BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada bagian ini disajikan beberapa penelitian terdahulu yang relevan dan dapat dijadikan sebagai referensi dalam penelitian ini. Penelitian-penelitian yang digunakan membahas mengenai perancangan dan penerapan mekanisme keamanan pada sistem basis data *chatbot* dan sistem informasi khususnya yang berkaitan dengan perlindungan data, enkripsi serta pengelolaan hak akses. Tabel 2.1 menyajikan ringkasan penelitian-penelitian tersebut yang meliputi judul artikel, tahun terbit, penulis, serta hasil utama yang diperoleh. Informasi ini digunakan untuk memberikan gambaran mengenai perkembangan penelitian sebelumnya serta memperkuat landasan teoritis dan metodologis yang menjadi dasar dalam perancangan mekanisme perlindungan basis data pada *chatbot* Palang Merah Indonesia (PMI).

Tabel 2.1 Penelitian Terdahulu

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
[23]	*A Comprehensive Review on FinWin's Architecture for Trust, Personalization, and Security*	Sarika Pabalkar, Snehal Jadhav, Rutuja Hulge, Sakshi Jagadale, Shreya Mamadapur	2024	*Database*: PostgreSQL Mekanisme Keamanan yang Digunakan: - AES-256 - TLS - RBAC Hasil Penelitian: Penelitian ini membahas tentang pengembangan platform *core banking* digital bernama FinWin yang dirancang untuk memberikan layanan perbankan yang lebih personal, responsif dan berbasis analitik. Dengan mekanisme keamanan yang diterapkan, data finansial dapat terlindungi dengan baik dari sisi penyimpanan, transmisi, maupun akses, sehingga meningkatkan kerahasiaan dan keamanan sistem secara keseluruhan.

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
[24]	*Addressing the Interoperability of Electronic Health Records: The Technical and Semantic Interoperability, Preserving Privacy and Security Framework*	Adetunji Ademola, Carlisle George, Glenford Mapp	2024	*Database*: PostgreSQL Mekanisme Keamanan yang Digunakan: - AES-256 - TLS - RBAC - SHA-512 - OAuth 2.0 Hasil Penelitian: Penelitian ini mengembangkan framework TASIPPS untuk interoperabilitas *Electronic Health Records* (EHR) yang aman dan menjaga privasi. Dari sisi keamanan, sistem ini menerapkan berbagai mekanisme seperti AES-256 untuk enkripsi data, SHA-512 untuk hashing, OAuth 2.0 untuk autentikasi, serta role-based access control (RBAC) untuk pembatasan akses. Kombinasi mekanisme tersebut memberikan dampak signifikan dalam menjaga kerahasiaan, integritas dan kontrol akses data pasien sehingga mampu mengurangi risiko kebocoran data serta meningkatkan kepercayaan dalam pertukaran data kesehatan antar sistem.
[25]	*Implementation of Secure End-to-End Encrypted Chat Application Using Diffie–Hellman Key Exchange and AES-256 in a Microservice Architecture*	Moch Juang Pajri Perkasa, Hari Muhammad Ramdan, Abdul Jabar Maliki, Somantri	2025	*Database*: PostgreSQL Mekanisme Keamanan yang Digunakan: - AES-256 - Diffie-Hellman Key Exchange Hasil Penelitian: Penelitian ini mengusulkan pengembangan aplikasi komunikasi terenkripsi end-to-end dengan memanfaatkan algoritma Diffie–Hellman Key Exchange untuk pertukaran kunci dan AES-256 sebagai mekanisme enkripsi simetris. Hasilnya, sistem

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				menunjukkan performa yang baik dengan latensi rendah dan *throughput* tinggi.
[26]	Generative AI Agents at Enterprise Scale: Architecting RAG-Enhanced LLM Systems for Production Deployment	Manikanteswara Yasaswi Kurra	2025	*Database*: Pinecone Vector Database Mekanisme Keamanan yang Digunakan: - AES-256 - TLS 1.3 - RBAC - *Attribute-Based Access Control* (ABAC) - *Zero Trust Architecture* Hasil Penelitian: Penelitian ini membahas framework untuk membangun agen AI berbasis LLM dengan pendekatan RAG guna meningkatkan akurasi dan efisiensi di lingkungan enterprise. Dari sisi keamanan, sistem ini menggunakan AES-256 dan TLS 1.3 untuk melindungi data, serta menerapkan RBAC, ABAC, dan *Zero Trust Architecture* untuk mengatur akses secara ketat. Pendekatan ini memberikan dampak yang cukup signifikan dalam menjaga keamanan data, mengurangi risiko akses tidak sah, serta mendukung kepatuhan pada lingkungan berskala besar.
[27]	*Privacy-by-Design in AI-Assisted Systems for Caregivers of Children with Autism: A Secure Multi-Agent Architecture*	Ionut Croitoru, Cristina Elena Turcu, Corneliu Octavian Turcu	2026	*Database*: Qdrant Vector Database Mekanisme Keamanan yang Digunakan: - AES-256 - TLS - RBAC Hasil Penelitian: Penelitian ini membahas pengembangan arsitektur sistem berbasis *Privacy-by-Design* yang bertujuan untuk menyediakan dukungan berbasis AI bagi *caregiver* anak dengan *Autism Spectrum Disorder*

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				(ASD) dengan tetap memperhatikan aspek privasi dan keamanan data. Hasilnya, data sensitif dapat terlindungi dengan baik pada proses penyimpanan, transmisi, dan akses, sehingga meningkatkan keamanan serta menjaga privasi pengguna dalam sistem.
[28]	*Secure and Explainable GenAI Chatbot for Financial Document Analysis*	Anil Kumar Shukla	2025	*Database*: Weavite Vector Database Mekanisme Keamanan yang Digunakan: - AES - TLS - RBAC Hasil Penelitian: Penelitian tersebut membahas pengembangan sistem chatbot berbasis Generative AI yang dirancang untuk mendukung berbagai proses kerja pada sektor keuangan yang memiliki tingkat regulasi tinggi. Hasil evaluasi menunjukkan adanya peningkatan performa sistem, ditunjukkan dengan penurunan *latency* sehingga meningkatkan kualitas dan kecepatan respons.
[29]	*Leveraging GenAI for Biometric Voice Print Authentication*	Erica Brooks, Lijo Jacob, Lani Lewis, Gaurav Mittal, Shivam Negi	2025	*Database*: Milvus Vector Database Mekanisme Keamanan yang Digunakan: - AES - TLS Hasil Penelitian: Penelitian ini membahas pengembangan sistem autentikasi suara yang aman dan inklusif bagi berbagai jenis pengguna termasuk pengguna dengan disabilitas. Hasil pengujian menunjukkan bahwa mekanisme keamanan tersebut dapat mendukung

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				pengembangan sistem autentikasi yang aman, efisien dan berpotensi digunakan dalam berbagai aplikasi berbasis identifikasi suara.
[30]	*AI-powered mental health application with data privacy preservation*	Pooja Bagane, Anushree Dahiya, Shardul Kacheria, Ansh Sehgal, Shrishti Bajpai, Obsa Amenu Jebessa	2026	*Database*: Supabase Mekanisme Keamanan yang Digunakan: - AES-256 - TLS v1.3 - RBAC Hasil Penelitian: Penelitian ini berfokus pada pengembangan sebuah sistem berbasis AI dan NLP untuk mengenali emosi manusia berdasarkan input text menggunakan *fine-tuned transformer-based architecture* yang memanfaatkan database berbasis Supabase. Hasilnya, sistem mampu menjaga kerahasiaan dan privasi pengguna melalui enkripsi, kontrol akses yang ketat serta penghapusan otomatis data sementara, sehingga meningkatkan keamanan dan akuntabilitas dalam pengolahan data sensitif.
[31]	Smart System For Blood Donation And Availability Finder	Laboni Nayak, Bal Krishan Choudhary, Sankhajit Das, Mandip Chowdhury, Ritika Halder	2025	*Database*: MongoDB & *Facebook AI Similarity Search* (FAISS) Mekanisme Keamanan yang Digunakan: - *JSON Web Token* (JWT) *based authentication* - TLS - CORS *Middleware* Hasil Penelitian: Penelitian ini mengembangkan *Smart System for Blood Donation and Availability Finder*, platform berbasis AI untuk mengotomatisasi proses donor darah, mulai dari pencarian donor hingga analisis dokumen medis.

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				Dari sisi keamanan, sistem ini menggunakan JWT untuk autentikasi, bcrypt untuk hashing password, HTTPS/TLS untuk enkripsi komunikasi, serta CORS middleware untuk membatasi akses lintas domain yang tidak sah. Kombinasi ini meningkatkan keamanan data, mencegah akses tidak sah, dan menjaga keamanan pertukaran informasi..
[32]	*Design and Implementation of Secure Image Encryption and Decryption using Advanced Cryptographic Techniques with Chatbot Integration*	K. Sreenivasulu, Godala Sunanda, B. Siva Durga, A. Lakshmi Sravanthi	2025	*Database*: SQLite Mekanisme Keamanan yang Digunakan: - AES-256 Hasil Penelitian: Penelitian ini mengusulkan model enkripsi dan dekripsi gambar yang terintegrasi dengan antarmuka chatbot untuk menyediakan komunikasi digital yang aman dan rahasia. Hasilnya, pesan teks dan gambar dapat diamankan baik saat disimpan maupun ditransmisikan, serta hanya dapat diakses oleh pihak yang berwenang melalui proses dekripsi di sisi penerima, sehingga meningkatkan kerahasiaan dan keamanan komunikasi digital.
[33]	FedVSE: A Privacy-Preserving and Efficient Vector Search Engine for Federated Databases	Zeheng Fan, Yuxiang Zeng, Zhuanglin Zheng, Yongxin Tong		*Database*: Milvus Vector Database Mekanisme Keamanan yang Digunakan: - AES-128 - Row Level Security via Access Policy Hasil Penelitian: Penelitian ini mengembangkan FedVSE, mesin pencarian vektor untuk *federated database* yang efisien dan mendukung privasi. Dari sisi

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				keamanan, penggunaan AES-128 dan *row-level security* memastikan data tetap terlindungi dan akses dibatasi sesuai otorisasi sehingga mengurangi risiko kebocoran data.
[34]	AI-powered Document Chatbot for data retrieval	Meera Raj, R. Srinivasan	2025	*Database:* FAISS *Vector Database* Mekanisme Keamanan yang Digunakan: - RBAC Hasil Penelitian: Penelitian ini mengembangkan chatbot berbasis AI untuk pengambilan data terstruktur secara otomatis dengan memanfaatkan teknologi seperti RAG, FAISS, GPT, DB2, dan LangChain. Dari sisi keamanan, penerapan RBAC memberikan dampak signifikan dengan membatasi akses pengguna hanya pada data yang diizinkan sehingga secara efektif menurunkan risiko akses tidak sah.
[35]	AI Tutors and the Transformation of Education: Opportunities, Challenges, and Future Directions	H. Vicky Zhao, Yanpin Ren, Chao Shang, Haonan Yin, Zhanwei Xu, Rui Jiang, Benben Jiang, Qing Zhuo, Hangjing Zhang, Xin Pei, Changshui Zhang, Hua Geng	2026	*Database: FAISS vector database* Mekanisme Keamanan yang Digunakan: - RBAC - JWT *based authentication* Hasil Penelitian: Penelitian ini mengembangkan platform *AI Learning Companion* untuk mendukung pembelajaran yang personal dan sesuai kurikulum. Hasilnya, sistem mampu meningkatkan akses belajar serta memberikan insight bagi pengajar. Dari sisi keamanan, penerapan JWT, RBAC, dan rate limiting memantu menjaga akses tetap terkontrol dan mencegah penyalahgunaan,

Ref	Judul Artikel	Penulis	Tahun	Detail Penelitian
				sehingga sistem berjalan lebih aman dan stabil.
[36]	*Chatbots: Security, privacy, data protection, and social aspects*	Martin Hasal, Jana Nowaková, Khalifa Ahmed Saghair, Hussam Abdulla, Václav Snášel, Lidia Ogiela	2021	Metode: *Systematic Literature Review* Hasil Penelitian: Studi ini menunjukkan bahwa arsitektur chatbot memiliki berbagai kerentanan keamanan, seperti penyadapan, manipulasi data, akses tidak sah, dan kebocoran informasi, terutama pada integrasi dengan platform pihak ketiga. Ancaman utama meliputi *spoofing*, *denial of service* dan eskalasi hak akses. Hasilnya, penelitian merekomendasikan penerapan enkripsi (AES dan TLS), autentikasi dan otorisasi yang kuat, pembatasan akses serta kebijakan minimisasi data untuk meningkatkan keamanan dan perlindungan privasi pada sistem chatbot.
[37]	Information security and confidentiality in health chatbots: A scoping review and development of a conceptual model	Tahere Talebi Azadboni, Fahimeh Solat, Meysam Rahmani, Hanieh Hematti	2025	Metode: *Systematic Literature Review* Hasil: Tantangan umum pada chatbot kesehatan meliputi pelanggaran privasi, kurangnya transparansi, persetujuan yang tidak lengkap, masalah teknis pengelolaan data, keterbatasan regulasi, serta ancaman yang terus berkembang. Literatur menyarankan solusi seperti enkripsi, manajemen risiko, kontrol akses, standarisasi, dan evaluasi berkala. Berdasarkan itu, dikembangkan model konseptual dengan empat dimensi utama yang mengintegrasikan perangkat lunak, perangkat keras dan middleware untuk meningkatkan keamanan serta kerahasiaan data.

Berdasarkan berbagai penelitian yang telah dikaji, dapat dilihat bahwa perkembangan sistem chatbot dan aplikasi berbasis AI saat ini sangat dipengaruhi oleh kebutuhan akan pemrosesan data berbasis semantic search serta penguatan mekanisme keamanan pada seluruh siklus data. Berbagai studi menunjukkan bahwa penggunaan vector embedding telah menjadi pendekatan umum dalam mendukung pencarian informasi berbasis makna, seperti yang diterapkan pada sistem chatbot, analisis dokumen, autentikasi biometrik, hingga sistem pembelajaran adaptif. Penelitian seperti Kurra [26], Croitoru et al. [27], Shukla [28], dan Brooks et al. [29]menunjukkan bahwa semantic search berbasis embedding dapat meningkatkan relevansi hasil pencarian dan efisiensi sistem dalam berbagai domain aplikasi. Hal serupa juga terlihat pada penelitian Raj & Srinivasan [34] serta Zhao et al. [35]yang memanfaatkan pendekatan retrieval untuk mendukung chatbot dokumen dan sistem pembelajaran berbasis AI. Selain itu, Nayak et al. [31]juga mengimplementasikan pendekatan AI untuk pencarian donor darah berbasis *similarity matching*, sementara Fan et al. [33] menunjukkan pengembangan vector search engine pada lingkungan federated database, yang semakin menegaskan luasnya penerapan teknik *semantic retrieval* dalam berbagai konteks sistem.

Namun secara lebih lanjut, peningkatan kemampuan pemrosesan berbasis AI tersebut belum sepenuhnya diimbangi dengan pendekatan keamanan yang terintegrasi secara menyeluruh. Sebagian besar penelitian masih menerapkan mekanisme keamanan secara parsial terutama pada aspek enkripsi data, autentikasi, dan kontrol akses. Mekanisme seperti AES dan TLS banyak digunakan untuk melindungi data saat disimpan maupun ditransmisikan, sedangkan RBAC dan JWT umum digunakan untuk mengatur hak akses pengguna. Beberapa penelitian juga menambahkan mekanisme yang lebih kompleks seperti OAuth 2.0, ABAC, Zero Trust Architecture, SHA-512, hingga row-level security, seperti yang terlihat pada penelitian Ademola et al. [24], Kurra [26], dan Fan et al. [33]. Selain itu, penelitian seperti Pabalkar et al. [23] pada sistem core banking serta Bagane et al. [30]pada aplikasi kesehatan mental berbasis AI juga menunjukkan penerapan kombinasi mekanisme keamanan yang kuat untuk menjaga kerahasiaan dan integritas data.

Meskipun demikian, penerapan tersebut masih cenderung terpisah dan belum sepenuhnya terintegrasi pada seluruh lapisan sistem. Selain itu, penelitian terkait interoperabilitas data kesehatan oleh Ademola et al. [24] serta studi pada sistem donor darah oleh Nayak et al. [31] menunjukkan bahwa integrasi antar sistem masih menjadi tantangan tersendiri, terutama dalam menjaga konsistensi keamanan dan kontrol akses. Sementara itu, penelitian Sreenivasulu et al. [32] dan Perkasa et al. [25] menekankan aspek keamanan komunikasi dan enkripsi data, namun masih berfokus pada skenario spesifik tanpa integrasi dengan sistem AI yang lebih kompleks. Studi SLR oleh Hasal et al. [36] dan Azadboni et al. [37] juga menegaskan bahwa chatbot dan sistem AI memiliki banyak titik kerentanan, seperti penyadapan, manipulasi data, akses tidak sah hingga kebocoran informasi, sehingga membutuhkan pendekatan keamanan yang mencakup seluruh siklus data.

Dari seluruh penelitian tersebut, dapat disimpulkan bahwa masih terdapat kesenjangan utama yaitu belum adanya pendekatan yang secara konsisten mengintegrasikan kemampuan semantic search berbasis AI dengan mekanisme keamanan pada layer basis data dalam satu arsitektur yang sederhana dan efisien. Di satu sisi, sistem telah berkembang dalam hal kemampuan pemrosesan dan kecerdasan, namun di sisi lain, aspek keamanan masih cenderung terfragmentasi dan belum terintegrasi secara end-to-end. Oleh karena itu, penelitian ini mengarah pada pendekatan berbasis pgvector pada PostgreSQL yang memungkinkan penggabungan data relasional dan vector embedding dalam satu sistem yang terintegrasi. Penggunaan pgvector memungkinkan proses semantic search dilakukan langsung pada lingkungan PostgreSQL tanpa memerlukan vector database terpisah, sehingga integrasi data, pengelolaan query, serta implementasi mekanisme keamanan dapat dilakukan dalam satu sistem basis data yang sama. Pendekatan ini dipilih untuk menyederhanakan arsitektur sistem sesuai kebutuhan penelitian tanpa menghilangkan kemampuan semantic search yang diperlukan pada chatbot berbasis AI. Selain itu, sistem ini dilengkapi dengan mekanisme perlindungan basis data yang mencakup enkripsi data at rest menggunakan AES-256, pengamanan komunikasi menggunakan TLS melalui OpenSSL, serta kontrol akses menggunakan RBAC dan RLS yang diimplementasikan langsung pada level

basis data. Penerapan mekanisme tersebut diharapkan dapat membantu menjaga kerahasiaan, integritas, dan kontrol akses data secara lebih terintegrasi pada sistem chatbot berbasis AI.

2.2 Teori yang Berkaitan

2.2.1 Data Rahasia (Confidential)

Data rahasia (*confidential data*) merupakan informasi yang harus dilindungi dari akses, penggunaan atau pengungkapan oleh pihak yang tidak berwenang [38] [39]. Dalam keamanan informasi, kerahasiaan (*confidentiality*) adalah salah satu prinsip utama yang bertujuan memastikan bahwa informasi hanya dapat diakses oleh individu atau sistem yang memiliki otorisasi yang sah. Data dikategorikan sebagai *confidential* apabila pengungkapan tanpa izin dapat menyebabkan kerugian, baik secara finansial, reputasi, hukum maupun operasional bagi individu atau organisasi [40]. Oleh karena itu, data yang bersifat rahasia memerlukan mekanisme pengamanan khusus dibandingkan dengan data publik.

Berdasarkan Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Perlindungan Data Pribadi, data pribadi dibedakan menjadi dua kategori, yaitu data pribadi yang bersifat spesifik dan data pribadi yang bersifat umum [41]. Data pribadi yang bersifat spesifik mencakup informasi yang memiliki tingkat sensitivitas tinggi seperti data dan informasi kesehatan, data biometrik, data genetika, catatan kejahatan, data anak serta data keuangan pribadi. Sementara itu, data pribadi yang bersifat umum meliputi informasi dasar seperti nama lengkap, jenis kelamin, kewarganegaraan, agama dan status perkawinan serta kombinasi data yang dapat digunakan untuk mengidentifikasi seseorang [41]. Selain diterapkan pada regulasi domestik, perlindungan terhadap data pasien juga diatur dalam berbagai standar internasional untuk memastikan keamanan dan privasi data secara lebih luas.

Pada ranah internasional, perlindungan terhadap data pasien diatur melalui berbagai standar dan regulasi internasional salah satunya adalah *Health Insurance Portability and Accountability Act* (HIPAA). Regulasi ini

mendefinisikan konsep *Protected Health Information* (PHI) sebagai setiap informasi kesehatan yang dapat digunakan untuk mengidentifikasi individu dan dimiliki atau ditransmisikan oleh *covered entity* atau pihak yang berafiliasi dengannya (*business associates*), yang berkaitan dengan kondisi kesehatan pasien di masa lalu, masa kini, maupun potensi kondisi kesehatan di masa mendatang [42]. PHI mencakup berbagai jenis data seperti identitas pasien, catatan medis, hasil pemeriksaan laboratorium, riwayat pengobatan hingga informasi administratif seperti nomor pasien dan nomor asuransi kesehatan [43]. HIPAA juga menetapkan sejumlah elemen identifikasi (*identifiers*) yang termasuk dalam kategori PHI seperti nama, alamat, tanggal lahir, nomor telepon, alamat email, nomor rekam medis serta informasi biometrik yang dapat digunakan untuk mengidentifikasi individu secara spesifik.

Keberadaan PHI menjadikan data kesehatan sebagai salah satu jenis informasi yang memiliki tingkat sensitivitas tinggi dan memerlukan perlindungan yang ketat. Pengungkapan informasi kesehatan tanpa izin dapat menimbulkan berbagai risiko seperti pelanggaran privasi, penyalahgunaan data, hingga kerugian bagi pasien maupun institusi layanan kesehatan [39]. Oleh karena itu, pengelolaan PHI dalam sistem informasi kesehatan umumnya memerlukan penerapan berbagai mekanisme keamanan, seperti pengendalian akses, enkripsi data serta pengamanan proses penyimpanan dan transmisi informasi [42]. Dengan demikian, seluruh sistem yang mengelola data kesehatan perlu memastikan bahwa seluruh proses pengolahan dan penyimpanan informasi pasien dilakukan dengan memperhatikan prinsip kerahasiaan serta perlindungan terhadap PHI guna mencegah terjadinya kebocoran atau akses yang tidak sah terhadap data sensitif tersebut.

Berdasarkan pada regulasi yang ditetapkan oleh *Department of Health and Human Services* melalui HIPAA *Security Rule* menegaskan pentingnya penerapan mekanisme keamanan yang komprehensif dalam melindungi data yang tergolong sebagai PHI khususnya dalam bentuk elektronik (*electronic PHI* atau ePHI). Salah satu aspek utama yang diatur adalah *access control* sebagaimana tercantum dalam *Section* 164.312(a)(1) yang menyatakan bahwa

akses terhadap sistem yang menyimpan atau memproses ePHI harus dibatasi hanya kepada individu atau entitas yang memiliki hak otorisasi. Ketentuan ini menekankan prinsip pembatasan akses secara ketat (*least privilege*) dimana setiap pengguna hanya dapat mengakses informasi yang relevan dengan tugas dan tanggung jawabnya. Selain itu, HIPAA juga mengatur perlindungan data melalui mekanisme enkripsi dan dekripsi sebagaimana dijelaskan dalam *Section* 164.312(b)(1) mengenai *Encryption and Decryption*. Dalam ketentuan ini, enkripsi diposisikan sebagai salah satu metode utama untuk menjaga kerahasiaan ePHI dari potensi ancaman baik yang berasal dari dalam maupun luar sistem. Dengan menerapkan enkripsi, data yang tersimpan tidak dapat diinterpretasikan secara langsung tanpa kunci atau proses dekripsi yang sesuai sehingga memberikan lapisan perlindungan tambahan terhadap informasi sensitif.

Lebih lanjut, aspek *transmission security* yang diatur dalam *Section* 164.312(g) menekankan pentingnya perlindungan data selama proses transmisi melalui jaringan. Dalam *system environment* yang saling terhubung, ePHI seringkali ditransmisikan antar sistem atau layanan sehingga rentan terhadap berbagai ancaman seperti penyadapan (*eavesdropping*) atau manipulasi data oleh pihak yang tidak berwenang. Oleh karena itu, regulasi ini mengharuskan adanya mekanisme yang mampu menjaga kerahasiaan dan integritas data selama proses pengiriman berlangsung. Perlindungan ini bertujuan untuk memastikan bahwa data yang dikirim tetap utuh, tidak berubah, dan hanya dapat diakses oleh pihak yang berhak. Sehingga, gabungan dari beberapa pendekatan tersebut mencerminkan penerapan prinsip keamanan informasi yang terintegrasi dimana setiap lapisan keamanan saling melengkapi untuk menjaga kerahasiaan, integritas dan ketersediaan data. Oleh karena itu, regulasi HIPAA dapat dijadikan sebagai landasan teoritis yang kuat dalam memahami standar dan kebutuhan keamanan yang harus dipenuhi dalam melakukan perancangan sistem maupun itu sistem basis data ataupun sistem informasi pada bidang kesehatan.

2.2.2 Vector Database pada Chatbot

Vector database merupakan salah satu jenis sistem basis data yang dirancang untuk menyimpan dan mengelola data dalam bentuk representasi numerik yang dikenal sebagai *vector embeddings* [44]. *Embedding* merupakan representasi matematis dari suatu data seperti teks, gambar atau audio yang diubah menjadi serangkaian nilai numerik sehingga memungkinkan sistem komputer untuk memahami hubungan semantik antar data [44]. Berbeda dengan basis data relasional tradisional yang menyimpan data dalam bentuk tabel terstruktur yang terdiri dari baris dan kolom serta melakukan pencarian berdasarkan kecocokan nilai atau kata kunci menggunakan query seperti SQL, *vector database* menyimpan data dalam bentuk vektor berdimensi tinggi dan melakukan proses pencarian berdasarkan tingkat kemiripan antar vektor [45]. Pendekatan ini memungkinkan sistem untuk menemukan hubungan makna antar data meskipun tidak memiliki kesamaan kata secara langsung. Contohnya, dalam *Natural Language Processing* (NLP), *embedding* digunakan untuk merepresentasikan makna suatu kalimat atau kata dalam bentuk vektor berdimensi tinggi sehingga sistem dapat mengidentifikasi kedekatan makna antara dua data secara semantik [46]. Melalui konsep tersebut, biasanya proses pencarian informasi akan dilakukan berdasarkan dari kesamaan konteks atau makna (*semantic similarity*) bukan hanya berdasarkan kecocokan kata kunci seperti pada sistem pencarian tradisional.

Selain itu, *vector database* juga seringkali ditemukan implementasinya pada sistem *chatbot*. Pada sistem *chatbot*, *vector database* digunakan untuk mendukung proses pencarian informasi secara lebih kontekstual. Ketika pengguna mengirimkan pertanyaan kepada *chatbot*, teks pertanyaan tersebut terlebih dahulu diubah menjadi *vector embedding* melalui sebuah model untuk mengolah bahasa tersebut [44]. Selanjutnya, sistem akan melakukan proses pencarian pada *vector database* dengan membandingkan *embedding* pertanyaan pengguna dengan *embedding* data yang tersimpan dalam basis data [47]. Proses ini umumnya dilakukan menggunakan metode pengukuran kemiripan seperti *cosine similarity*, *Euclidean distance* atau *dot product* untuk menemukan data yang memiliki kedekatan makna paling tinggi dengan pertanyaan pengguna.

Informasi yang memiliki tingkat kemiripan tertinggi kemudian digunakan sebagai dasar dalam menghasilkan respons yang relevan kepada pengguna. Pendekatan seperti ini umumnya digunakan dalam arsitektur *chatbot* berbasis pendekatan *Retrieval-Augmented Generation* (RAG) [47]. Penggunaan *vector database* dalam sistem chatbot memberikan kemampuan pencarian informasi yang lebih fleksibel dan kontekstual dibandingkan dengan sistem basis data relasional. Terdapat beberapa teknologi yang umum digunakan untuk mengimplementasikan *vector database* antara lain FAISS, Pinecone, Weaviate, Milvus serta *extension* pgvector pada PostgreSQL yang memungkinkan penyimpanan dan pemrosesan embedding secara langsung pada sistem basis data relasional.

2.2.3 Tantangan Implementasi Vector Database

Vector database banyak digunakan dalam sistem berbasis AI karena kemampuannya dalam menyimpan berbagai jenis data seperti teks, gambar, suara dan video dalam bentuk *vector embedding* sehingga memungkinkan dilakukannya *semantic search* berdasarkan kesamaan makna (*similarity*) atau konteks [44] [48]. Meskipun demikian, implementasi *vector database* tidak terlepas dari berbagai tantangan. Salah satu tantangan utama adalah menjaga keseimbangan antara kecepatan dan akurasi dimana sistem yang dibangun seringkali harus memilih antara kecepatan proses tinggi namun dengan hasil yang kurang akurat atau hasil yang akurat dengan waktu komputasi yang lebih lama [44]. Hal ini menjadi semakin kompleks ketika sistem harus menangani data dalam skala besar dan secara *real-time*.

Selain itu, kebutuhan penyimpanan pada *vector database* cenderung lebih besar dibandingkan dengan *relational database* karena data disimpan dalam bentuk vektor berdimensi tinggi [44]. Semakin tinggi dimensi vektor yang digunakan, maka kompleksitas perhitungan dalam proses *similarity search* juga meningkat yang dapat berdampak pada penurunan efisiensi sistem. Distribusi data yang semakin luas dan beragam juga turut mempersulit proses pencarian kemiripan karena hubungan antar data menjadi lebih sulit untuk direpresentasikan secara akurat dalam ruang vektor. Tantangan lain yang

signifikan adalah fluktuasi akurasi pada proses pengindeksan vektor [48]. Dalam ruang berdimensi tinggi, jarak antar data cenderung menjadi lebih seragam sehingga menyulitkan sistem dalam membedakan mana data yang benar-benar relevan. Selain itu, pembaruan data dalam jumlah besar atau perubahan yang dilakukan secara drastis dapat menyebabkan indeks yang sudah ada menjadi kurang optimal sehingga menurunkan akurasi pencarian dan berpotensi menimbulkan kesalahan dalam hasil retrieval.

Tantangan lain yang sering muncul berhubungan dengan data dari berbagai sumber seperti teks, gambar dan audio harus digabungkan dalam satu ruang representasi vektor. Perbedaan karakteristik, distribusi, dan metode ekstraksi fitur dari masing-masing jenis data dapat menyebabkan ketidaksesuaian semantik (*semantic mismatch*) sehingga menurunkan kualitas pencarian [48]. Proses *feature fusion* yang melibatkan transformasi dan normalisasi data juga meningkatkan kompleksitas dari proses yang berujung meningkatkan latensi [48]. Selain itu, *vector database* masih menghadapi kesulitan dalam menangani struktur data yang kompleks serta fitur antar data yang memiliki korelasi lemah yang pada akhirnya dapat memperlambat respons sistem dan menurunkan performa secara keseluruhan [48]. Berdasarkan tantangan tersebut, berbagai solusi diterapkan untuk melakukan mitigasi seperti menerapkan metode pendekatan untuk optimasi metode embedding, menggunakan teknik *indexing* yang lebih efisien serta meningkatkan strategi pengolahan data agar performa dan akurasi sistem tetap terjaga [48].

2.2.4 Enkripsi

Enkripsi merupakan salah satu mekanisme keamanan yang digunakan untuk melindungi data dengan cara mengubah data asli (*plaintext*) menjadi bentuk yang tidak dapat dipahami (*ciphertext*) [49]. Proses enkripsi dilakukan menggunakan algoritma dan kunci tertentu sehingga hanya pihak yang memiliki kunci yang sesuai yang dapat melakukan proses dekripsi untuk mengembalikan data ke bentuk semula [50]. Enkripsi bertujuan untuk menjaga kerahasiaan dan keamanan data dari akses yang tidak sah. Dalam sistem informasi, enkripsi diterapkan pada berbagai lapisan, baik untuk data yang disimpan (*data at rest*)

maupun data yang dikirimkan melalui jaringan (*data in transit*) [36]. Penggunaan enkripsi pada *data at rest* bertujuan untuk melindungi data dari kebocoran akibat akses fisik atau serangan terhadap sistem penyimpanan. Sementara itu, enkripsi pada *data in transit* digunakan untuk mencegah penyadapan atau manipulasi data selama proses komunikasi antar sistem. Berbagai algoritma enkripsi digunakan sesuai dengan kebutuhan sistem seperti algoritma simetris yang menggunakan satu kunci untuk enkripsi dan dekripsi, serta algoritma asimetris yang menggunakan pasangan kunci publik dan privat [51]. Penerapan enkripsi yang tepat dapat meningkatkan tingkat keamanan sistem, mengurangi risiko kebocoran data sensitif, serta menjaga kepercayaan pengguna terhadap sistem yang dikembangkan.

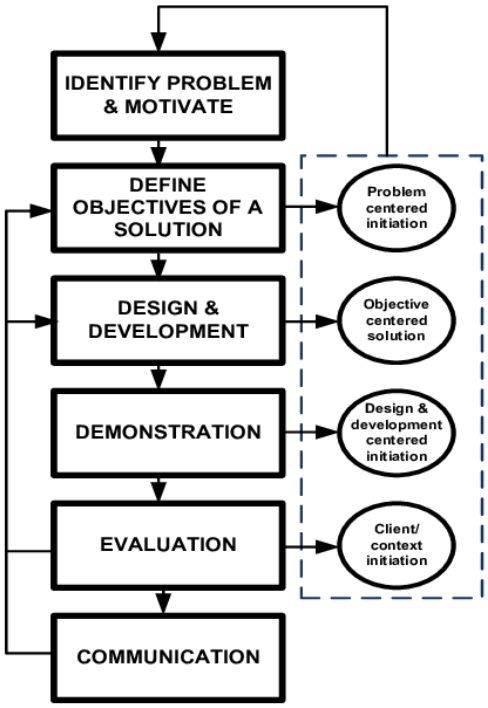
2.2.5 Kontrol Akses

Kontrol akses (access control) merupakan mekanisme keamanan yang digunakan untuk mengatur dan membatasi hak akses pengguna terhadap sumber daya dalam suatu sistem informasi [52]. Kontrol akses bertujuan untuk memastikan bahwa setiap pengguna hanya dapat mengakses data dan fungsi sistem sesuai dengan kewenangan yang dimilikinya. Mekanisme ini berperan penting dalam menjaga keamanan, kerahasiaan, dan integritas data. Penerapan kontrol akses umumnya melibatkan proses autentikasi dan otorisasi. Autentikasi digunakan untuk memverifikasi identitas pengguna, sedangkan otorisasi menentukan hak akses yang diberikan setelah identitas pengguna berhasil diverifikasi. Model kontrol akses yang umum digunakan antara lain Role-Based Access Control (RBAC) dimana hak akses ditentukan berdasarkan peran pengguna serta Attribute-Based Access Control (ABAC) yang mempertimbangkan atribut tertentu seperti waktu, lokasi atau jenis perangkat [53]. Dengan penerapan kontrol akses yang baik, sistem informasi dapat mencegah penyalahgunaan hak akses serta meminimalkan risiko kebocoran data. Selain itu, kontrol akses juga mendukung penerapan prinsip least privilege, di mana pengguna hanya diberikan hak akses minimum yang diperlukan untuk menjalankan tugasnya. Hal ini menjadikan sistem lebih aman, terstruktur, dan mudah untuk diaudit.

2.3 Framework/Algoritma yang digunakan

2.3.1 Design Science Research Methodology

Design Science Research Methodology (DSRM) merupakan pendekatan penelitian yang berfokus pada perancangan dan pengembangan artefak untuk menyelesaikan permasalahan yang dihadapi dalam suatu domain tertentu [54]. Artefak yang dihasilkan dapat berupa model, metode, kerangka kerja atau sistem informasi yang dirancang untuk memberikan solusi yang inovatif dan efektif. DSRM menekankan pada penciptaan pengetahuan melalui proses desain dan evaluasi artefak yang dikembangkan.



Gambar 2.1 *Design Science Research Methodology* [54]

Berdasarkan gambar 3.1, Design Science Research Methodology (DSRM) terdiri dari enam tahapan utama yang saling berkesinambungan dimulai dari *identify problem and motivate*, *define objectives of a solution*, *design & development*, *demonstration*, *evaluation* dan *communication* hingga komunikasi hasil penelitian. Berikut adalah penjelasan dari masing-masing tahapan dari Design Science Research.

1. *Identify Problem and Motivate*. Tahapan *Identify Problem and Motivate* merupakan tahap awal dalam DSRM yang berfokus pada identifikasi permasalahan yang terjadi dalam suatu konteks tertentu. Pada tahap ini, peneliti menganalisis kondisi eksisting untuk menemukan permasalahan yang relevan dan memiliki dampak signifikan. Permasalahan yang diidentifikasi harus memiliki nilai praktis dan layak untuk diselesaikan melalui pendekatan teknologi atau sistem informasi. Selain mengidentifikasi masalah, tahap ini juga bertujuan untuk memberikan motivasi mengapa permasalahan tersebut perlu diselesaikan. Motivasi ini menjelaskan urgensi penelitian serta manfaat yang dapat diperoleh apabila solusi berhasil dikembangkan. Pada gambar, tahap ini dikategorikan sebagai *problem centered initiation*, yang menekankan bahwa penelitian dimulai dari permasalahan nyata yang ada.
2. *Define Objectives of a Solution*. Tahap Define Objectives of a Solution bertujuan untuk menetapkan tujuan yang ingin dicapai dari solusi yang akan dikembangkan. Tujuan tersebut dirumuskan berdasarkan permasalahan yang telah diidentifikasi pada tahap sebelumnya. Pada tahap ini, peneliti menentukan kriteria keberhasilan solusi, baik dari sisi fungsionalitas, kinerja, maupun aspek keamanan dan efisiensi. Tahapan ini termasuk dalam objective centered solution, di mana fokus utama penelitian adalah mendefinisikan karakteristik solusi yang diharapkan. Tujuan yang dirumuskan harus jelas, terukur dan selaras dengan kebutuhan pengguna maupun konteks organisasi. Dengan adanya tujuan yang terdefinisi dengan baik, proses perancangan solusi dapat dilakukan secara lebih terarah.
3. *Design and Development*. Tahap *Design and Development* merupakan tahap inti dalam DSRM, di mana peneliti mulai merancang dan mengembangkan artefak sebagai solusi dari permasalahan yang telah didefinisikan. Artefak yang dikembangkan dapat berupa model, metode, arsitektur sistem, atau aplikasi berbasis teknologi informasi. Pada tahap ini, peneliti menerjemahkan tujuan solusi ke dalam desain teknis dan implementasi nyata. Proses desain dan pengembangan dilakukan secara

iteratif untuk memastikan bahwa artefak yang dihasilkan sesuai dengan kebutuhan dan tujuan penelitian. Dalam gambar, tahap ini dikategorikan sebagai *design & development centered initiation*, yang menekankan fokus pada proses perancangan solusi.

4. *Demonstration*. Tahap *Demonstration* bertujuan untuk menunjukkan bagaimana artefak yang telah dikembangkan dapat digunakan untuk menyelesaikan permasalahan yang telah diidentifikasi. Pada tahap ini, solusi diuji dalam skenario tertentu, baik melalui simulasi, studi kasus, maupun penerapan langsung dalam lingkungan terbatas. Demonstrasi dilakukan untuk membuktikan bahwa artefak yang dikembangkan dapat berfungsi sebagaimana mestinya dan mampu memenuhi tujuan yang telah ditetapkan. Tahap ini juga membantu dalam mengidentifikasi kekurangan awal pada solusi sebelum dilakukan evaluasi lebih lanjut.
5. *Evaluation*. Tahap *Evaluation* merupakan tahap untuk menilai kinerja dan efektivitas artefak yang telah dikembangkan. Evaluasi dilakukan dengan membandingkan hasil yang diperoleh dengan tujuan dan kriteria keberhasilan yang telah didefinisikan sebelumnya. Metode evaluasi dapat berupa pengujian fungsional, pengujian kinerja, maupun penilaian berdasarkan umpan balik pengguna. Dalam gambar, tahap evaluasi termasuk dalam *client context initiation*, yang menunjukkan bahwa penilaian dilakukan dengan mempertimbangkan konteks dan kebutuhan pengguna. Hasil evaluasi digunakan untuk menentukan apakah solusi telah memenuhi tujuan penelitian atau masih memerlukan perbaikan dan pengembangan lanjutan.
6. *Communication*. Tahap terakhir dalam DSRM adalah *Communication*, yaitu proses penyampaian hasil penelitian kepada pihak terkait. Komunikasi dilakukan melalui penulisan laporan penelitian, publikasi ilmiah, atau presentasi kepada pemangku kepentingan. Tahap ini bertujuan untuk menyebarluaskan pengetahuan yang dihasilkan dari penelitian. Pada tahap ini, peneliti menjelaskan permasalahan yang diangkat, proses pengembangan solusi, hasil evaluasi, serta kontribusi penelitian terhadap bidang keilmuan dan praktik. Komunikasi yang baik

memastikan bahwa hasil penelitian dapat dipahami, dimanfaatkan, dan dikembangkan lebih lanjut oleh peneliti atau praktisi lainnya.

2.3.2 *Database Life Cycle* (DBLC)

Database Life Cycle (DBLC) merupakan suatu pendekatan sistematis yang digunakan sebagai pedoman dalam mengembangkan basis data mulai dari tahap perencanaan hingga pemeliharaan [55]. Pendekatan ini memandang bahwa pembangunan basis data merupakan suatu proses yang berlangsung secara bertahap dan saling berkaitan, bukan sekadar aktivitas implementasi teknis. Setiap fase dalam DBLC memiliki tujuan tertentu untuk menghasilkan basis data yang mampu memenuhi kebutuhan informasi organisasi, menjaga konsistensi dan integritas data, serta mendukung keberlangsungan sistem informasi dalam jangka panjang. Dengan mengikuti tahapan DBLC, proses pengembangan basis data dapat dilakukan secara lebih terstruktur sehingga risiko kesalahan desain maupun inkonsistensi data selama implementasi dapat diminimalkan [55], [56], [57].

Fase pertama dalam DBLC adalah *database planning* dan *requirements analysis*. Tahap ini bertujuan untuk mendefinisikan tujuan pembangunan basis data sekaligus mengidentifikasi kebutuhan informasi yang diperlukan oleh pengguna maupun organisasi. Proses yang dilakukan meliputi analisis proses bisnis, identifikasi data yang dibutuhkan, penentuan sumber data serta pemahaman terhadap pola akses dan pemanfaatan data di dalam sistem. Luaran dari tahap ini adalah spesifikasi kebutuhan basis data yang menjadi landasan bagi proses perancangan pada tahap berikutnya. Dalam DBLC, analisis kebutuhan memiliki peran yang sangat penting karena kualitas rancangan basis data sangat dipengaruhi oleh ketepatan dalam mengidentifikasi kebutuhan informasi. Kesalahan pada tahap ini dapat menyebabkan basis data yang dikembangkan tidak mampu mendukung proses bisnis secara optimal.

Setelah kebutuhan sistem berhasil dirumuskan, proses dilanjutkan ke tahap perancangan konseptual (*conceptual database design*). Pada tahap ini dilakukan penyusunan model data yang menggambarkan struktur informasi secara abstrak tanpa mempertimbangkan teknologi maupun *Database Management System*

(DBMS) yang akan digunakan. Representasi tersebut umumnya menggunakan *Entity Relationship Diagram* (ERD) untuk memodelkan entitas, atribut dan hubungan antarentitas. Fokus utama dari perancangan konseptual adalah menghasilkan representasi data yang mampu menggambarkan kebutuhan sistem secara lengkap berdasarkan sudut pandang pengguna. Dengan demikian, perhatian perancang tidak terfokus pada implementasi teknis, melainkan pada makna dan hubungan antar data yang akan dikelola.

Tahapan berikutnya adalah perancangan logikal (*logical database design*), yaitu proses mengubah model konseptual menjadi skema logikal yang sesuai dengan model basis data yang dipilih, umumnya model relasional. Pada fase ini, setiap entitas dipetakan menjadi tabel, atribut diterjemahkan menjadi kolom, sedangkan hubungan antarentitas direalisasikan melalui penentuan *primary key* dan *foreign key*. Selain itu, proses normalisasi dilakukan untuk mengurangi redundansi data sekaligus meningkatkan konsistensi penyimpanan informasi. Hasil dari tahap ini adalah struktur basis data yang telah siap untuk diimplementasikan tanpa menghilangkan makna data yang telah dirancang pada tahap konseptual.

Sesudah desain logikal selesai disusun, proses berlanjut ke perancangan fisik (*physical database design*). Tahap ini berorientasi pada implementasi skema logikal ke dalam DBMS yang digunakan dengan mempertimbangkan karakteristik teknis sistem. Aktivitas yang dilakukan meliputi penentuan tipe data setiap atribut, pembuatan indeks, pengaturan struktur penyimpanan, serta penyusunan strategi akses terhadap data. Tujuan utama dari perancangan fisik adalah memperoleh performa basis data yang optimal, baik dari segi kecepatan pemrosesan maupun efisiensi penggunaan sumber daya. Oleh karena itu, keputusan yang diambil pada tahap ini memiliki pengaruh yang signifikan terhadap kinerja sistem, khususnya pada aplikasi yang mengelola data dalam jumlah besar dan memiliki frekuensi akses yang tinggi.

Tahap selanjutnya adalah implementasi dan pengujian basis data. Implementasi dilakukan dengan membangun skema basis data pada DBMS berdasarkan hasil perancangan fisik yang telah disusun sebelumnya. Setelah proses implementasi selesai, dilakukan pengujian untuk memastikan bahwa

seluruh komponen basis data telah berfungsi sebagaimana mestinya. Pengujian meliputi pemeriksaan struktur tabel, validasi integritas data, serta evaluasi performa terhadap operasi maupun query yang dijalankan. Melalui tahapan ini dapat dipastikan bahwa basis data tidak hanya memenuhi aspek desain, tetapi juga mampu mendukung kebutuhan operasional sistem secara andal.

Fase terakhir dalam DBLC adalah operasi dan pemeliharaan (*operation and maintenance*). Pada tahap ini basis data telah digunakan sebagai bagian dari sistem yang berjalan dan memerlukan pengelolaan secara berkelanjutan. Kegiatan pemeliharaan mencakup pelaksanaan pencadangan data (*backup*), optimasi query, pemantauan performa, peningkatan keamanan serta penyesuaian struktur basis data apabila terjadi perubahan kebutuhan organisasi atau sistem. DBLC menempatkan proses pemeliharaan sebagai bagian penting dari siklus hidup basis data karena kebutuhan informasi dapat berubah seiring waktu. Oleh sebab itu, basis data harus mampu beradaptasi terhadap perubahan tersebut agar tetap mendukung operasional sistem secara efektif.

2.3.3 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) adalah alat visual yang digunakan untuk merancang dan menggambarkan struktur suatu *database* [58]. ERD menunjukkan hubungan atau relasi antar entitas atau objek dalam basis data beserta atribut-atribut yang dimiliki oleh setiap entitas tersebut. Dengan kata lain, ERD merupakan model yang digunakan untuk menjelaskan bagaimana data dalam basis data saling berhubungan berdasarkan objek-objek utama yang memiliki keterkaitan [59]. Dengan merepresentasikan entitas dan hubungannya secara visual, ERD membantu dalam menganalisis dan memvalidasi desain *database* sehingga memastikan bahwa sistem yang dirancang dapat memenuhi kebutuhan yang diinginkan.

Secara umum, ERD terdiri atas beberapa komponen utama, yaitu entitas (*entity*), atribut (*attribute*), relasi (*relationship*) dan kardinalitas (*cardinality*) [60]. Entitas merepresentasikan objek atau konsep yang datanya akan disimpan dalam basis data, seperti pengguna, produk atau transaksi. Setiap entitas memiliki atribut yang berfungsi untuk mendeskripsikan karakteristik atau

informasi yang dimiliki oleh entitas tersebut misalnya nama, alamat atau tanggal. Relasi digunakan untuk menunjukkan hubungan yang terjadi antar entitas, sedangkan kardinalitas menjelaskan tingkat atau jumlah keterkaitan antar entitas seperti hubungan satu-ke-satu (*one-to-one*), satu-ke-banyak (*one-to-many*) maupun banyak-ke-banyak (*many-to-many*). Selain itu, beberapa notasi ERD juga membedakan atribut berdasarkan fungsinya seperti *primary key* sebagai pengenal unik setiap entitas dan *foreign key* sebagai atribut yang digunakan untuk membentuk hubungan antar entitas dalam basis data relasional.

2.3.4 Encryption at Rest

Encryption at Rest merupakan mekanisme keamanan yang digunakan untuk melindungi data yang disimpan dalam media penyimpanan seperti basis data, server, atau perangkat penyimpanan lainnya [61]. Pada kondisi ini, data tidak sedang diproses atau ditransmisikan namun tetap memiliki resiko terhadap akses tidak sah, baik melalui serangan siber maupun akses fisik terhadap sistem penyimpanan. Penerapan *Encryption at Rest* dilakukan dengan cara mengenkripsi data sebelum data tersebut disimpan sehingga data yang tersimpan berada dalam bentuk *ciphertext* [62]. Proses dekripsi hanya dapat dilakukan ketika data akan diakses oleh pihak yang memiliki kunci enkripsi yang sah. Dengan demikian, meskipun pihak yang tidak berwenang berhasil mengakses media penyimpanan, data yang diperoleh tidak dapat dibaca atau dimanfaatkan. *Encryption at Rest* berperan penting dalam melindungi data sensitif seperti data pengguna, informasi pribadi dan data sistem lainnya [63]. Penerapan mekanisme ini membantu organisasi dalam memenuhi standar keamanan dan regulasi perlindungan data. Selain itu, *Encryption at Rest* juga menjadi lapisan keamanan tambahan yang melengkapi mekanisme keamanan lainnya dalam sistem informasi.

2.3.5 Encryption in Transit

Encryption in transit merupakan mekanisme keamanan yang digunakan untuk melindungi data selama proses pengiriman dari satu sistem ke sistem lainnya melalui jaringan [61]. Mekanisme ini bertujuan untuk menjaga

kerahasiaan, integritas, dan keaslian data selama proses transmisi sehingga data tidak dapat disadap, diubah, atau dimanipulasi oleh pihak yang tidak berwenang. *Encryption in transit* biasanya umumnya diimplementasikan menggunakan protokol keamanan seperti *Transport Layer Security* (TLS) [64]. Protokol ini bekerja dengan mengenkripsi data yang dikirimkan antara klien dan server, sehingga data tersebut hanya dapat dibaca oleh pihak yang terlibat dalam komunikasi. Selain itu, TLS juga menyediakan mekanisme autentikasi untuk memastikan identitas pihak yang berkomunikasi [63]. Dengan penerapan , sistem informasi dapat menjamin bahwa data yang dikirimkan tetap aman meskipun melewati jaringan publik. Mekanisme ini sangat penting dalam sistem yang melibatkan pertukaran data sensitif, seperti sistem berbasis web, aplikasi layanan publik, dan sistem informasi terdistribusi. *Encryption in transit* menjadi komponen utama dalam menjaga keamanan komunikasi antar sistem.

2.3.6 Role Based Access Control

Role Based Access Control (RBAC) merupakan model kontrol akses yang memberikan hak akses kepada pengguna berdasarkan peran (role) yang dimilikinya dalam suatu sistem [53]. Dalam model ini, pengguna tidak diberikan hak akses secara langsung, melainkan melalui peran yang telah ditentukan sesuai dengan tanggung jawab dan fungsi pengguna tersebut. Pendekatan ini memudahkan pengelolaan hak akses dalam sistem yang memiliki banyak pengguna [52]. RBAC memungkinkan administrator sistem untuk mengelompokkan pengguna dengan peran yang sama seperti administrator, operator atau pengguna umum. Setiap peran memiliki sekumpulan hak akses tertentu terhadap sumber daya sistem. Dengan demikian, perubahan hak akses dapat dilakukan dengan lebih efisien cukup dengan mengubah definisi peran tanpa harus mengatur ulang hak akses setiap pengguna secara individual. Penerapan RBAC mendukung prinsip least privilege, di mana pengguna hanya diberikan hak akses yang diperlukan untuk menjalankan tugasnya. Model ini juga meningkatkan keamanan sistem dengan meminimalkan risiko akses yang tidak sah serta memudahkan proses audit dan pemantauan aktivitas pengguna. Oleh karena itu, RBAC banyak digunakan

dalam sistem informasi yang membutuhkan pengelolaan akses yang terstruktur dan aman.

2.3.7 Row Level Security

Row Level Security merupakan mekanisme keamanan pada basis data yang digunakan untuk membatasi akses pengguna terhadap baris data tertentu dalam sebuah tabel [65]. Dengan *Row Level Security*, pengguna yang berbeda dapat melihat atau mengakses data yang berbeda meskipun menggunakan tabel dan query yang sama. Mekanisme ini memungkinkan kontrol akses yang lebih granular dibandingkan kontrol akses pada tingkat tabel. Penerapan *Row Level Security* dilakukan dengan mendefinisikan kebijakan (*policy*) yang menentukan baris data mana yang dapat diakses oleh pengguna berdasarkan kondisi tertentu, seperti peran, identitas pengguna, atau atribut lainnya [66]. Kebijakan ini secara otomatis diterapkan oleh sistem basis data tanpa memerlukan perubahan pada logika aplikasi, sehingga meningkatkan konsistensi dan keamanan data. *Row Level Security* sangat berguna dalam sistem yang melibatkan data multi-pengguna atau multi-unit, di mana setiap pengguna hanya berhak mengakses data miliknya sendiri atau data yang sesuai dengan kewenangannya. Dengan mekanisme ini, risiko kebocoran data antar pengguna dapat diminimalkan, serta integritas dan kerahasiaan data dalam basis data dapat lebih terjaga.

2.3.8 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma kriptografi simetris yang digunakan untuk melindungi kerahasiaan data melalui proses enkripsi dan dekripsi menggunakan kunci yang sama [67]. AES dikembangkan sebagai standar enkripsi oleh *National Institute of Standards and Technology* (NIST) pada tahun 2001 untuk menggantikan *Data Encryption Standard* (DES) yang dinilai sudah tidak lagi memadai dari sisi keamanan [68]. AES didasarkan pada algoritma Rijndael yang dirancang oleh Joan Daemen dan Vincent Rijmen dan sejak diresmikan sebagai standar, AES telah menjadi salah satu metode enkripsi paling banyak digunakan dalam berbagai sistem keamanan informasi termasuk sistem perbankan, pemerintahan dan layanan kesehatan [69].

AES bekerja dengan metode kriptografi blok (*block cipher*), yaitu mengenkripsi data dalam blok berukuran tetap 128 bit. Algoritma ini mendukung tiga variasi panjang kunci, yaitu 128-bit, 192-bit, dan 256-bit, yang masing-masing menentukan jumlah putaran (*round*) dalam proses enkripsi [70]. Berdasarkan pada tabel 2.1, AES-128 menggunakan 10 putaran, AES-192 menggunakan 12 putaran, dan AES-256 menggunakan 14 putaran. Semakin panjang ukuran kunci yang digunakan, semakin tinggi tingkat keamanan yang diberikan karena semakin kompleks proses transformasi kriptografinya. Dalam implementasi sistem yang memerlukan perlindungan tingkat tinggi, seperti sistem informasi kesehatan, AES-256 sering dipilih karena memiliki tingkat ketahanan yang lebih kuat terhadap serangan kriptanalisis modern.

Tabel 2.2 Tabel jumlah *key* dan *round AES Encryption* [70]

Type	Key Length	Block Length	Number of Rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Dalam implementasinya, AES memiliki beberapa tahapan yang dilalui secara berurutan dalam bentuk *round* seperti yang ada pada gambar 2. , yaitu sebagai berikut:

1. *Add Round Key*

   Pada tahap ini, data *ciphertext* akan dikombinasikan dengan kunci enkripsi menggunakan operasi XOR. Kunci yang digunakan berbeda pada setiap *round* karena dihasilkan melalui proses *key expansion*.
2. *Subbytes*

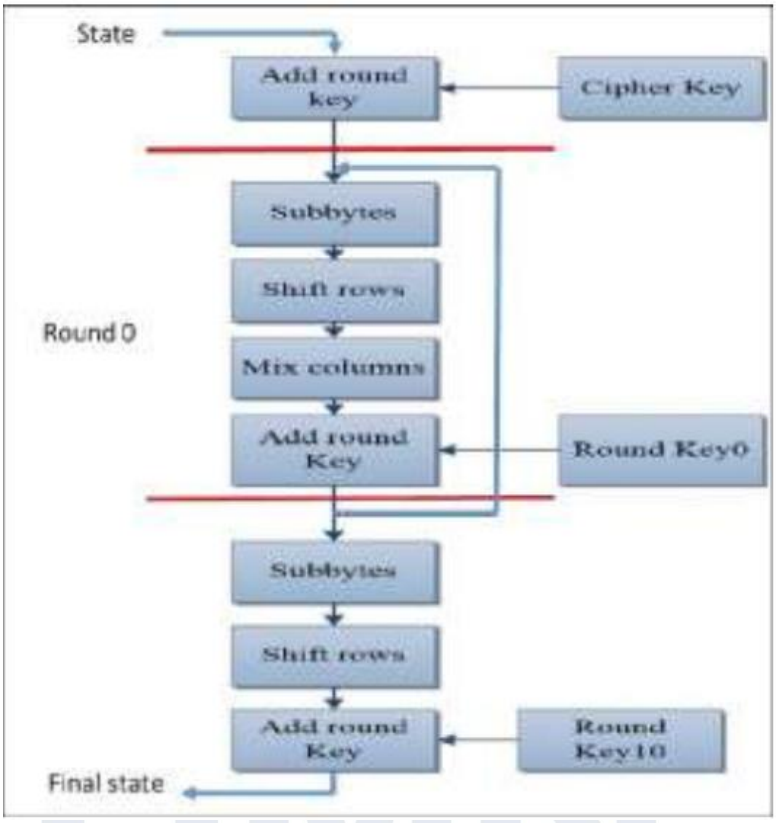
   Subbytes adalah sebuah tahapan subtitusi dimana setiap byte pada data digantikan dengan nilai lain berdasarkan tabel khusus yaitu *Rijndael S-Box*.
3. *Shift Rows*

   Selanjutnya, tahapan Shift Rows merupakan proses penggeseran atau pemindahan blok data ke arah kiri. Tidak ada data yang digeser pada

baris pertama, diikuti dengan pergeseran satu posisi pada baris kedua, dua posisi pada baris ketiga dan tiga posisi pada baris keempat.

4. *Mix Columns*

   Tahapan ini melakukan transformasi pada setiap kolom dalam matriks data dengan menggunakan operasi perkalian yang kemudian akan dimasukkan dalam blok *cipher* baru.



Gambar 2.2 Algoritma AES-128 [71]

Dalam implementasinya pada sistem basis data, penggunaan AES dapat memberikan tambahan beban komputasi karena setiap proses penyimpanan dan pengambilan data memerlukan operasi enkripsi maupun dekripsi. Meskipun demikian, berbagai penelitian menunjukkan bahwa dampak performa yang dihasilkan umumnya masih berada pada tingkat yang dapat diterima. Pada penelitian yang dilakukan oleh Carvalho et al., dilaporkan bahwa penerapan AES pada beberapa platform basis data menghasilkan peningkatan waktu eksekusi query akibat proses kriptografi yang dilakukan selama akses data [62]. Berdasarkan berbagai penelitian yang dirangkum, *overhead* performa yang

dihasilkan umumnya berada pada rentang sekitar 8% hingga 20%, meskipun pada kondisi dan konfigurasi tertentu dapat menghasilkan nilai yang lebih tinggi. Temuan tersebut menunjukkan bahwa penerapan AES memberikan trade-off antara peningkatan keamanan dan performa sistem, namun penurunan performa yang terjadi umumnya masih sebanding dengan peningkatan perlindungan terhadap kerahasiaan data yang tersimpan. Oleh karena itu, evaluasi terhadap dampak implementasi AES perlu dilakukan untuk memastikan bahwa mekanisme keamanan yang diterapkan tetap memenuhi kebutuhan kinerja sistem.

2.3.9 Transport Layer Security (TLS)

Transport Layer Security (TLS) merupakan protokol kriptografi yang dirancang untuk mengamankan komunikasi melalui jaringan komputer dengan menjamin kerahasiaan (*confidentiality*), integritas (*integrity*) dan autentikasi (*authenticity*) data yang ditransmisikan [72]. TLS bekerja pada lapisan transport dan banyak digunakan untuk melindungi pertukaran data antara *client* dan *server*. Protokol ini dikembangkan sebagai penerus *Secure Sockets Layer* (SSL) guna mengatasi berbagai kelemahan keamanan dan menyediakan perlindungan kriptografi yang lebih kuat. TLS mengamankan komunikasi melalui proses yang disebut TLS handshake yaitu tahapan awal ketika klien dan server melakukan proses autentikasi serta membentuk kunci kriptografi bersama untuk sesi komunikasi [64]. TLS dapat mencegah berbagai resiko penyerangan dengan mengenkripsi saluran komunikasi, memverifikasi identitas server menggunakan sertifikat digital yang diterbitkan oleh *Certificate Authority* (CA) terpercaya serta memastikan integritas pesan melalui mekanisme autentikasi pesan [63]. Penerapan TLS yang tepat dapat meningkatkan tingkat keamanan sistem informasi secara keseluruhan dan menjaga kepercayaan pengguna terhadap layanan digital yang digunakan.

2.3.10 STRIDE Threat Modeling

STRIDE merupakan salah satu metode threat modeling yang digunakan untuk mengidentifikasi potensi ancaman keamanan pada suatu sistem secara sistematis [73]. Model ini diperkenalkan oleh Microsoft dan banyak digunakan

dalam proses perancangan sistem keamanan karena mampu mengklasifikasikan ancaman berdasarkan jenisnya. STRIDE merupakan singkatan dari enam kategori ancaman yang mencakup Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service dan Elevation of Privilege. Setiap kategori merepresentasikan jenis serangan yang dapat mengancam keamanan sistem, sebagai berikut [73].

1. *Spoofing*

   Spoofing adalah ancaman yang berkaitan dengan pemalsuan identitas oleh pihak yang tidak berwenang. Proses *spoofing* dapat terjadi ketika pengguna mencoba mengakses *database* dengan menggunakan identitas pengguna lain secara ilegal. Ancaman ini biasanya disebabkan oleh lemahnya mekanisme autentikasi yang ada pada sistem yang dibangun dan tergolong sebagai pencurian identitas (*identity theft*).

2. *Tampering*

   Tampering merujuk pada upaya perubahan data secara tidak sah. *Tampering* dapat dilakukan dengan berbagai cara seperti perubahan terhadap data yang sedang ditransmisikan, manipulasi isi tabel, perubahan data tanpa izin atau modifikasi query yang dapat merusak integritas data yang dapat secara langsung mempengaruhi kinerja dari sistem.

3. *Repudiation*

   Repudiation merupakan tindakan menyangkal aktivitas yang telah dilakukan. Ancaman ini terjadi ketika pengguna melakukan modifikasi, penambahan atau penghapusan data kemudian menyangkal tindakan tersebut. Kondisi ini dapat berdampak signifikan terhadap proses pelacakan (*traceability*) serta pengelolaan tanggung jawab dalam sistem terutama apabila tidak terdapat mekanisme pencatatan (*logging*) yang memadai.

4. *Information Disclosure*

   Information Disclosure adalah ancaman yang berkaitan dengan kebocoran informasi dimana data atau sumber daya dapat diakses oleh pihak yang tidak berwenang. Kerentanan jenis ini berdampak langsung dalam sistem yang mengelola data sensitif seperti data pribadi atau data kesehatan. Ancaman

ini umumnya terjadi akibat lemahnya kontrol akses atau tidak adanya mekanisme enkripsi yang memadai.

5. *Denial of Service*

   Denial of Service merupakan kondisi dimana suatu sistem tidak dapat diakses pada saat dibutuhkan akibat adanya pembatasan atau gangguan pada sumber daya. Hal ini dapat terjadi ketika sistem dibanjiri oleh permintaan dalam jumlah besar dari sumber yang berbahaya atau mengalami *overload* hingga tidak mampu memproses permintaan secara normal.

6. *Elevation of Privilege*

   Elevation of Privilege adalah ancaman yang terjadi ketika pengguna dengan hak akses terbatas berhasil meningkatkan hak aksesnya menjadi lebih tinggi. Hal ini biasanya terjadi melalui eksploitasi kelemahan dalam sistem kontrol akses. Ancaman ini dapat menyebabkan pengguna memperoleh kebebasan lebih dalam menjalankan perintah dan mengakses sumber daya tanpa terdeteksi oleh sistem pemantauan.

2.3.11 Benchmarking

Benchmarking merupakan suatu proses sistematis untuk mengevaluasi, mengukur, dan membandingkan kinerja suatu sistem dalam kondisi tertentu guna menilai tingkat performa yang dimiliki [74]. Dalam konteks sistem basis data, benchmarking digunakan untuk mengukur berbagai aspek kinerja seperti kecepatan pemrosesan, efisiensi sistem, skalabilitas serta kemampuan sistem dalam menangani transaksi dan permintaan pengguna di bawah berbagai kondisi beban kerja [75]. Melalui proses *benchmarking*, diperoleh sebuah gambaran yang objektif mengenai performa suatu sistem serta membandingkannya dengan sistem lain berdasarkan metrik yang terukur.

Dalam operasionalnya, sistem basis data menjalankan berbagai jenis operasi dasar seperti *insert*, *delete*, *read* dan *update*, yang masing-masing memiliki karakteristik beban kerja yang berbeda. Variasi beban kerja tersebut dapat memengaruhi tingkat *latency* atau waktu respons sistem dalam memproses permintaan [74]. Oleh karena itu, pelaksanaan benchmarking menjadi penting untuk dilakukan sebelum suatu sistem diimplementasikan ke dalam aplikasi,

sehingga dapat diketahui apakah sistem tersebut mampu menangani beban kerja yang diharapkan secara efisien. Selain itu, *benchmarking* juga berperan sebagai alat evaluasi yang penting dalam memastikan bahwa sistem yang digunakan dapat memenuhi kebutuhan kinerja baik pada kondisi saat ini maupun ketika terjadi peningkatan volume data dan jumlah transaksi di masa mendatang [75]. Hasil dari pengujian *benchmarking* dapat menjadi dasar untuk melakukan perbandingan performa antara berbagai sistem serta memberikan wawasan bagi pengembang dalam menentukan solusi yang paling sesuai dengan kebutuhan aplikasi

2.4 Tools/software yang digunakan

2.4.1 PostgreSQL

PostgreSQL merupakan *Database Management System* (DBMS) relasional berbasis open-source yang dirancang untuk mengelola data secara andal, konsisten dan skalabel [76]. Sistem ini mendukung penyimpanan data dalam jumlah besar dengan struktur skema yang jelas serta menyediakan berbagai mekanisme pengelolaan data yang mendukung integritas dan konsistensi informasi. Sebagai basis data relasional, PostgreSQL menyimpan data dalam bentuk tabel yang saling terhubung melalui relasi antar tabel. Selain itu, PostgreSQL menyediakan berbagai fitur seperti constraint, indexing serta transaction management untuk memastikan bahwa setiap operasi yang dilakukan terhadap data tetap memenuhi aturan yang telah ditetapkan [77]. Dukungan terhadap standar SQL yang kuat serta fleksibilitas dalam pengelolaan data menjadikan PostgreSQL banyak digunakan dalam berbagai sistem informasi, mulai dari aplikasi skala kecil hingga sistem enterprise.

Dalam menjaga keamanan data pada sistem basis data, PostgreSQL menyediakan berbagai mekanisme perlindungan yang dirancang untuk menjaga kerahasiaan, integritas, dan ketersediaan data yang tersimpan di dalam sistem. PostgreSQL memungkinkan pengelolaan hak akses pengguna secara terstruktur sehingga administrator dapat mengatur siapa saja yang berhak mengakses, memodifikasi, atau mengelola data dalam basis data [78]. Mekanisme ini memungkinkan pembatasan akses terhadap objek basis data seperti tabel,

skema, maupun fungsi, sehingga hanya pengguna yang memiliki otorisasi yang dapat melakukan operasi tertentu. Selain itu, PostgreSQL juga mendukung pengaturan kebijakan keamanan yang memungkinkan pembatasan akses data secara lebih spesifik untuk memastikan bahwa data sensitif hanya dapat diakses oleh pihak yang berwenang. Lebih lanjut, PostgreSQL juga dapat diintegrasikan dengan berbagai teknik enkripsi yang bertujuan untuk melindungi data baik selama proses penyimpanan maupun saat proses transmisi [78]. Dengan penerapan mekanisme enkripsi pada media penyimpanan, data yang tersimpan tidak dapat dibaca secara langsung oleh pihak yang tidak memiliki akses yang sah . Kombinasi antara pengelolaan akses pengguna, kebijakan keamanan, serta mekanisme perlindungan data menjadikan PostgreSQL sebagai salah satu sistem basis data yang mampu mendukung kebutuhan keamanan pada berbagai jenis aplikasi yang memproses data sensitif.

Selain berfungsi sebagai basis data relasional tradisional, PostgreSQL juga dapat diperluas untuk mendukung penyimpanan dan pengolahan data berbasis vektor melalui penggunaan ekstensi tambahan. Pendekatan ini memungkinkan PostgreSQL untuk menyimpan representasi numerik dari data seperti teks, gambar, atau dokumen dalam bentuk embedding vector [79]. Data dalam bentuk vektor tersebut dapat digunakan untuk melakukan pencarian berbasis kemiripan (similarity search) yang banyak dimanfaatkan dalam aplikasi berbasis kecerdasan buatan seperti sistem rekomendasi, semantic search maupun chatbot berbasis RAG [45]. Salah satu *extension* yang biasanya sering digunakan untuk mengintegrasikan PostgreSQL sebagai *vector database* adalah PGVector [79]. Ekstensi ini memungkinkan PostgreSQL menyimpan embedding dalam kolom khusus bertipe vektor serta menyediakan berbagai metode perhitungan jarak atau kemiripan antar vektor seperti cosine similarity, Euclidean distance, dan inner product. Dengan dukungan tersebut, PostgreSQL dapat melakukan operasi pencarian kemiripan secara efisien sekaligus tetap mempertahankan kemampuan pengelolaan data relasional yang dimilikinya

Selain mendukung pengelolaan dan keamanan data, PostgreSQL juga menyediakan tools bawaan seperti pgbench yang dapat digunakan untuk

melakukan pengujian performa dan *stress testing* terhadap *database* [77]. pgbench memungkinkan simulasi eksekusi query secara bersamaan dalam jumlah besar untuk mengukur performa database ketika menerima beban akses yang tinggi. Tools ini umum digunakan untuk mengevaluasi *throughput*, *latency* serta kestabilan PostgreSQL pada berbagai skenario pengujian seperti simulasi *concurrent request*, pengujian transaksi maupun evaluasi mekanisme pembatasan *resource* pada *database server*.

2.4.2 OpenSSL

OpenSSL merupakan library kriptografi yang digunakan secara luas untuk mendukung implementasi mekanisme keamanan dalam sistem informasi. OpenSSL menyediakan berbagai fungsi kriptografi, termasuk enkripsi, dekripsi, hashing, serta pengelolaan sertifikat digital. Library ini banyak digunakan karena bersifat *open-source*, stabil, dan mendukung berbagai algoritma kriptografi yang telah teruji keamanannya. Dalam konteks penelitian ini, OpenSSL berperan sebagai pendukung penerapan mekanisme *Encryption at Rest*. Melalui fungsi kriptografi yang disediakan, OpenSSL dapat digunakan untuk mendukung proses pengamanan data sensitif agar tidak dapat dibaca tanpa kunci yang sesuai. Penerapan enkripsi ini bertujuan untuk menjaga kerahasiaan data yang tersimpan serta mengurangi risiko kebocoran data akibat akses tidak sah.

Selain itu, OpenSSL juga berperan penting dalam penerapan *Transfer Level Security* melalui protokol *Transport Layer Security* (TLS). Protokol ini digunakan untuk mengenkripsi data selama proses komunikasi antara client dan server, sehingga data tidak dapat disadap atau dimodifikasi oleh pihak lain selama proses transmisi. TLS juga menyediakan mekanisme autentikasi untuk memastikan identitas pihak yang berkomunikasi. Dengan dukungan OpenSSL, sistem informasi dapat menerapkan standar keamanan komunikasi yang umum digunakan secara global. Penggunaan OpenSSL membantu memastikan bahwa data yang dipertukarkan antar komponen sistem tetap terjaga kerahasiaan dan integritasnya. Sehingga, OpenSSL menjadi teknologi pendukung yang relevan dalam penerapan mekanisme keamanan terkait keamanan komunikasi data.

2.4.3 Wireshark

Wireshark merupakan perangkat lunak analisis jaringan (*network protocol analyzer*) yang digunakan secara luas untuk memantau dan menganalisis *data traffic* pada suatu jaringan komputer [80]. Wireshark memungkinkan pengguna untuk menangkap (*capture*) paket data yang melintas melalui antarmuka jaringan dan menampilkannya secara rinci berdasarkan struktur protokol komunikasi yang digunakan [81]. Wireshark seringkali dimanfaatkan dalam berbagai bidang yang berkaitan dengan analisa jaringan data karena kemampuannya dalam menampilkan *data traffic* secara detail. Dengan kemampuannya dalam melakukan *deep packet inspection*, Wireshark dapat mengidentifikasi berbagai jenis protokol seperti TCP/IP, HTTP, DNS serta SSL/TLS. Wireshark juga dapat berperan sebagai alat analisis dan verifikasi terhadap implementasi mekanisme keamanan komunikasi data [82]. Melalui proses *packet capturing*, Wireshark dapat digunakan untuk mengamati proses komunikasi antara client dan server termasuk mendeteksi apakah komunikasi tersebut telah menggunakan protokol keamanan seperti *Transport Layer Security* (TLS).