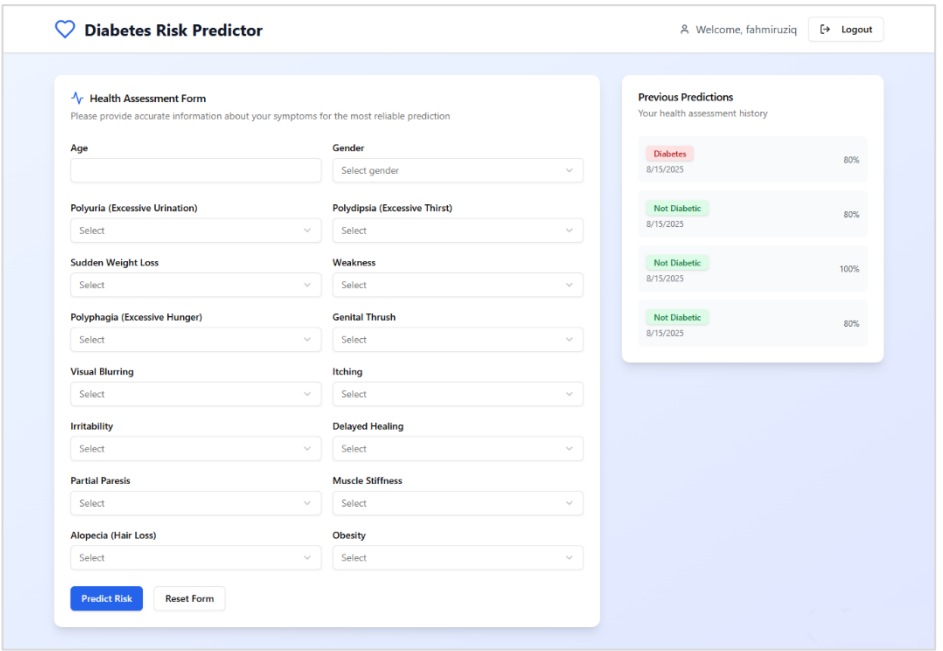
5.4. Pembangunan Antarmuka Pengguna (*User Interface*)

Sistem kecerdasan buatan paling canggih sekalipun tidak akan memiliki impact (dampak) sosial jika tidak dibungkus dengan antarmuka pengguna (*User Interface*/UI) yang komunikatif. Dalam domain kesehatan, perancangan antarmuka bukan hanya soal estetika, melainkan juga psikologi dan aksesibilitas.

5.4.1. Desain *Health Assessment Form*

Pintu gerbang sistem ini adalah Formulir Penilaian Kesehatan (*Health Assessment Form*). Berbeda dengan form administrasi biasa, form medis yang menanyakan 16 gejala simultan berisiko menimbulkan kelelahan kognitif (*cognitive load*) pada pengguna.

Untuk memudahkan interaksi pengguna dalam memasukkan data klinis secara mandiri, sistem menyediakan antarmuka formulir yang intuitif, sebagaimana ditunjukkan pada Gambar 5.1.



Gambar 5.1. Antarmuka Formulir Input Gejala Kesehatan pada Sistem Prediksi

Untuk mengatasi ini, antarmuka web dirancang dengan pendekatan Minimalist Card UI menggunakan framework antarmuka responsif (seperti Bootstrap atau Tailwind CSS).

Berdasarkan implementasi desain UI yang tergambar pada sistem:

1. Pengelompokan Grid: Input gejala tidak dijejer ke bawah secara tak terbatas, melainkan diatur dalam format kotak (grid layout), di mana setiap atribut medis memiliki blok dropdown atau kolom inputnya sendiri. Hal ini mempercepat pergerakan mata pengguna.
2. Pemahaman Terminologi: Pengguna awam seringkali teralienasi oleh istilah latin medis. Antarmuka yang baik menangani ini dengan membubuhkan padanan kata umum di samping istilah medis (misalnya: Polyuria (*Excessive Urination*) atau Polyphagia (*Excessive Hunger*)). Strategi UX (*User Experience*) ini memastikan validitas input, mencegah pasien memilih parameter yang salah murni karena ketidaktahuan linguistik.
3. Metode Transmisi Pengguna: Saat pengguna mengeklik tombol pemrosesan (*Predict Risk*), antarmuka akan memaketkan seluruh pilihan dropdown menggunakan metode HTTP POST. Metode POST memastikan bahwa paket data kesehatan dikirim secara tertutup di body request, bukan di URL bar, demi menjaga prinsip kerahasiaan (privacy) data medis dari pandangan publik.

5.4.2. Visualisasi Keputusan (*Output Presentation*)

Setelah Logic Tier menyelesaikan perhitungan K-NN melalui PHP-ML, antarmuka wajib merender hasil tersebut secara instan.

Tantangan psikologis dalam E-Health adalah bagaimana mengomunikasikan hasil kepada pasien tanpa menimbulkan diagnosis akhir yang menyesatkan (karena sistem ini berfungsi sebagai triage atau penapis awal, bukan pengganti diagnosis klinis mutlak).

Modul output mengandalkan tipografi visual:

1. Jika hasil klasifikasi mengembalikan nilai Positif, antarmuka akan menampilkan peringatan risiko tinggi menggunakan palet warna merah atau jingga (seperti yang ditunjukkan pada kolom *Previous Predictions* berupa label Diabetes merah).
2. Jika hasil mengembalikan nilai Negatif, sistem akan memunculkan lencana hijau bertuliskan Not Diabetic.

Selain itu, sisi sebelah antarmuka menyertakan panel khusus untuk menyajikan Riwayat Prediksi Sebelumnya (*Previous Predictions*). Panel ini melakukan pemanggilan (kueri) membaca tabel log riwayat dari MySQL. Fungsi ini esensial bagi pasien kronis untuk melihat trendline hasil pengecekan berkalanya seiring waktu. Jika pasien yang sebelumnya selalu masuk kategori *Not Diabetic* namun tiba-tiba hasil terbarunya berubah menjadi Diabetes (merah), maka antarmuka secara implisit memberikan teguran keras (*nudge*) agar pasien segera mengunjungi fasilitas kesehatan profesional untuk validasi darah komprehensif.

5.5. Ringkasan Bab

Bab ini telah mendemonstrasikan secara konkrit bagaimana konvergensi antara bahasa skrip *server-side* standar (PHP), pustaka *open-source* AI (PHP-ML), dan rekayasa antarmuka *frontend* dapat melahirkan sebuah instrumen medis preventif yang inovatif.

Eksekusi logika K-NN tidak lagi terkunci dalam kerumitan kalkulus murni, melainkan berhasil diabstraksikan ke dalam method pustaka yang elegan (*train* dan *predict*). Melalui desain antarmuka yang sensitif terhadap keterbatasan literasi pengguna, sistem berbasis web ini berhasil memenuhi tujuannya untuk mendemokratisasi akses menuju prediksi kesehatan yang sebelumnya eksklusif.

Implementasi teknis dan kode yang telah terpasang di atas infrastruktur VPS ini siap untuk dijalankan secara fungsional. Tantangan pamungkas dari pengembangan perangkat lunak ini adalah membuktikan secara matematis bahwa program yang dikonstruksi benar-benar mengeluarkan prediksi yang sahih dan kredibel. Oleh karena itu, pada bab selanjutnya, kita akan membedah secara saintifik metrik-metrik evaluasi model AI untuk membuktikan supremasi dan validitas dari algoritma yang telah di-*deploy* ini.

BAB 6 AUDIT KINERJA DAN MATRIKS VALIDASI ALGORITMA

6.1. Paradigma Pengujian Perangkat Lunak Medis

Membangun sebuah sistem cerdas—sebagaimana yang telah diuraikan pada tahap arsitektur dan implementasi PHP-ML di bab sebelumnya—hanyalah separuh dari siklus rekayasa perangkat lunak. Separuh lainnya yang tidak kalah esensial adalah fase audit dan validasi. Dalam domain informatika kesehatan (*Health Informatics*), merilis sebuah sistem diagnostik atau sistem triase ke publik tanpa melalui proses audit komputasional yang ketat adalah sebuah malapraktik teknologi.

Audit kinerja algoritma *Machine Learning* bertujuan untuk memberikan jaminan matematis bahwa model yang dibangun tidak hanya kebetulan benar pada saat dilatih, tetapi benar-benar memiliki kemampuan generalisasi yang kuat saat dihadapkan pada data pasien baru di dunia nyata. Evaluasi ini mencakup dua dimensi pengujian: pengujian fungsionalitas rekayasa perangkat lunak (*software functional testing*) dan evaluasi metrik kinerja model klasifikasi (*model performance metrics*) (Srinivas et al., 2022).

6.2. Skenario Validasi Fungsional (*Black-Box Testing*)

Sebelum menelaah akurasi matematis dari K-NN, lapisan terluar dari sistem—yaitu integrasi antara antarmuka web, pemrosesan PHP, dan basis data MySQL—harus divalidasi. Proses ini umumnya dilakukan menggunakan pendekatan *Black-Box Testing* (Pengujian Kotak Hitam) (Aghababaeyan et al., 2023).

Dalam Black-Box Testing, auditor atau penguji tidak perlu melihat ke dalam struktur kode (algoritma internal PHP). Fokus pengujian sepenuhnya bertumpu pada kesesuaian antara input yang diberikan dengan output yang dihasilkan oleh antarmuka sistem.

6.2.1. Validasi Transmisi Input Gejala

Skenario pengujian fungsional dimulai dengan menyimulasikan berbagai kombinasi pengisian formulir keluhan kesehatan oleh pasien (misalnya mengisi kombinasi Polyuria = Yes, Age = 45, dan Obesity = No) melalui *User Interface* web.

Tujuan pengujian ini adalah untuk memastikan bahwa:

1. Integritas Form: Tidak ada data yang hilang (*null*) saat paket formulir dikirimkan dari peramban ke server menggunakan metode HTTP POST.

2. Pemetaan Kategori (*Encoding*): Memastikan fungsi logika bersyarat di backend berhasil mengubah parameter teks manusia (Yes/No atau Male/Female) menjadi array biner numerik (1 dan 0) secara absolut tanpa error.
3. Kestabilan Normalisasi: Memastikan nilai numerik kontinu seperti usia berhasil direduksi oleh fungsi Min-Max Scaling agar tidak memicu kegagalan perhitungan di pustaka PHP-ML.

6.2.2. Validasi Keluaran dan Stabilitas Basis Data

Setelah input divalidasi, pengujian dilanjutkan ke tahap observasi hasil prediksi. Sistem yang stabil harus mampu memproses input tersebut dan mengembalikan nilai prediksi (Risiko Tinggi atau Risiko Rendah) dalam waktu kurang dari satu detik, tanpa memunculkan pesan *Fatal Error* atau *Timeout* pada server VPS.

Selain itu, skenario pengujian juga mencakup validasi aliran data ke MySQL. Penguji harus memastikan bahwa setiap sesi prediksi pasien secara otomatis tersimpan dan terekam di dalam tabel riwayat (*Prediction Logs*), dan dapat ditarik kembali untuk ditampilkan pada dashboard antarmuka. Kesuksesan pada tahap ini mengonfirmasi bahwa arsitektur operasional sistem siap pakai.

6.3. Optimasi Parameter *K* melalui *K-Fold Cross-Validation*

Setelah sistem secara fungsional dinyatakan stabil, audit bergeser ke level sains data. Tantangan terbesar dalam algoritma K-NN adalah sifatnya yang sangat bergantung pada nilai parameter *K* (jumlah tetangga terdekat). Pemilihan *K* secara acak sangat tidak saintifik. Oleh karena itu, pengembang sistem AI menggunakan metode *Cross-Validation* (Validasi Silang) untuk mencari nilai *K* yang paling optimal secara empiris.

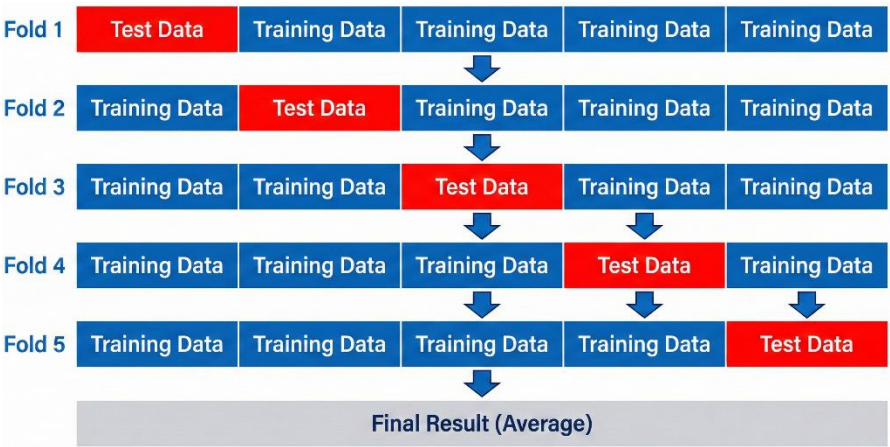
6.3.1. Teori *Cross-Validation*

Dalam praktik tradisional, sebuah dataset (misalnya 520 baris data pasien) biasanya dibagi menjadi dua bagian statis: 80% data latih dan 20% data uji. Namun, pendekatan pembagian statis ini rentan terhadap bias; akurasi yang dihasilkan mungkin sangat tinggi hanya karena kebetulan data uji yang terpilih sangat mudah diprediksi.

Untuk mengatasi bias tersebut, metode *K-Fold Cross Validation* membagi keseluruhan dataset ke dalam jumlah lipatan (*fold*) yang sama besar. Jika kita menggunakan *5-Fold Cross-Validation*, dataset dibagi menjadi 5 blok. Algoritma akan

diuji sebanyak 5 kali putaran (iterasi). Pada putaran pertama, blok 1 menjadi data uji dan blok 2-5 menjadi data latih. Pada putaran kedua, blok 2 menjadi data uji, dan sisanya menjadi latih. Proses ini berputar hingga seluruh baris data pernah merasakan posisi sebagai data uji. Akurasi akhir adalah nilai rata-rata dari kelima putaran tersebut.

Untuk memberikan gambaran yang lebih konkret mengenai bagaimana dataset dibagi menjadi beberapa lipatan pengujian dan pelatihan secara bergantian, mekanisme dari proses *5-Fold Cross-Validation* ini diilustrasikan secara visual pada Gambar 6.1.



Gambar 6.1. Ilustrasi Pembagian Data pada *5-Fold Cross-Validation*

6.3.2. Simulasi dan Analisis Eksperimental Nilai *K*

Sebagai ilustrasi komputasi dan studi kasus validasi, mari kita bedah skenario pengujian nilai *K* pada *Early Stage Diabetes Risk Prediction Dataset*. Algoritma diuji dengan tiga variasi parameter, yakni *K*=1, *K*=2, dan *K*=3, menggunakan skema 5-Fold Cross Validation.

Berdasarkan simulasi perhitungan pada dataset tersebut, matriks akurasi untuk setiap lipatan (fold) dapat direpresentasikan ke dalam tabel teoritis berikut:

Tabel 6.1. Matriks Evaluasi Akurasi berdasarkan Fold (Validasi Silang)

Putaran (Fold)	Akurasi untuk *K*=1	Akurasi untuk *K*=2	Akurasi untuk *K*=3
Fold 1	84,62%	78,85%	78,85%
Fold 2	94,23%	93,27%	86,54%
Fold 3	98,08%	95,19%	95,19%

Fold 4	99,04%	95,19%	93,27%
Fold 5	91,35%	89,42%	88,46%
Rata-Rata Final	93,46%	90,38%	88,46%

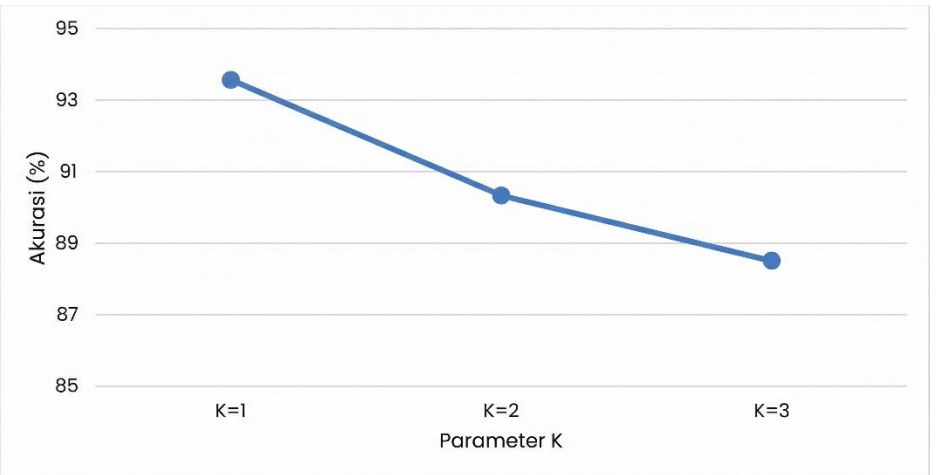
6.3.3. Justifikasi Pemilihan *K*=1

Dari hasil tabulasi di atas, terlihat fenomena yang menarik: seiring dengan penambahan jumlah tetangga (dari 1 ke 3), tingkat akurasi justru mengalami tren penurunan (dari 93,46% merosot menjadi 88,46%).

Secara teoritis, hal ini mengindikasikan bahwa batas keputusan (*decision boundary*) dari dataset gejala diabetes tahap awal ini memiliki struktur klasterisasi lokal yang sangat spesifik dan rapat. Menggunakan *K*=1 berarti algoritma K-NN mengambil satu referensi tetangga yang paling identik secara mutlak. Dalam konteks medis, ini bermakna bahwa pasien dengan kombinasi 16 gejala yang persis sama nyaris dapat dipastikan menderita kondisi yang sama.

Jika nilai *K* diperbesar menjadi 3, algoritma terpaksa menarik referensi dari pasien-pasien lain yang jarak Euclidean-nya sedikit lebih jauh. Penarikan referensi tambahan ini justru memasukkan noise (derau) dan mencemari hasil pemungutan suara mayoritas (*majority voting*), sehingga menyebabkan akurasi menurun. Oleh karena itu, pengembang mengunci nilai parameter *K*=1 sebagai model final untuk diimplementasikan ke lingkungan web.

Tren penurunan performa algoritma yang terjadi secara bertahap akibat penambahan jumlah tetangga terdekat (nilai parameter *K*) dapat dilihat secara lebih jelas pada visualisasi grafik di Gambar 6.2. berikut.



Gambar 6.2. Grafik Perbandingan Akurasi berdasarkan Nilai *K*

6.4. Matriks Kebingungan (*Confusion Matrix*) dan Metrik Lanjutan

Akurasi sebesar 93,46% adalah angka yang impresif. Namun, dalam disiplin ilmu informatika kesehatan, mengandalkan metrik Akurasi saja dianggap sebagai langkah yang dangkal dan berisiko. Jika kita memiliki data di mana 90 pasien sehat dan 10 pasien diabetes, sebuah sistem rusak yang selalu menebak Sehat tanpa berpikir sama sekali akan mendapatkan akurasi 90%. Ini disebut paradoks *Accuracy Paradox*.

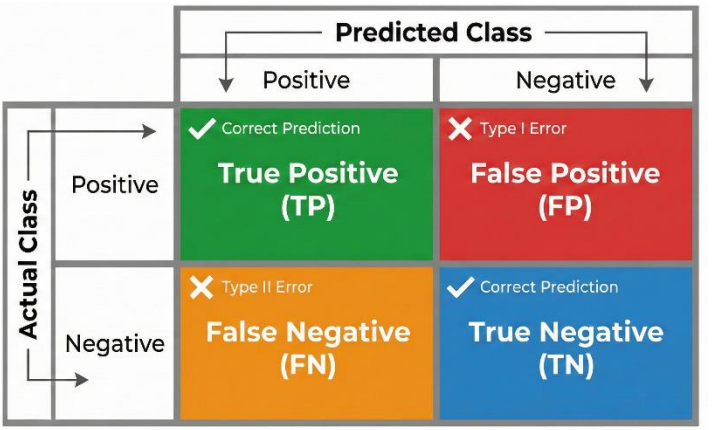
Untuk membedah kualitas kecerdasan buatan sesungguhnya, audit model medis menggunakan instrumen yang disebut Matriks Kebingungan (*Confusion Matrix*).

6.4.1. Anatomi *Confusion Matrix*

Matriks kebingungan adalah tabel kontingensi 2x2 yang memetakan prediksi sistem versus realita (kebenaran aktual). Matriks ini membagi hasil klasifikasi ke dalam empat sel logika:

1. True Positive (TP): Sistem memprediksi pasien berisiko Diabetes, dan faktanya memang pasien tersebut didiagnosis Diabetes oleh medis. (Prediksi benar).
2. True Negative (TN): Sistem memprediksi pasien Tidak Berisiko, dan faktanya pasien memang sehat. (Prediksi benar).
3. False Positive (FP - Kesalahan Tipe I): Sistem memprediksi pasien berisiko Diabetes, padahal faktanya pasien sehat. (Alarm palsu / *False Alarm*).
4. False Negative (FN - Kesalahan Tipe II): Sistem memprediksi pasien Tidak Berisiko, padahal faktanya pasien tersebut menderita Diabetes.

Keempat komponen evaluasi tersebut kemudian disusun ke dalam sebuah matriks dua dimensi. Struktur dasar dan anatomi dari matriks kebingungan (*confusion matrix*) ini dapat dilihat pada Gambar 6.3.



Gambar 6.3. *Template Confusion Matrix*

Dalam ranah kesehatan, False Negative adalah kesalahan yang paling fatal dan harus ditekan sekecil mungkin. Lebih baik sistem memberikan alarm palsu (pasien periksa ke dokter dan ternyata sehat), daripada sistem mengatakan pasien aman padahal ia memiliki diabetes tersembunyi yang terus merusak organ tubuhnya tanpa penanganan.

6.4.2. Derivasi Metrik Lanjutan: Presisi, Sensitivitas, dan F1-Score

Dari keempat nilai kuadran (TP, TN, FP, FN) di atas, ilmu statistik menurunkan rumusan-rumusan tingkat lanjut untuk menilai keandalan model K-NN:

1. Akurasi (*Accuracy*)
   Mengukur rasio total prediksi yang benar secara keseluruhan.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

   Sebagaimana ditunjukkan pada studi kasus optimasi K=1, sistem K-NN mampu mencapai akurasi hingga 93,46%, yang mengartikan bahwa 93 dari 100 pasien akan diprediksi statusnya dengan tepat.
2. Sensitivitas atau *Recall* (*True Positive Rate*)
   Metrik paling krusial di dunia medis. Recall mengukur proporsi pasien yang benar-benar sakit yang berhasil ditangkap (dideteksi) oleh sistem.

$$Recall = \frac{TP}{TP + FN}$$

Sebuah model AI klinis harus memiliki skor Recall yang sangat tinggi, sebagai jaminan bahwa angka False Negative berada pada batas toleransi minimal.

3. Presisi (*Precision*)
   Mengukur rasio dari total tebakan Positif oleh sistem yang benar-benar akurat. Presisi menjawab pertanyaan: "Dari semua orang yang dituduh diabetes oleh komputer, berapa banyak yang benar-benar sakit?"
   $$Precision = \frac{TP}{TP + FP}$$
4. *F1-Score*
   Merupakan nilai rata-rata harmonik (*harmonic mean*) antara Presisi dan Recall. *F1-Score* digunakan ketika kita ingin mencari keseimbangan antara mendeteksi semua pasien sakit (Recall) dan meminimalisir alarm palsu (Presisi), terutama jika distribusi kelas data pasien tidak seimbang (*imbalanced dataset*).
   $$F1 - Score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

6.5. Komparasi Kinerja K-NN dengan Literatur State-of-the-Art

Audit keilmuan sebuah algoritma tidak lengkap tanpa membandingkannya dengan metode lain yang beredar di literatur state-of-the-art (SOTA). Dalam berbagai penelitian klasifikasi penyakit kronis modern, perdebatan ilmiah sering kali mempertandingkan K-NN melawan algoritma Ensemble kompleks seperti *Random Forest* (RF) atau *Support Vector Machines* (SVM).

6.5.1. Analisis Trade-Off (Pertukaran Nilai) Akurasi vs Kompleksitas

Literatur sering kali mencatat bahwa Random Forest dan metode Gradient Boosting seperti XGBoost atau LightGBM dapat menembus ambang akurasi klasifikasi hingga 96% ke atas dalam dataset medis. Jika algoritma lain lebih superior, lantas apa justifikasi penggunaan K-NN (dengan akurasi 93,46%) sebagai fondasi sistem prediksi kesehatan berbasis web ini?

Jawabannya terletak pada analisis trade-off komputasional dan urgensi interpretability (keterjelasan).

Random Forest adalah algoritma yang membangun ratusan hingga ribuan pohon keputusan abstrak, menghasilkan model file statis yang berukuran sangat besar (bisa mencapai puluhan megabyte). Ketika diimplementasikan pada skrip web

interaktif seperti PHP, mengeksekusi struktur ribuan pohon keputusan secara seketika akan membebani RAM server secara masif, dan berpotensi memperlambat respon triage pada frontend.

6.5.2. Rasionalisasi Pemilihan Algoritma untuk *E-Health Web*

K-NN dipilih dan dievaluasi tinggi bukan semata karena ia berupaya menjadi algoritma yang paling akurat sedunia, melainkan karena ia menawarkan keseimbangan ekosistem yang paling sempurna untuk implementasi praktis berbasis PHP-ML. Keunggulan kompetitif K-NN dalam studi kasus ini meliputi:

1. Arsitektur Ringan (*Lightweight*): K-NN berbasis PHP-ML tidak memerlukan library pihak ketiga tambahan yang berat. Ia hanya menuntut operasi pengurangan matriks dan algoritma sorting dasar, membuatnya sangat adaptif berjalan di shared hosting maupun VPS standar dengan konsumsi daya minimal.
2. Kinerja yang Sangat Kompetitif: Memperoleh akurasi rata-rata 93,46% pada dataset kesehatan dunia nyata hanya bermodalkan kalkulasi Euclidean Distance sederhana adalah pembuktian bahwa dengan tahapan normalisasi data yang benar (Min-Max Scaling), algoritma tua dan sederhana mampu bersaing di liga algoritma modern tingkat tinggi.
3. Transparansi Etis Medis: Seperti yang dikupas di bab awal mengenai regulasi Black-Box di bidang medis, K-NN memungkinkan dokter pengawas untuk mengaudit secara harfiah "mengapa pasien A didiagnosis Positif". Dokter cukup membuka database, melihat 1 tetangga terdekat dari pasien tersebut (K=1), dan memvalidasi rekam medis historis tetangganya. Hal ini tidak bisa dilakukan pada Random Forest atau Neural Network.

6.6. Ringkasan Bab

Bab ini telah mendemonstrasikan bahwa sistem cerdas yang dikembangkan tidak hanya kokoh secara struktur pemrograman, namun juga sahih secara matematis. Melalui penerapan *Black-Box Testing*, alur transmisi formulir kesehatan pasien menuju mesin PHP-ML telah terbukti berfungsi tanpa cacat.

Di ranah komputasi data, pengujian empiris menggunakan metodologi ketat *K-Fold Cross-Validation* membuktikan bahwa penetapan parameter K=1 menghasilkan rentang prediksi yang sangat presisi dengan capaian rata-rata 93,46% pada matriks dataset. Diperkuat dengan instrumen Confusion Matrix, kita

memahami bagaimana K-NN meminimalisasi False Negative yang krusial dalam deteksi penyakit.

Komparasi kritis dengan literatur SOTA menyimpulkan bahwa K-NN berdiri tegak sebagai metode yang sangat relevan. Ia berhasil mendobrak mitos bahwa Kecerdasan Buatan harus senantiasa rumit dan membebani peladen (server). K-NN yang disandingkan dengan bahasa pemograman dinamis (PHP) memberikan solusi paripurna yang memadukan keandalan klinis, kecepatan web, dan transparansi keputusan.

Berbekal seluruh bukti saintifik, performa, dan implementasi yang diurai dari Bab 1 hingga Bab 6, buku ini akan ditutup dengan menyusun refleksi akhir mengenai implikasi teknologi ini di tengah masyarakat, serta merancang peta jalan inovasi bagi para periset *Health Informatics* di masa depan.

BAB 7 MASA DEPAN E-HEALTH DAN SKALABILITAS SISTEM

7.1. Sintesis Pembelajaran dan Refleksi Teknologi

Perjalanan panjang melintasi bab-bab dalam buku ini telah membawa kita pada satu pemahaman fundamental: demokratisasi layanan kesehatan melalui teknologi adalah sebuah keniscayaan. Kita telah memulai diskusi dari urgensi krisis kesehatan global akibat Penyakit Tidak Menular (seperti diabetes), membedah anatomi data medis yang kompleks, hingga menyelami kedalaman matematika dari algoritma *K-Nearest Neighbor* (K-NN). Lebih jauh lagi, kita telah mendemonstrasikan bagaimana teori-teori abstrak tersebut dapat dibumikan menjadi sebuah arsitektur perangkat lunak berbasis web menggunakan PHP dan pustaka PHP-ML.

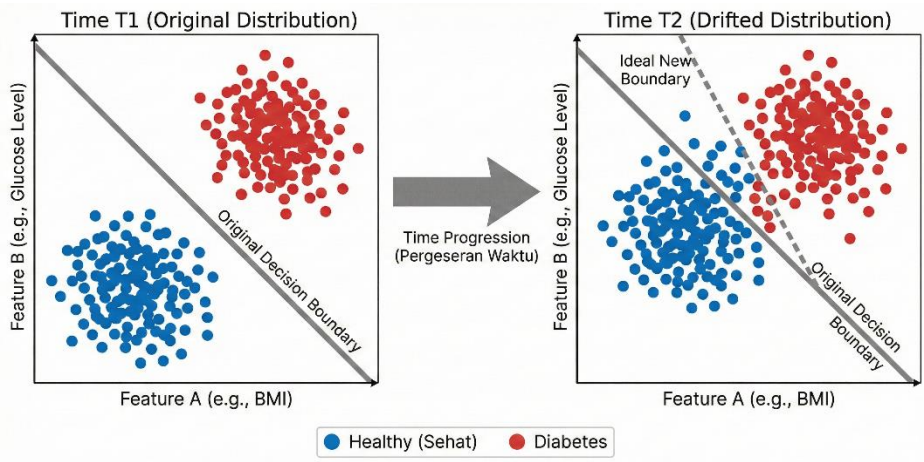
Sintesis utama dari seluruh pemaparan ini adalah bahwa Kecerdasan Buatan (*Artificial Intelligence*) dalam domain medis tidak selalu menuntut infrastruktur superkomputer yang eksklusif dan mahal. Dengan pemahaman yang presisi terhadap karakteristik data (melalui pra-pemrosesan dan normalisasi *Min-Max Scaling*) serta audit kinerja parameter yang teliti (seperti penetapan nilai K=1 melalui *Cross-Validation*), sebuah bahasa skrip tingkat server konvensional seperti PHP mampu disulap menjadi mesin *decision-support system* (sistem pendukung keputusan) yang memiliki tingkat akurasi di atas 93%.

Namun demikian, di dalam dunia rekayasa perangkat lunak dan sains data, penyelesaian sebuah sistem bukanlah akhir dari sebuah proses, melainkan titik awal dari evolusi berikutnya. Model yang telah di-deploy ke dalam lingkungan web saat ini beroperasi pada batasan-batasan tertentu yang harus diakui secara akademis dan strategis.

7.2. Keterbatasan Ekosistem Statis dan Fenomena Concept Drift

Salah satu karakteristik utama dari implementasi purwarupa (seperti penggunaan dataset publik berisi 520 rekam medis historis) adalah sifatnya yang tertutup dan statis. Dalam lingkungan akademik atau pengujian terkontrol, dataset statis sangat ideal untuk membuktikan validitas rumus matematika. Namun, ketika sistem ini dihadapkan pada ekosistem medis di dunia nyata, data bersifat dinamis, organik, dan terus berubah.

Fenomena di mana pola hubungan antara data gejala pasien dan label diagnosis mengalami pergeseran seiring berjalannya waktu—yang dikenal dengan istilah *Concept Drift*—diilustrasikan dampaknya pada Gambar 7.1.



Gambar 7.1. Ilustrasi *Concept Drift*

7.2.1. Ancaman Concept Drift pada K-NN

Dalam terminologi *Machine Learning*, terdapat sebuah anomali yang dikenal sebagai *Concept Drift* (Pergeseran Konsep). *Concept Drift* terjadi ketika properti statistik dari variabel target yang diprediksi oleh model berubah seiring berjalannya waktu dengan cara yang tidak terduga.

Sebagai contoh, pola gejala diabetes pada dekade 1990-an mungkin memiliki bobot korelasi yang sedikit berbeda dibandingkan dengan pasien di era 2020-an, di mana gaya hidup sedentari, jenis makanan olahan, dan tingkat stres telah mengalami transformasi masif. Jika algoritma K-NN hanya berpatokan pada 520 data latih statis dari tahun tertentu, maka seiring berjalannya waktu, tetangga terdekat yang dijadikan referensi oleh algoritma akan menjadi usang (*obsolete*). Pasien dengan keluhan modern mungkin dipetakan ke tetangga historis yang konteks klinisnya sudah tidak lagi relevan, yang pada akhirnya akan menyebabkan degradasi (penurunan) akurasi prediksi secara perlahan.

7.2.2. Kebutuhan akan *Continuous Learning*

Sifat K-NN sebagai algoritma *Lazy Learner* sebenarnya memberikan keuntungan taktis untuk mengatasi *Concept Drift*. Berbeda dengan *Neural Networks*

yang harus dilatih ulang secara total berhari-hari ketika ada pergeseran data, K-NN dapat beradaptasi secara instan hanya dengan memutakhirkan pangkalan datanya.

Oleh karena itu, sistem *E-Health* yang matang harus menerapkan paradigma *Continuous Learning* (Pembelajaran Berkelanjutan). Log riwayat prediksi dari pasien-pasien baru yang menggunakan aplikasi web ini (yang disimpan di dalam MySQL seperti yang dibahas pada Bab 4), setelah divalidasi kebenarannya oleh tenaga medis profesional, harus dilebur dan diintegrasikan kembali ke dalam matriks data latih **($samples** dan **$labels**). Dengan demikian, algoritma K-NN akan selalu memiliki tetangga yang *up-to-date* dan relevan dengan kondisi demografi pasien masa kini.

7.3. Skalabilitas Arsitektur Menuju Big Data

Tantangan kedua yang menanti sistem klasifikasi web di masa depan adalah lonjakan volume data. Algoritma K-NN menghitung jarak Euclidean dari data uji terhadap seluruh data latih yang ada di memori. Jika sistem saat ini menghitung jarak terhadap 520 baris data, proses tersebut dapat diselesaikan dalam hitungan milidetik oleh mesin PHP.

Namun, bagaimana jika sistem ini diadopsi secara nasional oleh Kementerian Kesehatan, dan pangkalan data tumbuh dari 520 baris menjadi 5.000.000 baris rekam medis?

Kalkulasi Euclidean distance terhadap jutaan titik koordinat pada setiap kali seorang pasien menekan tombol Prediksi akan memicu kemacetan prosesor (CPU bottleneck), menghabiskan kapasitas RAM peladen, dan menyebabkan halaman web mengalami Timeout (gagal muat).

Untuk menghadapi era Big Data medis, arsitektur perangkat lunak harus berevolusi. Berikut adalah beberapa skenario skalabilitas yang dapat diimplementasikan di masa depan:

7.3.1. Skalabilitas Vertikal dan Horizontal

Skalabilitas Vertikal (*Scaling Up*): Solusi paling primitif adalah dengan meningkatkan spesifikasi *Virtual Private Server* (VPS). Mengganti prosesor dengan clock speed yang lebih tinggi atau melipatgandakan kapasitas RAM agar skrip PHP memiliki ruang memori yang cukup untuk memuat jutaan array. Namun, solusi ini memiliki batasan fisik dan beban biaya yang eksponensial.

Skalabilitas Horizontal (*Scaling Out*): Solusi modern melibatkan penambahan jumlah peladen (server). Aplikasi web dan beban komputasi K-NN dipecah dan didistribusikan ke beberapa server berbeda menggunakan Load

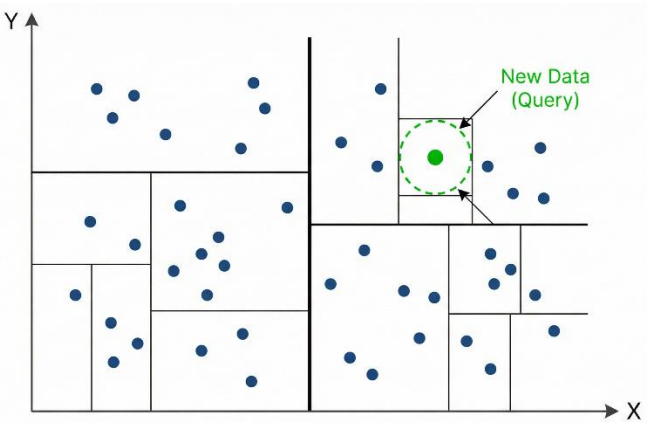
Balancer. Ketika ada 1.000 pasien yang mengakses web secara bersamaan, Load Balancer akan membagi rata permintaan lalu lintas tersebut ke server-server yang menganggur.

7.3.2. Optimasi Pencarian Tetangga dengan Struktur Data Tree

Dalam tingkat komputasi lanjutan, pencarian K-NN yang membandingkan jarak satu-per-satu (*Brute-force search*) harus ditinggalkan. Sistem dapat dioptimasi dengan menerapkan struktur data spasial yang lebih canggih, seperti K-D Tree (*K-Dimensional Tree*) atau *Ball Tree*.

Struktur data ini mempartisi ruang multidimensi ke dalam blok-blok area. Ketika data pasien baru masuk, algoritma tidak perlu lagi menghitung jarak ke jutaan pasien secara membabi-buta, melainkan hanya menghitung jarak terhadap pasien-pasien historis yang berada di dalam blok partisi yang sama. Logika ini mampu memangkas waktu komputasi dari orde linear $O(N)$ menjadi orde logaritmik $O(log\ N)$, memastikan bahwa web tetap merespons dalam sekejap mata meskipun menangani lautan Big Data.

Sebagai solusi untuk mereduksi beban komputasi yang tinggi pada K-NN konvensional, teknik pemisahan ruang pencarian menggunakan struktur K-D Tree dapat diterapkan. Mekanisme pelokalisasian pencarian tetangga ini ditunjukkan pada Gambar 7.2.



Gambar 7.2. Ilustrasi Pemisahan Ruang pada Struktur K-D Tree

7.3.3. Dekopling Arsitektur melalui Microservices

Jika beban komputasi K-NN menjadi terlalu berat bagi kerangka kerja PHP konvensional, arsitektur Three-Tier dapat didekonstruksi menjadi arsitektur Microservices. Modul antarmuka (UI/UX) tetap dipertahankan menggunakan

ekosistem web yang ringan, namun modul perhitungan K-NN dipisahkan menjadi layanan mandiri (*standalone service*) yang ditulis menggunakan bahasa tingkat rendah yang dikompilasi (seperti C++, Go, atau Rust) atau infrastruktur khusus Python. Aplikasi web PHP kemudian hanya bertugas mengirimkan data gejala via *Application Programming Interface* (API) ke layanan tersebut, dan menerima hasil prediksinya secara asinkron.

7.4. Konvergensi Teknologi: Integrasi *Internet of Medical Things* (IoMT)

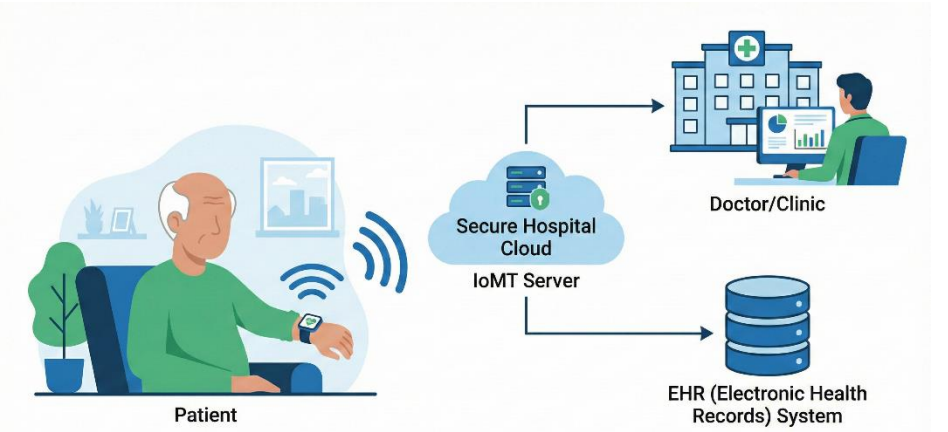
Kebergantungan sistem saat ini pada input manual (di mana pasien harus membaca form di web dan memilih dropdown Ya atau Tidak) masih memiliki celah subjektivitas. Pasien mungkin salah menafsirkan gejalanya sendiri, atau berbohong saat mengisi formulir.

Masa depan sistem prediksi *E-Health* terletak pada kemampuannya untuk mengeliminasi input manual dan beralih pada objektivitas mesin. Hal ini dicapai melalui konvergensi antara *Machine Learning* dan *Internet of Medical Things* (IoMT).

Perangkat wearable cerdas (seperti jam tangan pintar, sensor glukosa transdermal, atau bahkan sensor urinalisis pada kloset pintar di masa depan) akan terus-menerus memantau parameter biometrik pengguna secara *real-time*. Data seperti fluktuasi detak jantung, pola tidur (indikator kelelahan/kelemahan), hingga frekuensi buang air kecil (Polyuria) akan dikirim secara nirkabel langsung ke antarmuka pemrograman web (API).

Dalam skenario utopis ini, sistem algoritma K-NN yang kita bahas dalam buku ini tidak lagi menunggu pengguna membuka peramban web. Algoritma akan berjalan secara otomatis di latar belakang (*background job*). Ketika sensor mendeteksi kombinasi anomali yang mendekati klaster pasien Positif Diabetes di ruang Euclidean, sistem akan secara otonom mengirimkan notifikasi peringatan dini ke gawai pengguna atau langsung menjadwalkan konsultasi telemedis dengan dokter rujukan.

Untuk mencapai pemantauan klinis yang berkelanjutan, sistem prediksi ini di masa depan dapat diintegrasikan dengan perangkat wearable pasien. Konsep arsitektur aliran data *Internet of Medical Things* (IoMT) tersebut disajikan pada Gambar 7.3.



Gambar 7.3. Konsep *Internet of Medical Things* (IoMT)

7.5. Data Landasan Etika Lanjutan dan Kedaulatan

Semakin otonom dan terintegrasinya sebuah sistem E-Health, semakin besar pula tanggung jawab etis dan yuridis yang membebaninya. Pada Bab 1, kita telah menyinggung prinsip dasar kerahasiaan data. Namun, di masa depan, sistem prediksi berskala besar akan dituntut untuk patuh pada regulasi perlindungan data tingkat tinggi, seperti *General Data Protection Regulation* (GDPR) di ranah global, atau Undang-Undang Pelindungan Data Pribadi (UU PDP) di Indonesia.

Beberapa agenda krusial yang harus disiapkan oleh para perancang arsitektur perangkat lunak kesehatan masa depan meliputi:

1. Hak untuk Dilupakan (*Right to be Forgotten*): Basis data sistem tidak boleh bersifat mengikat permanen. Jika seorang pengguna mencabut persetujuannya (*consent*), sistem harus memiliki mekanisme algoritma yang mampu menghapus titik koordinat pasien tersebut dari memori K-NN tanpa merusak integritas perhitungan data lainnya.
2. Auditabilitas Bias Demografis: Perlu ada dewan pengawas independen yang secara rutin menguji dataset sistem. K-NN sangat bergantung pada demografi tetangganya. Jika pangkalan data didominasi oleh genetik ras tertentu atau kelas ekonomi menengah ke atas, prediksi sistem mungkin menjadi cacat dan diskriminatif ketika digunakan oleh populasi dari latar belakang genetik atau sosio-ekonomi yang berbeda. Model kecerdasan buatan medis harus inklusif dan berkeadilan.
3. Batas Tanggung Jawab Medis (*Liability*): Harus ada batas demarkasi yang jelas antara saran algoritma dan diagnosis medis. Syarat dan Ketentuan

(*Terms of Service*) pada aplikasi web harus secara eksplisit mendidik pengguna bahwa hasil pemungutan suara K-NN adalah alat deteksi dini murni probabilitas, yang tidak memiliki kekuatan hukum layaknya hasil uji laboratorium darah klinis.

7.6. Penutup: Mendemokratisasi Layanan Kesehatan

Abad ke-21 menuntut redefinisi tentang bagaimana layanan kesehatan didistribusikan. Selama berabad-abad, kecerdasan diagnostik terkunci secara eksklusif di dalam benak para praktisi medis dan di balik tembok rumah sakit-rumah sakit besar. Bagi masyarakat di daerah terpencil dengan keterbatasan akses terhadap dokter spesialis dalam (internis), fenomena keterlambatan deteksi—seperti gunung es penyakit diabetes—terus menelan korban jiwa.

Buku ini diakhiri dengan sebuah optimisme empiris. Melalui perpaduan antara ketersediaan data terbuka (*Open Data*), kesederhanaan logika matematis K-NN yang transparan, dan universalitas platform berbasis Web, kita telah membuktikan bahwa gerbang untuk memecahkan masalah tersebut kini telah terbuka lebar.

Teknologi Kecerdasan Buatan tidak hadir untuk menggantikan empati dan keahlian tangan seorang dokter, melainkan hadir sebagai kompas awal bagi masyarakat luas. Dengan mengonversi miliaran bait data historis menjadi alat penapis (*screening tool*) yang dapat diakses hanya melalui layar peramban di genggaman tangan, para akademisi, ilmuwan komputer, dan praktisi informatika medis tengah membangun jembatan peradaban yang paling mulia: memberikan peringatan dini yang mampu menyelamatkan kualitas hidup manusia, jauh sebelum komplikasi penyakit itu tiba.

DAFTAR PUSTAKA

Aghababaeyan, Z., Abdellatif, M., Briand, L., Ramesh, S., & Bagherzadeh, M. (2023). Black-Box Testing of Deep Neural Networks through Test Case Diversity. *IEEE Transactions on Software Engineering*, *49*(5), 3182–3204. https://doi.org/10.1109/TSE.2023.3243522

Ali, A. (2025). Ethics, Privacy, and Security in Healthcare Informatics. *Healthcare Informatics Innovation Post-COVID-19 Pandemic*, 180–197. https://doi.org/10.1201/9781003485629-13

Allam, H., Makubvure, L., Gyamfi, B., Graham, K. N., & Akinwolere, K. (2025). Text Classification: How Machine Learning Is Revolutionizing Text Categorization. *Information 2025, Vol. 16, 16*(2). https://doi.org/10.3390/info16020130

Almalki, W. H., & Khan, M. S. (2025). Burden of diabetes mellitus on health and economy of the Arab world: current situation and perspectives. *Journal of Public Health 2025*, 1–20. https://doi.org/10.1007/s10389-025-02448-7

An, Q., Rahman, S., Zhou, J., & Kang, J. J. (2023). A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges. *Sensors 2023, Vol. 23, 23*(9). https://doi.org/10.3390/s23094178

Arokiasamy, P., Salvi, S., & Selvamani, Y. (2021). Global Burden of Diabetes Mellitus. *Handbook of Global Health*, 1–44. https://doi.org/10.1007/978-3-030-05325-3_28-2

Awad, M. (2024). Study on Neural Network Development Tools for Web Applications and an Attempt to Advance PHP in Machine Learning Field. *Authorea Preprints*. https://doi.org/https://doi.org/10.36227/TECHRXIV.170631202.29693430/V1

Badnjević, A., & Spahić, L. (2026). *Intelligent Systems in Biomedicine*. https://doi.org/10.1007/978-3-032-14785-1

Bratianu, C., & Bejinaru, R. (2023). From Knowledge to Wisdom: Looking beyond the Knowledge Hierarchy. *Knowledge 2023, Vol. 3, Pages 196-214*, *3*(2), 196–214.

https://doi.org/10.3390/knowledge3020014
Chahar, S., & Singh, S. (2024). Analysis of SDLC Models with Web Engineering Principles. *2024 2nd International Conference on Advancements and Key Challenges in Green Energy and Computing, AKGEC 2024*. https://doi.org/10.1109/AKGEC62572.2024.10868694
Collins, T., Tello, J., Van Hilten, M., Mahy, L., Banatvala, N., Fones, G., Akselrod, S., Bull, F., Cieza, A., Farrington, J., Fisher, J., Gonzalez, C., Guerra, J., Hanna, F., Jakab, Z., Kulikov, A., Saeed, K., Abdel Latif, N., Mikkelsen, B., … Willumsen, J. (2021). Addressing the double burden of the COVID-19 and noncommunicable disease pandemics: a new global governance challenge. *International Journal of Health Governance*, *26*(2), 199–212. https://doi.org/10.1108/IJHG-09-2020-0100
Cunningham, P., & Delany, S. J. (2022). K-Nearest Neighbour Classifiers-A Tutorial. *ACM Computing Surveys*, *54*(6). https://doi.org/10.1145/3459665
Deshkar, P. A., Laghate, K., Ghorpade, A., Padole, D., Shende, H., Kawale, P., & Sakhare, P. (2024). Data Pre-processing Solution Using Statistical and Data Mining Techniques. *Lecture Notes in Networks and Systems*, *1136 LNNS*, 84–112. https://doi.org/10.1007/978-3-031-70789-6_8
Dichenko, S. A., & Finko, O. A. (2021). Controlling and Restoring the Integrity of Multi-Dimensional Data Arrays through Cryptocode Constructs. *Programming and Computer Software 2021 47:6*, *47*(6), 415–425. https://doi.org/10.1134/S0361768821060049
Eloranta, S., & Boman, M. (2022). Predictive models for clinical decision making: Deep dives in practical machine learning. *Journal of Internal Medicine*, *292*(2), 278–295. https://doi.org/10.1111/joim.13483
GabAllah, N., Farrag, I., Khalil, R., Sharara, H., & ElBatt, T. (2023). IoT systems with multi-tier, distributed intelligence: From architecture to prototype. *Pervasive and Mobile Computing*, *93*, 101818. https://doi.org/10.1016/j.pmcj.2023.101818
Goldmann, N., Skalicky, S. E., Weinreb, R. N., Guedes, R. A. P., Baudouin, C., Zhang, X., van Gestel, A., Blumenthal, E. Z., Kaufman, P. L., Rothman, R., Vasquez, A. M., Harasymowycz, P., Welsbie, D. S., & Goldberg, I. (2023). Defining functional requirements for a patient-centric computerized glaucoma treatment

and care ecosystem. *Journal of Medical Artificial Intelligence*, *6*(0). https://doi.org/10.21037/jmai-22-33

Gordon, D., & Mary Oluwaseun, J. (2025). A Web-Based Real Time Pandemic Monitoring System . *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5353919

Kim, K. (2021). Normalized class coherence change-based kNN for classification of imbalanced data. *Pattern Recognition*, *120*, 108126. https://doi.org/10.1016/j.patcog.2021.108126

Kirpalani, C., & Kumar, D. (2024). *Integrating Ethics and Social Responsibility in Health Informatics*. 99–120. https://doi.org/10.1007/978-981-97-6706-9_5

Mailagaha Kumbure, M., & Luukka, P. (2021). A generalized fuzzy k-nearest neighbor regression model based on Minkowski distance. *Granular Computing 2021 7:3*, *7*(3), 657–671. https://doi.org/10.1007/s41066-021-00288-w

Mukherjee, A., Islam, M. Z., & Ali, L. E. (2024). Human Iris Classification through Histogram of Oriented Gradient Features with Various Distance Metrics. *Machine Graphics and Vision*, *Vol. 33*, *No. 3/4*(3–4), 97–124. https://doi.org/10.22630/MGV.2024.33.3.5

Rustam, & Usman, K. (2025). A Novel Fuzzy Clustering Framework with Manhattan Distance and Weighted Median Centroids for Outlier-Resilient Data Analysis. *IEEE Access*, *13*, 190964–190979. https://doi.org/10.1109/ACCESS.2025.3629679

Saharan, S. S., Nagar, P., Creasy, K. T., Stock, E. O., Feng, J., Malloy, M. J., & Kane, J. P. (2021). Machine learning and statistical approaches for classification of risk of coronary artery disease using plasma cytokines. *BioData Mining 2021 14:1*, *14*(1), 26-. https://doi.org/10.1186/s13040-021-00260-z

Saravanos, A., & Curinga, M. X. (2023). Simulating the Software Development Lifecycle: The Waterfall Model. *Applied System Innovation 2023, Vol. 6, 6*(6). https://doi.org/10.3390/asi6060108

Saturi, S. (2022). Review on Machine Learning Techniques for Medical Data Classification and Disease Diagnosis. *Regenerative Engineering and Translational Medicine 2022 9:2*, *9*(2), 141–164. https://doi.org/10.1007/s40883-022-00273-y

Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised Classification Algorithms in Machine Learning: A Survey and Review. *Advances in Intelligent Systems and Computing*, *937*, 99–111.

https://doi.org/10.1007/978-981-13-7403-6_11
Shu, J., & Jin, W. (2023). Prioritizing non-communicable diseases in the post-pandemic era based on a comprehensive analysis of the GBD 2019 from 1990 to 2019. *Scientific Reports 2023 13:1*, *13*(1), 13325-. https://doi.org/10.1038/s41598-023-40595-7
Singh, J., Singh, J., & Gosain, A. (2026). Enhancing medical data completeness using an iterative KNN based-Kernelized fuzzy c-means imputation method. *Discover Computing 2026 29:1*, *29*(1), 9-. https://doi.org/10.1007/s10791-025-09889-4
Srinivas, N., Mandaloju, N., & Nadimpalli, S. V. (2022). Ensuring Excellence in Medical Software: A Comprehensive Guide to Quality Assurance in Healthcare Technology. *International Journal of Modern Computing*, *5*(1), 99–107. https://yuktabpublisher.com/index.php/IJMC/article/view/210
Tsaneva-Atanasova, K., Pederzanil, G., & Laviola, M. (2025). Decoding uncertainty for clinical decision-making. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *383*(2292). https://doi.org/10.1098/rsta.2024.0207
Uddin, S., Haque, I., Lu, H., Moni, M. A., & Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, *12*(1), 1–11. https://doi.org/10.1038/S41598-022-10358-X;SUBJMETA=1041,639,692,699,705;KWRD=APPLIED+MATHEMATICS,DISEASES
Vaduganathan, M., Mensah, G. A., Turco, J. V., Fuster, V., & Roth, G. A. (2022). The Global Burden of Cardiovascular Diseases and Risk: A Compass for Future Health. *Journal of the American College of Cardiology*, *80*(25), 2361–2371. https://doi.org/10.1016/j.jacc.2022.11.005
Van Der Loo, M. P. J., & De Jonge, E. (2021). Data Validation Infrastructure for R. *Journal of Statistical Software*, *97*, 1–31. https://doi.org/10.18637/jss.v097.i10
Vasey, B., Ursprung, S., Beddoe, B., Taylor, E. H., Marlow, N., Bilbro, N., Watkinson, P., & McCulloch, P. (2021). Association of Clinician Diagnostic Performance With Machine Learning–Based Decision Support Systems: A Systematic Review. *JAMA Network Open*, *4*(3), e211276–e211276. https://doi.org/10.1001/jamanetworkopen.2021.1276

Verma, N., Sharma, T., & Kaur, B. (2025). Explanation of Machine Learning Algorithms Used in Disease Detection, Such as Decision Trees and Neural Networks. *AI in Disease Detection: Advancements and Applications*, 27–52. https://doi.org/10.1002/9781394278695.ch2

Yogesh, M. J., & Karthikeyan, J. (2022). Health Informatics: Engaging Modern Healthcare Units: A Brief Overview. *Frontiers in Public Health*, *10*, 854688. https://doi.org/10.3389/fpubh.2022.854688

Younas, H. A., Shoaib Khan, B., Khan, A. H., Bilal, A., Algarni, A., Sarwar, R., & Mousavirad, S. J. (2026). Prediction of β-thalassemia carrier using federated learning and explainable AI. *Frontiers in Medicine*, *13*, 1687773. https://doi.org/10.3389/fmed.2026.1687773

Zhang, S., & Li, J. (2023). KNN Classification With One-Step Computation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(3), 2711–2723. https://doi.org/10.1109/TKDE.2021.3119140

PROFIL PENULIS

Fahmi Ruziq, S.T., M.Kom.

Penulis lahir di Banda Aceh, Aceh tanggal 16 Juni 1989. Penulis menamatkan pendidikan dasar dan menengah di Banda Aceh, setelah lulus dari SMA Negeri 1 Banda Aceh melanjutkan kuliah S1 di Universitas Serambi Mekkah Jurusan Teknik Informatika, kemudian melanjutkan S2 di Universitas Sumatera Utara Jurusan Teknik Informatika. Kemudian saat ini Penulis sedang melanjutkan Pendidikan Doktoral (S-3) Ilmu Komputer di Universitas Sumatera Utara (*on-going*). Berkarir sebagai dosen dimulai dari tahun 2020 di Universitas Battuta, Medan.

M. Rhifky Wayahdi, S.Kom., M.Kom.

.
Penulis lahir di Medan, 05 Februari 1993, merupakan anak pertama dari tiga bersaudara. Penulis merupakan alumni Program Sarjana (S-1) di Universitas Potensi Utama pada Jurusan Sistem Informasi dan lulus tahun 2015. Penulis melanjutkan studi Program Magister (S-2) Teknik Informatika di Universitas Sumatera Utara dan lulus tahun 2019. Kemudian saat ini Penulis sedang melanjutkan Pendidikan Doktoral (S-3) Ilmu Komputer di Universitas Sumatera Utara mulai 2024 sampai sekarang (on-going). Berkarir sebagai dosen dimulai dari tahun 2020 di Universitas Battuta.