

**PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN
MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB**

SKRIPSI

DISUSUN OLEH

CHARISYA SILVARA MALIGAN

NPM. 2209010203



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

**PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN
MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas
Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah
Sumatera Utara**

CHARISYA SILVARA MALIGAN

NPM. 2209010203

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN**

2026

LEMBAR PENGESAHAN

Judul Skripsi : PENGAMANAN DATA PASIEN PADA PUSKESMAS
BATAHAN MENGGUNAKAN ALGORITMA AES-128
BERBASIS WEB
Nama Mahasiswa : Charisya Silvara Maligan
NPM : 2209010203
Program Studi : Sistem Informasi

Menyetujui

Komisi Pembimbing



(Mhd. Basri, S.Si, M.Kom)

NIDN. 0111078802

Ketua Program Studi



(Mahardika Abdi Prawira Tanjung, M.Kom)
NIDN. 0117088902

Dekan



(Dr. Al Khoyarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN
MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Oktober 2026

Yang membuat
pernyataan



Charisya Silvara Maligan
NPM. 2209010203

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Charisya Silvara Maligan
NPM : 2209010203
Program Studi : Sistem Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

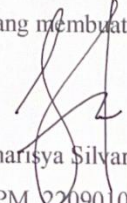
**PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN
MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Mei 2026

Yang membuat pernyataan


Charisya Silvara Maligan

NPM. 2209010203

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Charisya Silvara Maligan
Tempat dan Tanggal Lahir	: Batahan, 17 Februari 2005
Alamat Rumah	: Desa Sari Kenanga, Batahan, Mandailing Natal
Telepon/Faks/HP	: 082183307372
E-mail	: charisyamaligan@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SD NEGERI 338 BATAHAN	TAMAT: 2016
SMP	: MTS MUHAMMADIYAH 11 BATAHAN	TAMAT: 2019
SMA	: SMA NEGERI 1 BATAHAN	TAMAT: 2022

KATA PENGANTAR



Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Besar Muhammad SAW yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh dengan ilmu pengetahuan seperti sekarang ini. Dalam penyusunan skripsi ini, penulis menyadari bahwa masih terdapat berbagai kekurangan. Namun demikian, skripsi ini dapat diselesaikan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi di Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) Universitas Muhammadiyah Sumatera Utara. Skripsi ini berjudul “Pengamanan Data Pasien pada Puskesmas Batahan Menggunakan Algoritma AES-128 Berbasis Web” dan dapat diselesaikan sesuai dengan harapan penulis.

Selama proses penyusunan skripsi ini, penulis menghadapi berbagai kesulitan dan tantangan. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah memberikan dukungan, bantuan, serta doa sehingga skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd., selaku Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).

2. Bapak Assoc. Prof. Dr. Al-Khowarizmi, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) Universitas Muhammadiyah Sumatera Utara.
3. Ibu Dr. Firaumi Rizky, S.Kom., M.Kom., selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) Universitas Muhammadiyah Sumatera Utara.
4. Bapak Mhd. Basri, S.Si., M.Kom., selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) Universitas Muhammadiyah Sumatera Utara sekaligus dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, arahan, masukan, serta motivasi kepada penulis selama proses penyusunan hingga penyelesaian skripsi ini.
5. Bapak Mahardika Abdi Prawira Tanjung, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Informasi.
6. Bapak Mulkan Azhari, S.Kom., M.Kom., selaku Sekretaris Program Studi Sistem Informasi.
7. Kepada Umak dan Abak tercinta, sosok yang selalu menjadi sumber kekuatan, doa, dan semangat bagi penulis dalam menjalani setiap proses kehidupan. Terima kasih atas kasih sayang yang tidak pernah berkurang, doa yang tidak pernah putus, serta segala pengorbanan yang telah diberikan kepada penulis selama menempuh pendidikan hingga dapat menyelesaikan skripsi ini. Setiap pencapaian yang berhasil penulis raih tidak terlepas dari doa, dukungan, serta perjuangan Umak dan Abak yang mungkin tidak akan pernah mampu penulis balas sepenuhnya. Terima kasih telah mengusahakan segala hal, baik dalam bentuk dukungan materi, perhatian, maupun pengorbanan yang mungkin sering kali tidak penulis sadari. Segala kelelahan dan perjuangan Umak dan Abak menjadi alasan bagi penulis untuk terus berusaha dan tidak menyerah ketika menghadapi berbagai kesulitan selama proses perkuliahan dan penyusunan skripsi. Semoga Allah SWT senantiasa memberikan kesehatan, kebahagiaan, keberkahan, serta umur yang

panjang kepada Umak dan Abak, sehingga penulis masih diberikan kesempatan untuk membalas sedikit demi sedikit segala kebaikan dan pengorbanan yang telah diberikan. Gelar yang penulis peroleh hari ini bukan hanya menjadi pencapaian bagi penulis, tetapi juga merupakan bagian dari doa dan harapan Umak dan Abak yang akhirnya dapat terwujud.

8. Kepada kakak dan kanda penulis, terima kasih atas segala doa, dukungan, perhatian, bantuan, serta semangat yang telah diberikan selama penulis menjalani perkuliahan hingga menyelesaikan skripsi ini. Terima kasih telah menjadi tempat berbagi cerita, memberikan dorongan ketika penulis menghadapi kesulitan, serta selalu memberikan dukungan dalam setiap langkah yang penulis jalani. Terima kasih juga atas segala bantuan dan kepedulian yang diberikan selama proses pendidikan penulis, baik dalam hal kecil maupun besar. Kehadiran kakak dan kanda menjadi salah satu bagian penting dalam perjalanan penulis hingga dapat mencapai tahap ini. Semoga segala kebaikan dan dukungan yang telah diberikan dapat menjadi hal yang selalu penulis ingat dan syukuri.
9. Kepada teman-teman terkasih terima kasih atas kebersamaan, dukungan, bantuan, dan semangat yang telah diberikan selama masa perkuliahan hingga proses penyusunan skripsi ini. Terima kasih atas setiap cerita, tawa, serta bantuan yang membuat perjalanan perkuliahan menjadi lebih ringan dan bermakna. Setiap kenangan yang telah dilalui bersama akan selalu menjadi bagian yang tidak terlupakan. Semoga persahabatan dan hubungan baik yang telah terjalin tetap terjaga, serta kita semua dapat meraih kesuksesan dan kebahagiaan di masa yang akan datang.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan masih terdapat kekurangan dalam penyusunan maupun pembahasannya. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi penyempurnaan skripsi ini di masa yang akan datang.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca dan dapat menjadi referensi bagi penelitian selanjutnya, khususnya dalam bidang keamanan data dan sistem informasi.

“PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB”

ABSTRAK

Data identitas dan rekam medis pasien di Puskesmas Batahan masih tersimpan dalam bentuk yang dapat dibaca langsung, sehingga berpotensi diakses oleh pihak yang tidak berwenang. Penelitian ini bertujuan menerapkan algoritma Advanced Encryption Standard 128-bit (AES-128) untuk mengamankan data pasien pada sistem informasi berbasis web. Penelitian ini merupakan penelitian rekayasa perangkat lunak yang dikembangkan dengan model Waterfall, menggunakan Next.js, React, Prisma, dan MySQL 8.0. Algoritma AES-128 diterapkan pada mode Cipher Block Chaining (CBC) dengan padding PKCS#7 dan Initialization Vector acak untuk mengenkripsi atribut sensitif pasien dan rekam medis, dilengkapi kontrol akses berbasis peran dan pencatatan jejak audit. Hasil pengujian menunjukkan seluruh 38 skenario black box berhasil dengan tingkat keberhasilan 100%, ciphertext yang dihasilkan cocok persis dengan vektor uji resmi NIST FIPS PUB 197, dan nilai avalanche effect rata-rata mencapai 50,70% sehingga memenuhi sifat difusi. Penambahan waktu akibat enkripsi hanya 0,53 milidetik untuk penyimpanan satu data pasien, jauh di bawah batas dua detik yang ditetapkan. Dengan demikian, penerapan AES-128 terbukti meningkatkan keamanan data pasien tanpa mengorbankan kenyamanan penggunaan sistem.

Kata Kunci: AES-128, Kriptografi, Keamanan Data, Rekam Medis, Sistem Informasi Berbasis Web

**“WEB-BASED PATIENT DATA SECURITY AT BATAHAN COMMUNITY
HEALTH CENTER USING THE AES-128 ALGORITHM”**

ABSTRACT

Patient identity and medical record data at Batahan Community Health Center are still stored in a directly readable form, creating the risk of access by unauthorized parties. This study aims to implement the Advanced Encryption Standard 128-bit (AES-128) algorithm to secure patient data in a web-based information system. This is a software engineering study developed using the Waterfall model with Next.js, React, Prisma, and MySQL 8.0. AES-128 was applied in Cipher Block Chaining (CBC) mode with PKCS#7 padding and a random Initialization Vector to encrypt sensitive patient and medical record attributes, complemented by role-based access control and audit trail logging. Testing results show that all 38 black box scenarios passed with a 100% success rate, the resulting ciphertext matched exactly the official NIST FIPS PUB 197 test vector, and the average avalanche effect reached 50.70%, satisfying the diffusion property. The additional processing time caused by encryption was only 0.53 milliseconds per patient record, far below the defined two-second threshold. Thus, the implementation of AES-128 is proven to improve patient data security without sacrificing system usability.

Keywords: *AES-128, Cryptography, Data Security, Medical Records, Web-Based Information System*

DAFTAR ISI

ABSTRAK	x
ABSTRACT	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI	6
2.1 Sistem Informasi	6
2.2 Sistem Informasi Kesehatan.....	7
2.3 Data	8
2.4 Rekam Medis / Data Pasien	9
2.5 Keamanan Informasi	10
2.6 Kriptografi.....	11
2.6.1 Kriptografi Kunci Simetris.....	11
2.6.2 Kriptografi Kunci Asimetris.....	12
2.6.3 Enkripsi dan Dekripsi.....	13
2.6.4 Plaintext dan Ciphertext.....	14
2.6.5 Kunci Kriptografi (Key).....	15
2.9 Algoritma Advanced Encryption Standard (AES) 128	16
2.9.1 Key Expansion	16
2.9.2 Tahapan Transformasi AES.....	17
2.9.3 Alur Proses Enkripsi dan Dekripsi AES-128	19
2.10 Database	20
2.11 Keamanan Database	20
2.12 Sistem Berbasis Web	21
2.13 Autentikasi dan Otorisasi	21

2.14 Penelitian Terdahulu.....	22
BAB III ANALISA DAN PERANCANGAN SISTEM.....	26
3.1 Jenis Penelitian.....	26
3.2 Lokasi dan Waktu Penelitian.....	26
3.3 Metode Pengumpulan Data	27
3.3.1 Observasi.....	27
3.3.2 Wawancara	27
3.3.3 Studi Literatur	27
3.4 Analisis Sistem yang Sedang Berjalan.....	28
3.5 Analisis Kebutuhan Sistem	28
3.5.1 Kebutuhan Fungsional	28
3.5.2 Kebutuhan Non-Fungsional	29
3.5.3 Spesifikasi Perangkat yang Digunakan.....	29
3.6 Metode Pengembangan Sistem	30
3.6.1 Analisis Kebutuhan	30
3.6.2 Perancangan Sistem	30
3.6.3 Implementasi	30
3.6.4 Pengujian.....	31
3.6.5 Pemeliharaan	31
3.7 Perancangan Sistem	31
3.7.1 Use Case Diagram.....	31
3.7.2 Activity Diagram.....	32
3.7.3 Flowchart Proses Enkripsi dan Dekripsi AES-128	33
3.7.4 Entity Relationship Diagram (ERD)	35
3.8 Perancangan Database.....	36
3.9 Perancangan Antarmuka (Interface).....	39
3.9.1 Rancangan Halaman Login.....	39
3.9.2 Rancangan Halaman Manajemen Data Pasien.....	40
3.9.3 Rancangan Halaman Input Rekam Medis.....	41
3.10 Pengujian Sistem.....	41
3.10.1 Black Box Testing	41
3.10.2 Pengujian Enkripsi dan Dekripsi AES-128	43
3.10.3 Pengujian Performa.....	44
BAB IV HASIL DAN PEMBAHASAN.....	46
4.1 Hasil Pengumpulan Data dan Analisis Kebutuhan.....	46

4.1.1	Gambaran Umum Puskesmas Batahan	46
4.1.2	Hasil Analisis Sistem yang Sedang Berjalan	48
4.1.3	Identifikasi Permasalahan Keamanan	49
4.1.4	Kebutuhan Fungsional dan Non-Fungsional.....	51
4.2	Implementasi Lingkungan Sistem.....	54
4.2.1	Struktur Direktori Aplikasi.....	55
4.2.4	Konfigurasi Kunci dan Environment	56
4.3	Implementasi Basis Data.....	57
4.3.1	Struktur Tabel.....	57
4.3.2	Relasi Antar Tabel	60
4.3.3	Penentuan Atribut yang Dienkripsi	60
4.3.4	Perbandingan Data Sebelum dan Sesudah Enkripsi	62
4.4	Implementasi Algoritma AES-128	63
4.4.1	Skema Umum Proses Pengamanan Data	64
4.4.2	Pembangkitan Kunci (Key Expansion).....	65
4.4.3	Proses Enkripsi.....	68
4.4.4	Proses Dekripsi	74
4.4.5	Penerapan Mode Operasi CBC	77
4.4.6	Potongan Kode Implementasi	82
4.5	Implementasi Antarmuka Sistem	87
4.5.1	Halaman Login.....	88
4.5.2	Halaman Dashboard	90
4.5.3	Halaman Manajemen Pengguna.....	91
4.5.4	Halaman Data Pasien	92
4.5.5	Halaman Rekam Medis	94
4.5.6	Halaman Log Aktivitas	95
4.5.7	Halaman Demo Enkripsi	96
4.6	Pengujian Sistem.....	97
4.6.1	Pengujian Black Box.....	97
4.6.2	Pengujian Validitas Algoritma terhadap NIST FIPS 197	103
4.6.3	Pengujian Avalanche Effect	107
4.6.4	Pengujian Performa.....	110
4.6.5	Pengujian Keamanan.....	113
4.7	Pembahasan.....	115
4.7.1	Analisis Hasil Pengujian	116
4.7.2	Analisis Keamanan Sistem.....	117
4.7.3	Perbandingan dengan Penelitian Terdahulu	118
4.7.4	Kelebihan dan Keterbatasan Sistem.....	120

BAB V KESIMPULAN DAN SARAN	123
5.1 Kesimpulan	123
5.2 Saran.....	124
DAFTAR PUSTAKA	127

DAFTAR GAMBAR

Gambar 2.1	Komponen Sistem Informasi	7
Gambar 2.2	Cakupan Sistem Informasi Kesehatan	8
Gambar 2.3	Transformasi Data menjadi Informasi.....	8
Gambar 2.4	Komponen Rekam Medis Pasien	9
Gambar 2.5	Proses Enkripsi dan Dekripsi pada Kriptografi.....	11
Gambar 2.6	Perbandingan Kriptografi Kunci Simetris dan Asimetris.....	13
Gambar 2.7	Transformasi Plaintext menjadi Ciphertext.....	14
Gambar 2.8	Struktur Kunci AES-128 (128 bit / 16 byte)	15
Gambar 2.9	Tabel Forward S-Box AES.....	18
Gambar 2.10	Tabel Inverse S-Box AES.....	18
Gambar 2.11	Proses Enkripsi AES	19
Gambar 2.12	Proses Dekripsi AES	20
Gambar 3.1	Use Case Diagram Sistem.....	31
Gambar 3.2	Activity Diagram Input dan Penyimpanan Data Pasien.....	33
Gambar 3.3	Flowchart Proses Enkripsi dan Dekripsi AES-128	34
Gambar 3.4	Entity Relationship Diagram (ERD) Sistem	36
Gambar 3.5	Rancangan Antarmuka Halaman Login	40
Gambar 3.6	Rancangan Antarmuka Halaman Manajemen Data Pasien.....	40
Gambar 3.7	Rancangan Antarmuka Halaman Input Rekam Medis	41
Gambar 4.1	Struktur Organisasi Puskesmas Batahan.....	47
Gambar 4.2	Flowmap Sistem yang Sedang Berjalan.....	49
Gambar 4.3	Struktur Direktori Aplikasi.....	54
Gambar 4.4	Konfigurasi Kunci AES-128 pada Berkas .env dan lib/aes.ts.....	54
Gambar 4.5	Isi Tabel pasien pada Basis Data MySQL.....	59

Gambar 4.6	Skema Umum Proses Pengamanan Data pada Sistem.....	61
Gambar 4.7	Susunan State Kunci Awal (Round Key 0)	62
Gambar 4.8	Perhitungan Word Keempat (W4) pada Key Expansion.....	63
Gambar 4.9	Susunan State Plaintext Awal	64
Gambar 4.10	Ilustrasi Transformasi ShiftRows pada Putaran 1	66
Gambar 4.11	Perhitungan MixColumns pada Kolom Pertama Putaran 1	67
Gambar 4.12	Perbandingan Skema Mode ECB dan Mode CBC.....	72
Gambar 4.13	Potongan Kode Fungsi encrypt() dan decrypt()	76
Gambar 4.14	Potongan Kode Fungsi encryptRecord() dan decryptRecord()	77
Gambar 4.15	Potongan Kode Penyimpanan Data Pasien dengan Enkripsi AES-128	78
Gambar 4.16	Potongan Kode Kontrol Akses Berbasis Peran	79
Gambar 4.17	Tampilan Halaman Login.....	81
Gambar 4.18	Penanganan Kesalahan pada Proses Login	81
Gambar 4.19	Tampilan Dashboard Peran Admin	82
Gambar 4.20	Tampilan Dashboard Peran Petugas	83
Gambar 4.21	Tampilan Halaman Manajemen Pengguna.....	83
Gambar 4.22	Tampilan Halaman Daftar Data Pasien	84
Gambar 4.23	Tampilan Formulir Pendaftaran Pasien Baru	85
Gambar 4.24	Tampilan Halaman Detail Data Pasien	85
Gambar 4.25	Tampilan Halaman Daftar Rekam Medis.....	86
Gambar 4.26	Tampilan Formulir Input Rekam Medis.....	87
Gambar 4.27	Tampilan Halaman Log Aktivitas	87
Gambar 4.28	Tampilan Halaman Demo Enkripsi AES-128	88
Gambar 4.29	Potongan Kode Pengujian Vektor Uji NIST FIPS 197	92

Gambar 4.30	Grafik Hasil Pengujian Avalanche Effect.....	95
Gambar 4.31	Grafik Panjang Data terhadap Waktu Proses	97
Gambar 4.32	Hasil Pengujian Dekripsi dengan Kunci yang Salah.....	99

DAFTAR TABEL

Tabel 2.1	Perbandingan Kriptografi Kunci Simetris dan Asimetris.....	13
Tabel 2.2	Perbandingan Varian AES	16
Tabel 2.3	Ringkasan Penelitian Terdahulu.....	22
Tabel 3.1	Jadwal Pelaksanaan Penelitian.....	26
Tabel 3.2	Spesifikasi Perangkat Keras	29
Tabel 3.3	Spesifikasi Perangkat Lunak	29
Tabel 3.4	Deskripsi Use Case Sistem	32
Tabel 3.5	Struktur Tabel users.....	37
Tabel 3.6	Struktur Tabel pasien.....	37
Tabel 3.7	Struktur Tabel rekam_medis	38
Tabel 3.8	Struktur Tabel audit_log.....	39
Tabel 3.9	Contoh Skenario Pengujian Black Box.....	42
Tabel 3.10	Format Pengukuran Waktu Eksekusi	45
Tabel 4.1	Identifikasi Permasalahan dan Risiko Keamanan	50
Tabel 4.2	Kebutuhan Fungsional Sistem	51
Tabel 4.3	Kebutuhan Non-Fungsional Sistem	52
Tabel 4.4	Struktur Tabel users.....	55
Tabel 4.5	Struktur Tabel pasien.....	55
Tabel 4.6	Struktur Tabel rekam_medis	56
Tabel 4.7	Struktur Tabel audit_log.....	56
Tabel 4.8	Penentuan Atribut yang Dienkripsi	57
Tabel 4.9	Perbandingan Data Sebelum dan Sesudah Enkripsi.....	59
Tabel 4.10	Hasil Key Expansion: Round Key 0 sampai Round Key 10.....	63
Tabel 4.11	Hasil AddRoundKey Putaran 0	64

Tabel 4.12	Hasil Transformasi SubBytes pada Putaran 1	65
Tabel 4.13	Hasil AddRoundKey Putaran 1	67
Tabel 4.14	State Hasil Setiap Putaran pada Proses Enkripsi	68
Tabel 4.15	Hasil AddRoundKey Awal pada Proses Dekripsi	69
Tabel 4.16	State Hasil Setiap Putaran pada Proses Dekripsi	70
Tabel 4.17	Perbandingan Ciphertext Mode ECB dan Mode CBC.....	73
Tabel 4.18	Contoh Penerapan Padding PKCS#7 pada Data Pasien.....	74
Tabel 4.19	Daftar Halaman Sistem dan Hak Aksesnya	80
Tabel 4.20	Hasil Pengujian Black Box	89
Tabel 4.21	Hasil Pengujian terhadap Vektor Uji NIST FIPS 197	91
Tabel 4.22	Hasil Pengujian Bolak-balik Enkripsi dan Dekripsi	93
Tabel 4.23	Hasil Pengujian Avalanche Effect — Perubahan 1 Bit Plaintext ...	94
Tabel 4.24	Hasil Pengujian Avalanche Effect — Perubahan 1 Bit Kunci	94
Tabel 4.25	Waktu Enkripsi dan Dekripsi Berdasarkan Panjang Data	96
Tabel 4.26	Waktu Proses pada Tingkat Record Data Pasien.....	97
Tabel 4.27	Hasil Pengujian Keamanan Sistem	98
Tabel 4.28	Perbandingan dengan Penelitian Terdahulu	102
Tabel 4.29	Kelebihan dan Keterbatasan Sistem.....	103

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi mendorong instansi pelayanan kesehatan, termasuk puskesmas, untuk memanfaatkan sistem informasi berbasis web dalam pengelolaan data pasien. Sistem ini diharapkan dapat meningkatkan efisiensi pelayanan serta menjamin keamanan dan kerahasiaan data pasien yang bersifat sensitif, seperti identitas pribadi dan riwayat medis.

Namun, pada kenyataannya di Puskesmas Batahan, pengelolaan data pasien berbasis web belum sepenuhnya didukung oleh mekanisme pengamanan yang memadai. Data pasien masih berpotensi diakses oleh pihak yang tidak berwenang, terutama pada proses penyimpanan dan pertukaran data, sehingga membuka peluang terjadinya kebocoran dan penyalahgunaan informasi. Hal ini sejalan dengan penelitian (Hakim & Margolang, 2025) yang menyatakan bahwa sistem rekam medis digital tanpa pengamanan yang kuat sangat rentan terhadap akses tidak sah.

Kondisi tersebut menunjukkan adanya kesenjangan antara kebutuhan akan sistem informasi kesehatan yang aman dan terpercaya dengan penerapan pengamanan data yang belum optimal. Dampak dari permasalahan ini tidak hanya bersifat teknis, tetapi juga dapat menurunkan kepercayaan masyarakat serta meningkatkan risiko pelanggaran privasi dan permasalahan hukum bagi instansi pengelola.

Urgensi pengamanan data pasien semakin diperkuat dengan berlakunya Undang-Undang Nomor 27 Tahun 2022 tentang Perlindungan Data Pribadi (UU PDP) yang mulai efektif sepenuhnya pada 17 Oktober 2024. Dalam UU PDP, data kesehatan dikategorikan sebagai data pribadi spesifik sebagaimana diatur dalam Pasal 4 ayat (2), sehingga fasilitas kesehatan sebagai pengendali data wajib memastikan pengumpulan, penyimpanan, pengolahan, dan distribusi data pasien

dilakukan sesuai prinsip-prinsip perlindungan (Rospita Adelina Siregar, 2025). Lebih lanjut (Alaikha Annan, 2024) menegaskan bahwa sektor kesehatan merupakan salah satu sektor dengan tingkat risiko penyalahgunaan data tertinggi, sehingga penerapan mekanisme pengamanan teknis seperti enkripsi menjadi kewajiban yuridis, bukan sekadar pilihan teknis. Hal ini menempatkan Puskesmas Batahan dalam posisi yang menuntut implementasi pengamanan data yang sesuai standar regulasi nasional.

Oleh karena itu, diperlukan solusi pengamanan data yang efektif untuk melindungi kerahasiaan informasi pasien, dan salah satu metode yang dapat diterapkan adalah algoritma kriptografi. Kriptografi merupakan teknik pengamanan data dengan cara mengubah informasi asli (plaintext) menjadi bentuk tersandi (ciphertext) sehingga pihak yang tidak memiliki wewenang tidak akan dapat mengaksesnya. Di dalam sistem informasi, kriptografi berfungsi untuk menjaga kerahasiaan, keutuhan, dan keaslian dari data yang dikirim ataupun disimpan di sistem.

Salah satu algoritma kriptografi modern yang banyak digunakan adalah Advanced Encryption Standard (AES) 128. AES-128 merupakan algoritma kriptografi simetris jenis block cipher yang bekerja dengan mengenkripsi data dalam blok berukuran 128 bit menggunakan kunci sepanjang 128 bit dengan 10 putaran transformasi. Algoritma ini memiliki tingkat keamanan tinggi, performa yang efisien, serta telah ditetapkan sebagai standar enkripsi resmi oleh National Institute of Standards and Technology (NIST) dalam publikasi FIPS PUB 197. Penerapan AES-128 terbukti mampu meningkatkan keamanan database dan menurunkan risiko kebocoran data, sebagaimana ditunjukkan dalam penelitian (Ramadhani et al., 2022).

Oleh karena itu, penerapan algoritma kriptografi sangat penting pada sistem berbasis web karena proses pertukaran data berlangsung melalui jaringan yang rentan terhadap serangan siber. Selain itu, pengamanan data juga harus memenuhi prinsip confidentiality, integrity, dan availability (CIA) guna mencegah akses tidak sah dan menjaga kualitas layanan kesehatan (Puteri et al., 2024).

Berdasarkan permasalahan keamanan data pasien yang dihadapi oleh Puskesmas Batahan dan keunggulan yang dimiliki oleh algoritma AES-128, maka penelitian ini akan merancang dan membangun sistem pengamanan data pasien berbasis web dengan menerapkan algoritma Advanced Encryption Standard (AES) 128. Dengan diterapkannya algoritma AES-128, data pasien di Puskesmas Batahan diharapkan dapat tersimpan dalam bentuk terenkripsi sehingga kerahasiaannya tetap terlindungi meskipun terjadi akses tidak sah terhadap database, sekaligus meningkatkan kepercayaan masyarakat terhadap pelayanan kesehatan di Puskesmas Batahan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan algoritma AES-128 untuk melakukan proses enkripsi dan dekripsi data pasien pada Puskesmas Batahan?
2. Bagaimana merancang sistem keamanan data pasien di Puskesmas Batahan menggunakan algoritma AES-128 berbasis web?
3. Bagaimana meningkatkan keamanan data pasien pada Puskesmas Batahan melalui penerapan algoritma AES-128?

1.3 Batasan Masalah

Agar penelitian ini terarah dan tidak menyimpang dari tujuan awal yang telah ditetapkan, maka ruang lingkup dan batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem dibangun berbasis web menggunakan bahasa pemrograman JavaScript (Node.js dan Express.js) pada sisi backend, serta React.js / Next.js pada sisi frontend.
2. Sistem manajemen basis data yang digunakan adalah MySQL versi 8.0.
3. Implementasi algoritma AES-128 dilakukan menggunakan pustaka kriptografi crypto-js yang mengikuti standar NIST FIPS PUB 197.

4. Data yang diamankan meliputi data identitas pasien dan data rekam medis yang tersimpan dalam database sistem.
5. Penelitian ini tidak membahas keamanan jaringan, keamanan perangkat keras, maupun aspek keamanan fisik di lingkungan puskesmas.
6. Penelitian ini tidak membahas algoritma kriptografi lain selain AES-128.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma AES-128 untuk melakukan proses enkripsi dan dekripsi data identitas dan rekam medis pasien di Puskesmas Batahan, sehingga data tersimpan dalam bentuk tersandi yang tidak dapat dibaca oleh pihak yang tidak berwenang.
2. Merancang sistem keamanan data pasien berbasis web menggunakan algoritma AES-128 yang siap diimplementasikan oleh Puskesmas Batahan untuk menggantikan sistem pengelolaan data pasien yang belum aman.
3. Meningkatkan keamanan data pasien di Puskesmas Batahan melalui penerapan algoritma AES-128, sehingga data identitas dan rekam medis pasien tersimpan dalam bentuk terenkripsi dan tidak dapat dibaca oleh pihak yang tidak memiliki wewenang.

1.5 Manfaat Penelitian

Adanya manfaat dari dilakukannya penelitian ini diharapkan dapat memberikan kontribusi, baik secara teoritis maupun praktis, dalam upaya pengamanan data pasien di Puskesmas Batahan. Adapun manfaat penelitian ini adalah sebagai berikut:

1. Memberikan gambaran mengenai tingkat keamanan data pasien di Puskesmas Batahan setelah penerapan algoritma AES-128, sehingga dapat

diketahui sejauh mana data pasien terlindungi dari ancaman akses yang tidak sah.

2. Meningkatkan keamanan data pasien melalui penerapan algoritma AES-128, sehingga risiko kebocoran dan penyalahgunaan data pasien dapat diminimalkan.
3. Memberikan rasa aman dan kepercayaan dalam menggunakan layanan kesehatan di Puskesmas Batahan, karena data pribadi dan riwayat medis pasien terlindungi dengan baik.
4. Menambah wawasan dan pengalaman dalam mengimplementasikan algoritma kriptografi pada sistem berbasis web, serta menjadi sarana penerapan teori yang telah diperoleh selama perkuliahan dalam bentuk penelitian nyata.

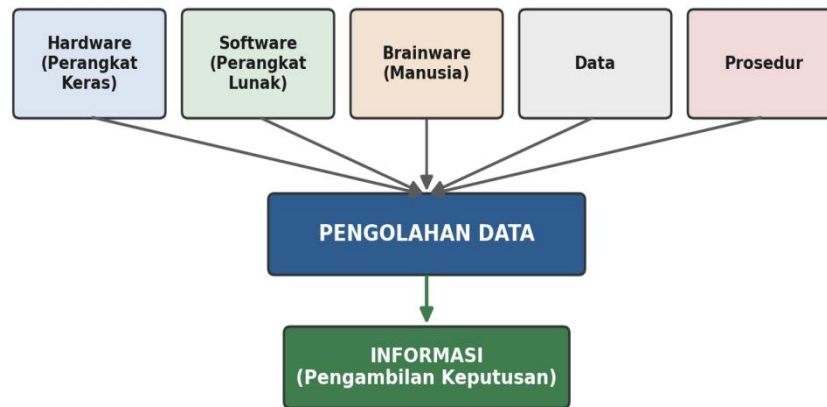
BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari komponen perangkat keras (hardware), perangkat lunak (software), manusia (brainware), data, dan prosedur yang saling berinteraksi untuk mengolah data menjadi informasi yang berguna dalam mendukung pengambilan keputusan. Kelima komponen tersebut bekerja secara terpadu, di mana perangkat keras berfungsi sebagai media pemrosesan dan penyimpanan fisik, perangkat lunak sebagai instruksi yang menjalankan proses pengolahan, manusia sebagai pengguna sekaligus pengelola sistem, data sebagai bahan baku yang diolah, dan prosedur sebagai aturan baku yang mengatur jalannya proses. Sistem informasi berbasis web memungkinkan proses pengolahan dan penyebaran informasi dilakukan secara online melalui jaringan internet, sehingga informasi dapat diakses kapan saja dan dari mana saja sepanjang terhubung dengan jaringan (Bakri et al., 2024).

Informasi yang dihasilkan dari sistem ini digunakan sebagai dasar dalam proses pengambilan keputusan yang tepat, cepat, dan akurat. Dengan adanya sistem informasi, suatu organisasi dapat meningkatkan efisiensi kerja, mengurangi kesalahan dalam pengolahan data, serta meningkatkan kualitas layanan dan kinerja secara keseluruhan. Dalam konteks instansi pelayanan kesehatan seperti puskesmas, sistem informasi berperan penting dalam mempercepat pencatatan data pasien, menyimpan riwayat pelayanan secara terstruktur, dan meminimalkan risiko kehilangan atau duplikasi data yang sebelumnya kerap terjadi pada pencatatan manual. Hubungan antar komponen sistem informasi dalam menghasilkan informasi dapat dilihat pada Gambar 2.1 berikut.

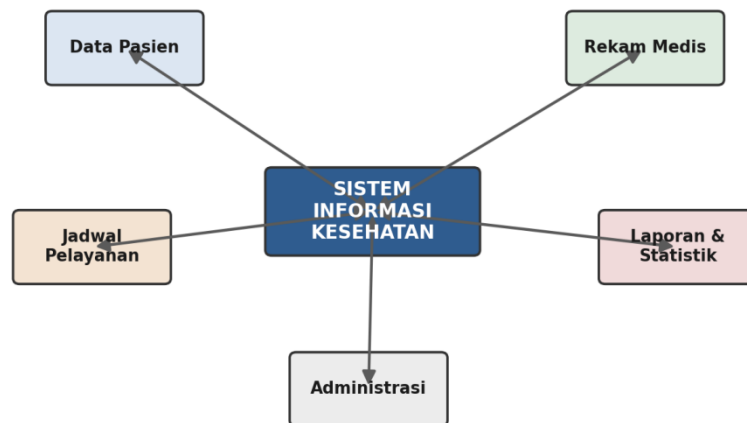


Gambar 2. 1 Komponen Sistem Informasi

2.2 Sistem Informasi Kesehatan

Sistem informasi kesehatan adalah sistem yang digunakan untuk mengelola data dan informasi terkait pelayanan kesehatan, seperti data pasien, rekam medis, jadwal pelayanan, akurasi data, serta kualitas pengambilan keputusan di instansi kesehatan. Sistem ini menjadi tulang punggung operasional fasilitas kesehatan karena seluruh aktivitas pelayanan—mulai dari pendaftaran pasien, pemeriksaan, pencatatan diagnosa, hingga pelaporan—bergantung pada ketersediaan dan keakuratan data yang dikelolanya. Dengan sistem informasi kesehatan yang baik, tenaga medis dapat mengambil keputusan klinis secara lebih cepat dan tepat, sementara pihak manajemen dapat menyusun laporan serta merencanakan layanan secara lebih terukur.

Karena memuat data pribadi dan medis pasien, sistem ini memerlukan mekanisme pengamanan yang kuat untuk melindungi kerahasiaan informasi (Indriyajati et al., 2024). Data kesehatan tergolong sebagai data pribadi yang bersifat spesifik dan sensitif, sehingga apabila terjadi kebocoran dapat menimbulkan kerugian serius bagi pasien, baik dari sisi privasi maupun potensi penyalahgunaan. Oleh sebab itu, penerapan teknik pengamanan seperti enkripsi, autentikasi, dan pembatasan hak akses menjadi kebutuhan mendasar dalam pembangunan sistem informasi kesehatan. Cakupan data dan informasi yang dikelola dalam sistem informasi kesehatan ditunjukkan pada Gambar 2.2 berikut:



Gambar 2. 2 Cakupan Sistem Informasi Kesehatan

2.3 Data

Data merupakan sekumpulan fakta mentah yang belum diolah dan belum memiliki makna yang jelas. Data dapat berupa angka, teks, simbol, gambar, maupun kombinasi dari berbagai bentuk informasi lainnya. Dalam konteks sistem informasi, data menjadi komponen utama yang diproses untuk menghasilkan informasi yang berguna bagi pengambilan keputusan. Data yang masih berdiri sendiri belum dapat memberikan manfaat secara langsung; baru setelah melalui proses pengolahan—seperti pengurutan, pengelompokan, perhitungan, atau penyajian dalam bentuk tertentu—data tersebut berubah menjadi informasi yang memiliki nilai dan makna bagi penggunaannya. Hubungan antara data dan informasi tersebut digambarkan pada Gambar 2.3 berikut.

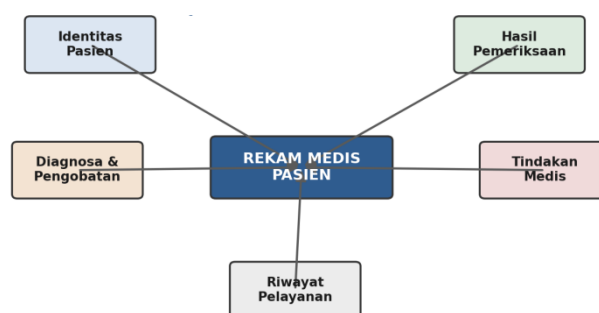


Gambar 2. 3 Transformasi Data menjadi Informasi

Dalam sistem informasi kesehatan, data yang dikelola meliputi identitas pasien, riwayat pemeriksaan, diagnosa, tindakan medis, serta hasil laboratorium. Data tersebut bersifat sensitif karena berkaitan langsung dengan informasi pribadi dan kondisi kesehatan seseorang, sehingga pengelolaan data harus dilakukan secara sistematis serta dilengkapi dengan mekanisme pengamanan untuk mencegah akses, perubahan, atau penyalahgunaan oleh pihak yang tidak berwenang (Indriyajati et al., 2024). Pengamanan data sejak tahap penyimpanan menjadi sangat penting, mengingat data kesehatan merupakan aset yang dilindungi secara hukum dan menjadi dasar kepercayaan pasien terhadap fasilitas pelayanan kesehatan.

2.4 Rekam Medis / Data Pasien

Rekam medis adalah berkas yang berisi catatan dan dokumen mengenai identitas pasien, pemeriksaan, pengobatan, tindakan, serta pelayanan lain yang telah diberikan kepada pasien. Rekam medis dapat berbentuk fisik maupun digital dalam sistem informasi kesehatan (Puteri et al., 2024). Sebagai dokumen yang merekam seluruh perjalanan pelayanan kesehatan seorang pasien, rekam medis memiliki fungsi penting baik dari aspek administratif, hukum, finansial, maupun penelitian. Kelengkapan dan keakuratan rekam medis sangat menentukan kualitas pelayanan, karena menjadi rujukan utama tenaga medis dalam mengambil keputusan klinis pada kunjungan berikutnya. Komponen-komponen yang umumnya terdapat dalam rekam medis pasien ditunjukkan pada Gambar 2.4 berikut.



Gambar 2. 4 Komponen Rekam Medis Pasien

Data dalam rekam medis bersifat sangat rahasia karena mengandung informasi pribadi dan kondisi kesehatan pasien. Dalam era digital, rekam medis elektronik memerlukan sistem pengamanan yang kuat, seperti penggunaan enkripsi, untuk mencegah kebocoran data serta menjaga privasi pasien (Hakim & Margolang, 2025). Penyimpanan rekam medis dalam bentuk digital memang meningkatkan efisiensi dan kemudahan akses, namun di sisi lain memunculkan risiko keamanan baru apabila tidak dilindungi dengan baik, seperti pencurian data, akses tidak sah, maupun manipulasi data. Oleh karena itu, penerapan algoritma kriptografi seperti AES-128 pada penyimpanan rekam medis menjadi salah satu solusi yang efektif untuk menjamin kerahasiaan dan integritas data pasien.

2.5 Keamanan Informasi

Keamanan informasi adalah upaya untuk melindungi data dari akses, penggunaan, pengungkapan, perubahan, dan penghancuran yang tidak sah. Dalam sistem informasi, keamanan bertujuan menjaga informasi agar tetap aman dan hanya dapat diakses oleh pihak yang memiliki wewenang. Konsep dasar keamanan informasi dikenal dengan prinsip CIA (Confidentiality, Integrity, Availability) (Puteri et al., 2024), yaitu:

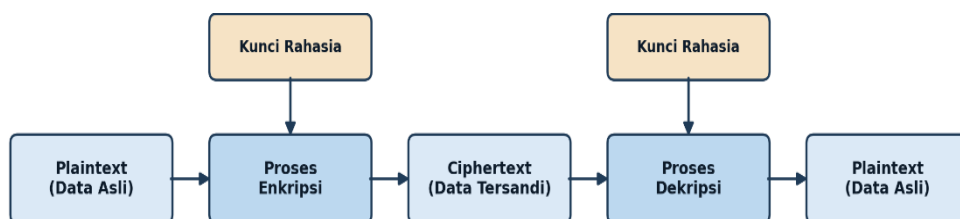
1. Confidentiality (Kerahasiaan) – Menjamin bahwa informasi terlindungi dan hanya dapat diakses oleh pihak yang berwenang.
2. Integrity (Integritas) – Menjamin bahwa data tidak dapat diubah oleh pihak yang tidak memiliki wewenang, menjaga keakuratan dan integritas data.
3. Availability (Ketersediaan) – Menjamin bahwa data dapat diakses saat dibutuhkan oleh pihak yang berhak.

Penerapan prinsip CIA pada sistem informasi kesehatan menjadi semakin krusial mengingat tingginya ancaman siber yang menyasar fasilitas kesehatan. (Vaiyapuri et al., 2021) dalam penelitiannya pada sistem Medical IoT membuktikan bahwa serangan Man-in-the-Middle (MITM) terhadap perangkat kesehatan dapat dilakukan secara real-time dan mengakibatkan kebocoran data pasien yang masif. Penelitian tersebut juga menunjukkan bahwa implementasi AES-128 sebagai

lightweight encryption mampu memberikan perlindungan yang kuat terhadap serangan tersebut dengan dampak performa yang minimal, sehingga layak diterapkan pada fasilitas kesehatan dengan sumber daya komputasi terbatas seperti puskesmas.

2.6 Kriptografi

Kriptografi merupakan ilmu dan seni untuk menjaga keamanan informasi dengan cara mengubah pesan asli menjadi bentuk tersandi sehingga tidak dapat dipahami oleh pihak yang tidak berwenang. Tujuan utama kriptografi adalah untuk menjamin kerahasiaan (confidentiality), keutuhan (integrity), keaslian (authenticity), dan nirpenyangkalan (non-repudiation) data yang dikirim maupun disimpan dalam suatu sistem informasi (Tamin et al., 2025). Sebelum AES menjadi standar enkripsi modern, algoritma Data Encryption Standard (DES) banyak digunakan untuk pengamanan data pada berbagai sistem, termasuk pada model keamanan cloud computing yang dikombinasikan dengan teknik steganografi Least Significant Bit (LSB). Namun, seiring berkembangnya kebutuhan keamanan dan adanya kelemahan pada panjang kunci DES, AES kemudian ditetapkan sebagai standar enkripsi resmi karena memiliki tingkat keamanan dan efisiensi yang lebih baik.



Gambar 2. 5 Proses Enkripsi dan Dekripsi pada Kriptografi

2.6.1 Kriptografi Kunci Simetris

Kriptografi kunci simetris merupakan metode kriptografi yang menggunakan satu kunci yang sama untuk melakukan proses enkripsi dan dekripsi data. Pada kriptografi kunci simetris, baik pengirim maupun penerima pesan harus

memiliki kunci yang identik dan bersifat rahasia, sehingga distribusi kunci antar-pihak menjadi salah satu tantangan utama dalam implementasinya.

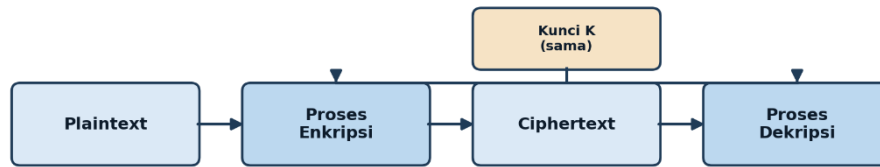
Algoritma kriptografi kunci simetris secara umum dibagi menjadi dua jenis, yaitu Block Cipher yang mengenkripsi data dalam blok berukuran tetap (contoh: AES, DES, dan Blowfish) dan Stream Cipher yang mengenkripsi data bit per bit atau byte per byte secara berurutan (contoh: RC4 dan Salsa20). Karena hanya menggunakan satu kunci, kriptografi simetris umumnya memiliki kinerja yang lebih cepat dibandingkan kriptografi asimetris, sehingga lebih cocok diterapkan untuk pengamanan data dalam jumlah besar pada sistem berbasis web (Tamin et al., 2025).

Studi komparatif yang dilakukan oleh (Hidayatuloh et al., 2026) algoritma AES-128 dan Blowfish menunjukkan bahwa meskipun Blowfish unggul sekitar 18,5% dari sisi kecepatan enkripsi karena arsitektur Feistel yang efisien, AES-128 memiliki stabilitas keamanan yang lebih tinggi dan proses pembentukan kunci yang lebih cepat, sehingga AES-128 lebih direkomendasikan untuk pengamanan data sensitif tingkat tinggi seperti data rekam medis pasien.

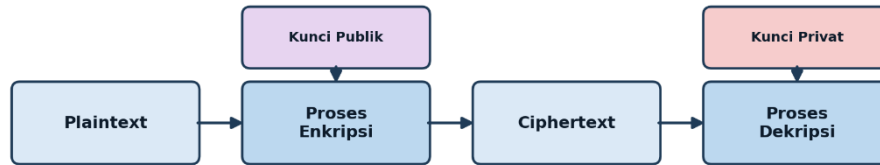
2.6.2 Kriptografi Kunci Asimetris

Kriptografi kunci asimetris merupakan metode kriptografi yang menggunakan dua kunci yang berbeda namun saling berhubungan, yaitu kunci publik (public key) dan kunci privat (private key). Kunci publik dapat dibagikan secara luas, sedangkan kunci privat harus dijaga kerahasiaannya oleh pemiliknya. Dalam penelitian ini digunakan kriptografi kunci simetris karena lebih efisien untuk pengamanan data dalam sistem berbasis web (Hakim & Margolang, 2025).

(a) Kriptografi Kunci Simetris



(b) Kriptografi Kunci Asimetris



Gambar 2. 6 Perbandingan Kriptografi Kunci Simetris dan Asimetris

2.6.3 Enkripsi dan Dekripsi

Enkripsi merupakan proses pengamanan data dengan cara mengubah data asli (plaintext) menjadi bentuk yang tidak dapat dibaca (ciphertext) menggunakan algoritma kriptografi tertentu. Sedangkan dekripsi adalah proses kebalikan dari enkripsi, yaitu mengubah ciphertext kembali menjadi bentuk semula (plaintext) menggunakan kunci yang sesuai (Ramadhani et al., 2022).

Tabel 2.1 Perbandingan Kriptografi Kunci Simetris dan Asimetris

Aspek	Kunci Simetris	Kunci Asimetris
Jumlah kunci	Satu kunci sama untuk enkripsi & dekripsi	Dua kunci berbeda (publik & privat)
Kecepatan proses	Lebih cepat	Lebih lambat
Distribusi kunci	Sulit (kunci harus dirahasiakan kedua pihak)	Lebih mudah (kunci publik dapat disebar)
Cocok untuk Data berukuran besar Pertukaran kunci & tanda tangan digital Contoh	Cocok untuk Data berukuran besar Pertukaran kunci & tanda tangan digital Contoh algoritma	Cocok untuk Data berukuran besar Pertukaran kunci & tanda tangan digital Contoh algoritma

Aspek	Kunci Simetris	Kunci Asimetris
algoritma AES, DES, Blowfish RSA, ECC, ElGamal	AES, DES, Blowfish RSA, ECC, El Gamal	AES, DES, Blowfish RSA, ECC, El Gamal

2.6.4 Plaintext dan Ciphertext

Plaintext adalah data asli yang masih dapat dibaca sebelum dilakukan proses enkripsi. Dalam konteks sistem informasi kesehatan, plaintext dapat berupa data identitas pasien, riwayat penyakit, hasil pemeriksaan, dan informasi medis lainnya. Plaintext merupakan bentuk data yang paling rentan apabila tersimpan atau terkirim tanpa perlindungan, karena isinya dapat langsung dipahami oleh siapa pun yang memperolehnya.

Ciphertext adalah hasil dari proses enkripsi, yaitu data yang telah diubah menjadi bentuk tersandi sehingga tidak dapat dipahami tanpa melalui proses dekripsi (Tamin et al., 2025). Ciphertext tampak sebagai rangkaian karakter acak yang tidak memiliki makna apabila dibaca secara langsung, sehingga meskipun berhasil diakses oleh pihak yang tidak berwenang, isinya tetap tidak dapat dimanfaatkan. Transformasi dari plaintext menjadi ciphertext inilah yang menjadi inti dari pengamanan data menggunakan kriptografi, sebagaimana diilustrasikan pada Gambar 2.7 berikut.

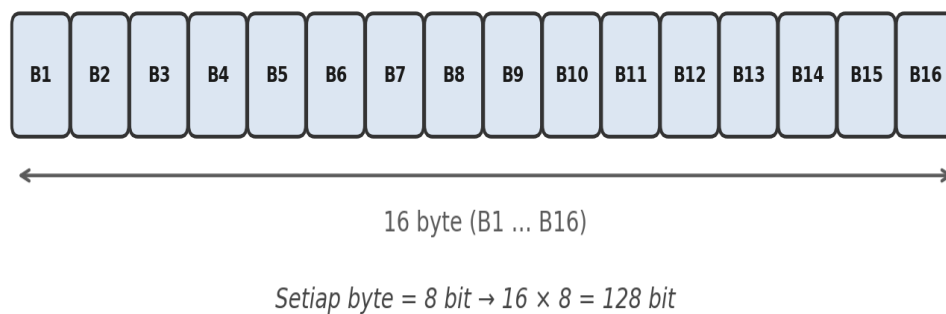


Gambar 2. 7 Transformasi Plaintext menjadi Ciphertext

2.6.5 Kunci Kriptografi (Key)

Kunci kriptografi (key) adalah parameter rahasia yang digunakan dalam proses enkripsi dan dekripsi. Kunci inilah yang menentukan bagaimana plaintext diubah menjadi ciphertext dan sebaliknya, sehingga kerahasiaan kunci menjadi faktor paling menentukan dalam keamanan sebuah sistem kriptografi. Pada algoritma AES-128, panjang kunci yang digunakan adalah 128 bit atau setara dengan 16 byte. Panjang kunci ini menghasilkan jumlah kemungkinan kombinasi kunci yang sangat besar, yaitu sebanyak 2 pangkat 128, sehingga secara praktis tidak mungkin ditembus melalui serangan brute force menggunakan teknologi komputasi saat ini.

Keamanan suatu sistem kriptografi sangat bergantung pada kerahasiaan kunci ini (Donald L. Evans, Phillip J. Bond, 2022). Sekuat apa pun algoritma yang digunakan, keamanan data akan tetap terancam apabila kunci tersebut bocor atau dikelola dengan tidak baik. Oleh karena itu, dalam implementasi sistem, kunci tidak boleh disimpan secara hardcoded di dalam kode program, melainkan dikelola melalui mekanisme penyimpanan yang aman seperti variabel lingkungan (environment variable) atau layanan manajemen kunci khusus. Struktur kunci AES-128 yang terdiri dari 16 byte ditunjukkan pada Gambar 2.8 berikut.



Gambar 2. 8 Struktur Kunci AES-128 (128 bit / 16 byte)

2.9 Algoritma Advanced Encryption Standard (AES) 128

Advanced Encryption Standard (AES) adalah algoritma enkripsi simetris berbasis block cipher yang telah ditetapkan oleh National Institute of Standards and Technology (NIST) sebagai standar enkripsi resmi melalui publikasi FIPS PUB 197 (Donald L. Evans, Phillip J. Bond, 2022) AES memiliki tiga varian berdasarkan panjang kunci, yaitu AES-128, AES-192, dan AES-256.

Posisi AES sebagai standar enkripsi modern terus dipertahankan hingga saat ini. Survei panoramik yang dilakukan oleh peneliti dalam International Journal of Information Security (Ganesh et al., 2025) menyatakan bahwa AES tetap menjadi tulang punggung perlindungan data kontemporer, mulai dari penyimpanan enterprise, lalu lintas web, perangkat Internet of Things (IoT), hingga infrastruktur cloud berkinerja tinggi. AES juga ditetapkan sebagai persyaratan wajib pada industri yang teregulasi seperti sektor kesehatan dan keuangan untuk memastikan praktik keamanan yang kuat sesuai standar FIPS PUB 197.

Table 2. 2 Perbandingan Varian AES

Varian	Panjang Kunci (Nk)	Panjang Blok (Nb)	Jumlah Putaran (Nr)
AES-128	4 (128 bit)	4 (128 bit)	10
AES-192	6 (192 bit)	4 (128 bit)	12
AES-256	8 (256 bit)	4 (128 bit)	14

Algoritma AES memiliki tiga parameter utama, yaitu panjang blok 128 bit, panjang kunci yang bervariasi (128/192/256 bit), dan jumlah putaran (10/12/14). Pada penelitian ini digunakan varian AES-128.

2.9.1 Key Expansion

Key Expansion adalah proses pembangkitan kunci ronde (round key) dari kunci utama. Pada AES-128, kunci utama 128 bit dikembangkan menjadi 11 kunci

ronde (1 kunci awal + 10 kunci putaran) melalui operasi substitusi byte, rotasi, dan XOR.

2.9.2 Tahapan Transformasi AES

Proses enkripsi AES-128 terdiri dari empat tahapan transformasi utama:

1. SubBytes – substitusi non-linear setiap byte menggunakan tabel S-Box.
2. ShiftRows – pergeseran baris matriks state (baris ke-2 digeser 1 byte, baris ke-3 digeser 2 byte, baris ke-4 digeser 3 byte).
3. MixColumns – transformasi kolom dengan operasi perkalian dalam medan hingga (finite field) untuk meningkatkan difusi data.
4. AddRoundKey – operasi XOR antara state dengan round key.

Tabel S-Box merupakan tabel substitusi tetap berukuran 16×16 (256 nilai) yang digunakan pada tahap SubBytes untuk menggantikan setiap byte state dengan byte lain secara non-linear. Substitusi dilakukan dengan membaca nibble tinggi sebagai indeks baris dan nibble rendah sebagai indeks kolom. Sebagai contoh, byte 0x53 disubstitusi menjadi 0xED. Pada proses dekripsi (InvSubBytes) digunakan tabel Inverse S-Box. Nilai-nilai pada tabel ini bersifat baku sesuai standar AES (FIPS 197).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
C	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
D	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
E	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
F	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

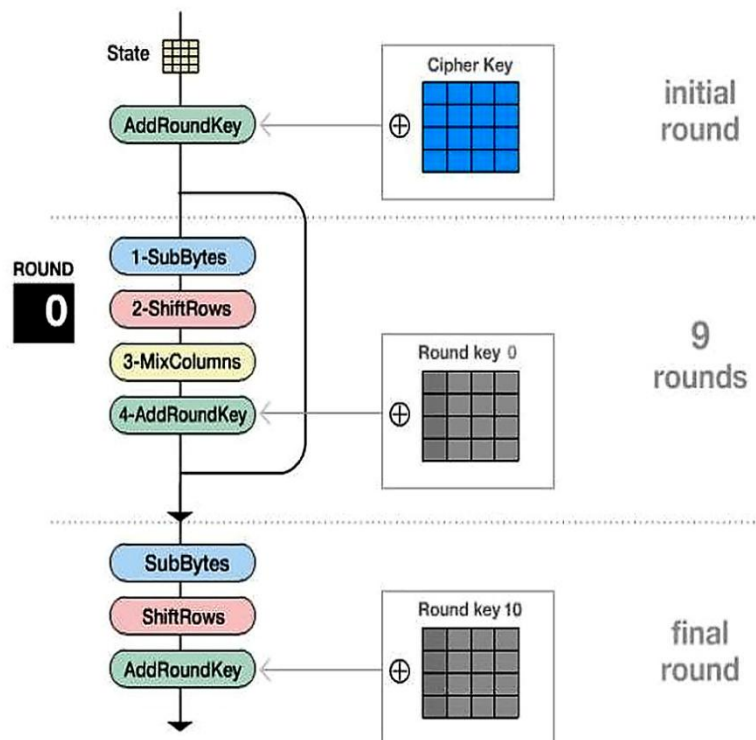
Gambar 2.9 Tabel Forward S-Box AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
A	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
B	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
C	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
D	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
E	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
F	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

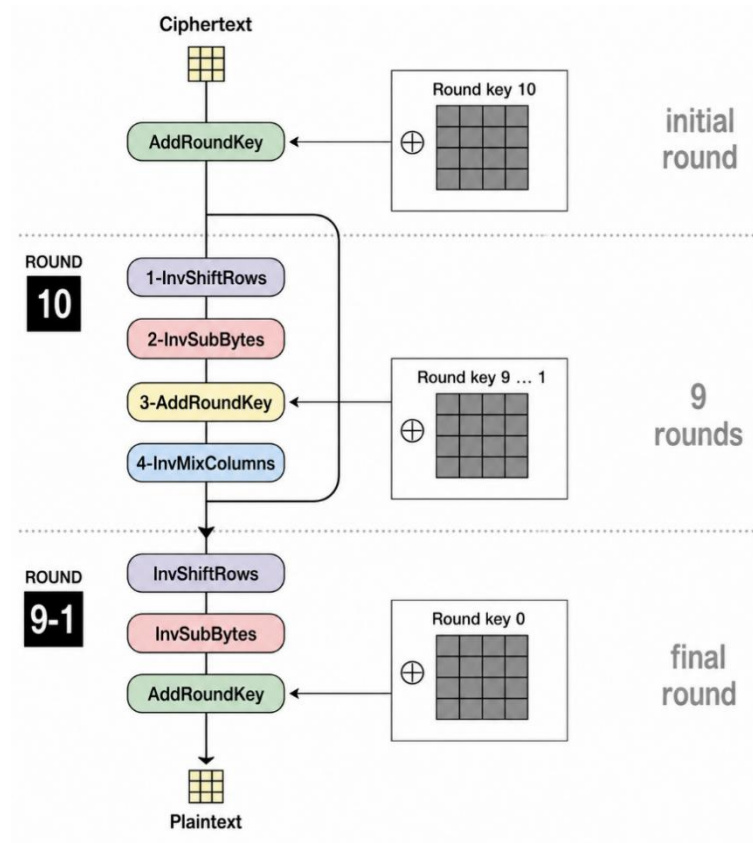
Gambar 2. 10 Tabel Inverse S-Box AES

2.9.3 Alur Proses Enkripsi dan Dekripsi AES-128

Proses enkripsi AES-128 secara keseluruhan terdiri dari: (1) AddRoundKey awal, (2) 9 putaran (SubBytes $\rightarrow$ ShiftRows $\rightarrow$ MixColumns $\rightarrow$ AddRoundKey), dan (3) putaran akhir (SubBytes $\rightarrow$ ShiftRows $\rightarrow$ AddRoundKey tanpa MixColumns). Untuk proses dekripsi, digunakan transformasi invers (InvSubBytes, InvShiftRows, InvMixColumns) dengan urutan yang dibalik (Tamin et al., 2025).



Gambar 2. 11 Proses Enkripsi AES



Gambar 2. 12 Proses Dekripsi AES

2.10 Database

Database merupakan kumpulan data yang terstruktur dan disimpan secara sistematis sehingga dapat diakses dan dikelola dengan mudah. Dalam sistem informasi berbasis web, database digunakan untuk menyimpan data pasien, rekam medis, serta informasi administrasi lainnya.

2.11 Keamanan Database

Keamanan database merupakan upaya untuk melindungi data yang tersimpan dari berbagai ancaman seperti akses tidak sah, pencurian data, maupun manipulasi data. Beberapa ancaman yang umum terjadi antara lain SQL Injection, akses ilegal, serta kebocoran data. Penerapan algoritma kriptografi seperti AES-128 dalam database dapat meningkatkan keamanan data dengan menyimpan data dalam bentuk terenkripsi (Ramadhani et al., 2022).

2.12 Sistem Berbasis Web

Sistem berbasis web adalah aplikasi yang dapat diakses melalui browser menggunakan jaringan internet atau intranet. Karakteristik sistem berbasis web yang terhubung ke jaringan menjadikannya rentan terhadap ancaman keamanan, sehingga diperlukan penerapan algoritma kriptografi untuk melindungi data yang disimpan maupun dikirimkan (Bakri et al., 2024). Penerapan kriptografi pada sistem berbasis web telah banyak dilakukan dalam berbagai bentuk, seperti enkripsi alamat URL pada aplikasi berbasis PHP menggunakan algoritma MD5 untuk menyembunyikan parameter sensitif dari pengguna (Basri & Sulaiman, 2024). Berbagai penelitian terkini juga telah membuktikan kelayakan penerapan AES-128 pada sistem berbasis web di berbagai domain. (Sidqon & Nafis, 2024) berhasil menerapkan AES-128 pada Sistem E-Arsip Berbasis Website di KPU Sidoarjo dengan hasil pengujian avalanche effect rata-rata mencapai 50,15%, mendekati nilai ideal. Sementara itu, (Vaiyapuri et al., 2021) menerapkan AES-128 untuk pengamanan dokumen berbasis web pada PT. XYZ dengan hasil performa yang memuaskan. Hal ini menunjukkan bahwa penerapan algoritma kriptografi pada sistem berbasis web merupakan kebutuhan penting untuk menjaga kerahasiaan data, baik pada level URL, sesi pengguna, maupun penyimpanan data pada database.

2.13 Autentikasi dan Otorisasi

Autentikasi adalah proses verifikasi identitas pengguna, biasanya melalui login menggunakan username dan password. Otorisasi adalah proses pemberian hak akses kepada pengguna setelah berhasil melalui proses autentikasi. Dalam sistem informasi kesehatan, autentikasi dan otorisasi penting untuk memastikan hanya pihak yang berwenang yang dapat mengakses data pasien (Puteri et al., 2024).

2.14 Penelitian Terdahulu

Beberapa penelitian terdahulu yang berkaitan dengan penerapan algoritma AES untuk pengamanan data sistem informasi kesehatan dijadikan referensi dalam penelitian ini. Ringkasan penelitian terdahulu disajikan pada Tabel 2.2 berikut.

Table 2. 3 Ringkasan Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Hasil / Temuan	Relevansi
1	(Ramadhani et al., 2022)	Implementasi Algoritma Advanced Encryption Standard 128 untuk Pengamanan Database Sistem	AES-128 berhasil diterapkan untuk mengenkripsi data sensitif dalam database, dengan overhead waktu yang dapat diterima.	Menjadi dasar penerapan AES-128 pada lapisan database sistem yang dirancang.
2	(Hakim & Margolang, 2025)	Analisis Keamanan Data Rekam Medis Digital Menggunakan Algoritma Kriptografi AES	Sistem rekam medis digital tanpa pengamanan AES sangat rentan terhadap akses tidak sah. Penerapan AES meningkatkan kerahasiaan data secara signifikan.	Memperkuat urgensi penerapan AES pada sistem rekam medis seperti pada Puskesmas Batahan.
3	(Tamin et al., 2025)	Penerapan Algoritma AES-128 untuk Mengamankan File Rekam Medis Pasien	Implementasi AES-128 mampu menjaga kerahasiaan file rekam medis dengan proses enkripsi/dekripsi yang efisien.	Mendukung penggunaan AES-128 spesifik untuk pengamanan data rekam medis pasien.

No	Peneliti & Tahun	Judul Penelitian	Hasil / Temuan	Relevansi
4	(Puteri et al., 2024)	Keamanan dan Kerahasiaan Data Medis Pasien dalam Implementasi Rekam Medis Elektronik	Penerapan AES-128 menghasilkan overhead waktu yang dapat diterima dan tetap menjaga prinsip CIA pada sistem informasi kesehatan.	Menjadi landasan analisis performa dan prinsip keamanan CIA pada sistem yang dibangun.
5	(Indriyajati et al., 2024)	Analisis Keamanan Data Electronic Medical Record DTO Kementerian Kesehatan Indonesia	Sebagian besar sistem informasi kesehatan tingkat primer di Indonesia belum menerapkan enkripsi pada data sensitif.	Menegaskan adanya kesenjangan keamanan pada puskesmas yang menjadi latar belakang penelitian.
6	(Bakri et al., 2024)	Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem	Pendekatan rekayasa perangkat lunak yang sistematis menghasilkan sistem yang valid, terukur, dan teruji.	Menjadi acuan metodologi penelitian rekayasa perangkat lunak yang digunakan.
7	(Candra Aditya Pinuyut et al., 2024)	Analisis Kinerja Algoritma Advanced Encryption Standard (AES) Termodifikasi	AES standar memiliki Avalanche Effect 49,63%, mendekati nilai ideal 50%. Throughput stabil dibandingkan	Menjadi acuan benchmark performa dan parameter pengujian

No	Peneliti & Tahun	Judul Penelitian	Hasil / Temuan	Relevansi
		dalam Enkripsi dan Dekripsi Data	algoritma Serpent, Twofish, DES, dan 3DES.	avalanche effect pada sistem yang dibangun.
8	(Prabowo & Nizirwan, 2025)	Pengujian Model Simulasi Efek Avalanche Kriptografi Simetris Algoritma AES 128-bit Mode ECB dan CBC	Mode CBC pada AES-128 menghasilkan avalanche effect mendekati 50% (ideal) dan difusi bit yang lebih merata dibandingkan mode ECB.	Mendukung pemilihan mode operasi CBC pada implementasi AES-128 di sistem Puskesmas Batahan.
9	(Sidqon & Nafis, 2024)	Rancang Bangun Sistem E-Arsip Berbasis Website Menggunakan Metode Enkripsi AES Studi Kasus KPU Sidoarjo	Implementasi AES pada sistem berbasis web menghasilkan avalanche effect rata-rata 50,15% dengan tingkat keandalan, keamanan, dan efisiensi yang baik.	Membuktikan kelayakan penerapan AES-128 pada arsitektur sistem berbasis web seperti yang dibangun dalam penelitian ini.

Berdasarkan penelitian terdahulu di atas, dapat diidentifikasi bahwa AES-128 merupakan algoritma yang konsisten terbukti efektif dalam pengamanan data sistem informasi kesehatan, baik dari sisi keamanan (avalanche effect mendekati 50%) maupun performa (overhead waktu yang dapat diterima). Penelitian terbaru

juga merekomendasikan penggunaan mode operasi CBC untuk menghasilkan difusi bit yang lebih merata serta menempatkan AES-128 sebagai pilihan ideal untuk fasilitas kesehatan dengan sumber daya komputasi terbatas. Namun, sebagian besar penelitian belum secara khusus mengkaji implementasi AES-128 pada konteks puskesmas tingkat kecamatan seperti Puskesmas Batahan.

Penelitian oleh (Siregar et al., 2024) mengimplementasikan kombinasi kriptografi simetris dan asimetris (AES, RSA, dan Arnold Cat Map) untuk mengamankan citra digital. Penelitian tersebut menggunakan mode CBC pada AES dan mengukur kualitas enkripsi melalui analisis histogram, koefisien korelasi, dan entropi. Perbedaan dengan penelitian ini terletak pada objek yang diamankan, yaitu data teks rekam medis pasien berbasis web, bukan citra digital, serta penggunaan AES-128 secara mandiri tanpa kombinasi algoritma lain.

Oleh karena itu, penelitian ini memberikan kontribusi berupa rancangan dan implementasi sistem pengamanan data pasien berbasis web menggunakan AES-128 dengan mode CBC yang disesuaikan dengan kebutuhan operasional puskesmas, sekaligus memenuhi tuntutan regulasi UU No. 27 Tahun 2022 tentang Perlindungan Data Pribadi.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Jenis Penelitian

Penelitian ini merupakan jenis penelitian rekayasa perangkat lunak (software engineering research) yang bersifat deskriptif dan implementatif. Pendekatan deskriptif digunakan untuk mengidentifikasi dan menganalisis kondisi pengamanan data pasien yang sedang berjalan di Puskesmas Batahan, sedangkan pendekatan implementatif digunakan untuk merancang dan mengembangkan sistem pengamanan berbasis web dengan mengimplementasikan algoritma Advanced Encryption Standard 128-bit (AES-128).

Metode penelitian rekayasa perangkat lunak dipilih karena penelitian ini tidak hanya bersifat kajian teoritis, melainkan menghasilkan artefak berupa perangkat lunak yang dapat digunakan secara nyata. Menurut (Bakri et al., 2024), penelitian rekayasa perangkat lunak yang baik harus memiliki validitas internal yang kuat, dibangun berdasarkan kebutuhan nyata, dan menghasilkan solusi yang terukur melalui pengujian sistematis.

3.2 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di Puskesmas Batahan yang berlokasi di Kecamatan Batahan, Kabupaten Mandailing Natal, Provinsi Sumatera Utara. Waktu pelaksanaan penelitian dijadwalkan berlangsung dari bulan Januari hingga Juni tahun 2026, mencakup seluruh tahapan mulai dari pengumpulan data awal, analisis sistem, perancangan, implementasi, pengujian, hingga penyusunan laporan akhir.

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No	Kegiatan	Jan	Feb	Mar	Apr	Mei	Jun
1	Pengumpulan Data & Observasi	✓					

No	Kegiatan	Jan	Feb	Mar	Apr	Mei	Jun
2	Analisis Sistem Berjalan	✓	✓				
3	Perancangan Sistem		✓	✓			
4	Implementasi (Coding)			✓	✓		
5	Pengujian Sistem				✓	✓	
6	Penyusunan Laporan					✓	✓

3.3 Metode Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui tiga metode utama, yaitu observasi, wawancara, dan studi literatur.

3.3.1 Observasi

Observasi dilakukan dengan mengamati secara langsung proses pengelolaan dan pencatatan data pasien di Puskesmas Batahan dengan teknik observasi non-partisipan. Melalui observasi, ditemukan bahwa sistem yang berjalan belum menerapkan mekanisme enkripsi pada data yang disimpan di database, sehingga data pasien tersimpan dalam bentuk plaintext yang rentan terhadap akses tidak sah (Puteri et al., 2024).

3.3.2 Wawancara

Wawancara dilakukan secara terstruktur dengan kepala puskesmas, petugas administrasi, dan tenaga medis. Wawancara bertujuan untuk menggali informasi mengenai kondisi sistem saat ini, kebutuhan fitur sistem yang akan dikembangkan, tingkat pemahaman pengguna terhadap keamanan data, dan harapan terhadap gsistem baru yang lebih aman.

3.3.3 Studi Literatur

Studi literatur dilakukan dengan mengkaji jurnal nasional dan internasional, prosiding konferensi, buku teks, serta dokumentasi resmi dari NIST. Sumber

literatur diambil dari Google Scholar, IEEE Xplore, ScienceDirect, dan portal jurnal Sinta dengan kriteria terbitan lima tahun terakhir (2020–2025).

3.4 Analisis Sistem yang Sedang Berjalan

Berdasarkan hasil observasi dan wawancara, sistem yang berjalan saat ini memiliki beberapa kelemahan mendasar dalam aspek keamanan data. Pertama, data pasien disimpan dalam database dalam format plaintext tanpa mekanisme enkripsi (Indriyajati et al., 2024). Kedua, sistem autentikasi yang digunakan belum cukup kuat, kata sandi pengguna disimpan tanpa proses hashing yang memadai. Ketiga, belum terdapat mekanisme audit log yang mencatat aktivitas akses terhadap data sensitif.

3.5 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan yang harus dipenuhi oleh sistem. Kebutuhan dibagi menjadi kebutuhan fungsional dan non-fungsional.

3.5.1 Kebutuhan Fungsional

1. Sistem harus menyediakan fitur autentikasi (login dan logout) dengan validasi username dan password yang di-hash.
2. Sistem harus dapat melakukan manajemen data pasien (CRUD).
3. Sistem harus dapat melakukan manajemen data rekam medis (CRUD).
4. Sistem harus dapat mengenkripsi data sensitif pasien menggunakan AES-128 sebelum disimpan.
5. Sistem harus dapat mendekripsi data ketika ditampilkan kepada pengguna berwenang.
6. Sistem harus membedakan hak akses pengguna berdasarkan peran (admin, dokter, petugas).

3.5.2 Kebutuhan Non-Fungsional

1. Keamanan: Sistem menerapkan enkripsi AES-128 pada field sensitif. Kunci enkripsi tidak boleh disimpan secara hardcoded.
2. Keandalan: Sistem harus stabil dan memberikan hasil enkripsi/dekripsi yang konsisten.
3. Kegunaan: Antarmuka intuitif dan mudah digunakan tanpa pelatihan teknis yang mendalam.
4. Kinerja: Waktu respons di bawah dua detik untuk setiap operasi.
5. Portabilitas: Dapat diakses melalui Chrome, Firefox, Edge tanpa instalasi tambahan.

3.5.3 Spesifikasi Perangkat yang Digunakan

Tabel 3. 2 Spesifikasi Perangkat Keras

No	Komponen	Spesifikasi
1	Prosesor	Intel Core i3 / AMD Ryzen 3 (min. 2.0 GHz)
2	RAM	Minimal 4 GB
3	Penyimpanan	Minimal 128 GB SSD
4	Sistem Operasi	Windows 10/11 atau Ubuntu 20.04 LTS
5	Jaringan	Koneksi internet minimal 1 Mbps

Tabel 3. 3 Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Node.js v20 LTS	Runtime JavaScript untuk server-side
2	Express.js v4	Framework backend web
3	React.js v18 / Next.js v14	Framework frontend

No	Perangkat Lunak	Keterangan
4	MySQL 8.0	Sistem manajemen basis data
5	Visual Studio Code	Editor kode
6	XAMPP / Laragon	Lingkungan pengembangan lokal
7	Postman	Pengujian API
8	Git	Version control

3.6 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah model Waterfall. Model ini dipilih karena ruang lingkup dan kebutuhan sistem telah teridentifikasi secara jelas, sistem yang dikembangkan bersifat terbatas dan terukur, serta model ini menyediakan dokumentasi yang sistematis pada setiap tahapan (Saravanos, 2023).

3.6.1 Analisis Kebutuhan

Fase awal mengidentifikasi, mendokumentasikan, dan memvalidasi seluruh kebutuhan sistem melalui observasi dan wawancara. Output: dokumen spesifikasi kebutuhan sistem.

3.6.2 Perancangan Sistem

Menerjemahkan kebutuhan menjadi cetak biru sistem. Mencakup perancangan arsitektur, basis data, proses enkripsi/dekripsi, dan antarmuka. Alat bantu: Use Case Diagram, Activity Diagram, Flowchart, dan ERD.

3.6.3 Implementasi

Penerjemahan desain ke kode program menggunakan Node.js, Express.js, React.js, dan MySQL. AES-128 diimplementasikan menggunakan pustaka crypto-js yang mengikuti FIPS PUB 197.

3.6.4 Pengujian

Pengujian dilakukan menggunakan Black Box Testing dan pengujian khusus enkripsi/dekripsi (Utomo et al., 2020).

3.6.5 Pemeliharaan

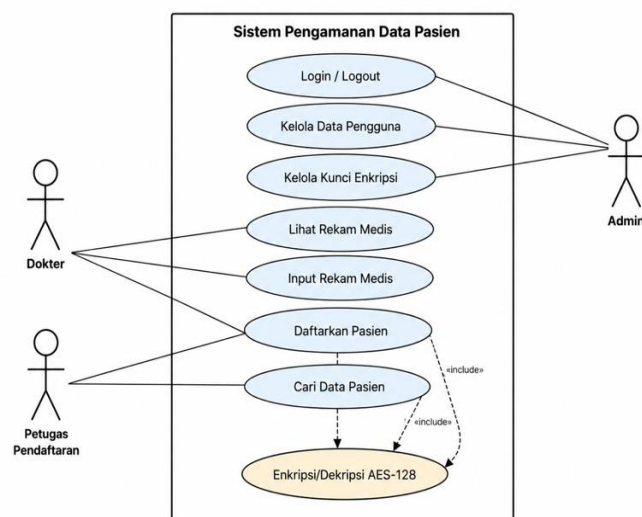
Evaluasi kinerja sistem, perbaikan bug pasca-implementasi, serta dokumentasi untuk mendukung pengembangan lanjutan.

3.7 Perancangan Sistem

Perancangan sistem mencakup use case diagram, activity diagram, flowchart enkripsi/dekripsi AES-128, dan Entity Relationship Diagram (ERD). Seluruh diagram dirancang mengikuti notasi Unified Modeling Language (UML) standar. Perancangan sistem pada bab ini disusun sebagai gambaran awal (blueprint) yang menjadi acuan pada tahap implementasi. Rancangan ini bersifat usulan dan masih dapat disesuaikan ketika pengembangan sistem dilaksanakan.

3.7.1 Use Case Diagram

Use case diagram menggambarkan interaksi antara aktor (Admin, Dokter, dan Petugas Pendaftaran) dengan sistem. Rancangan use case diagram sistem ditampilkan pada Gambar 3.1 berikut.



Gambar 3. 1 Use Case Diagram Sistem

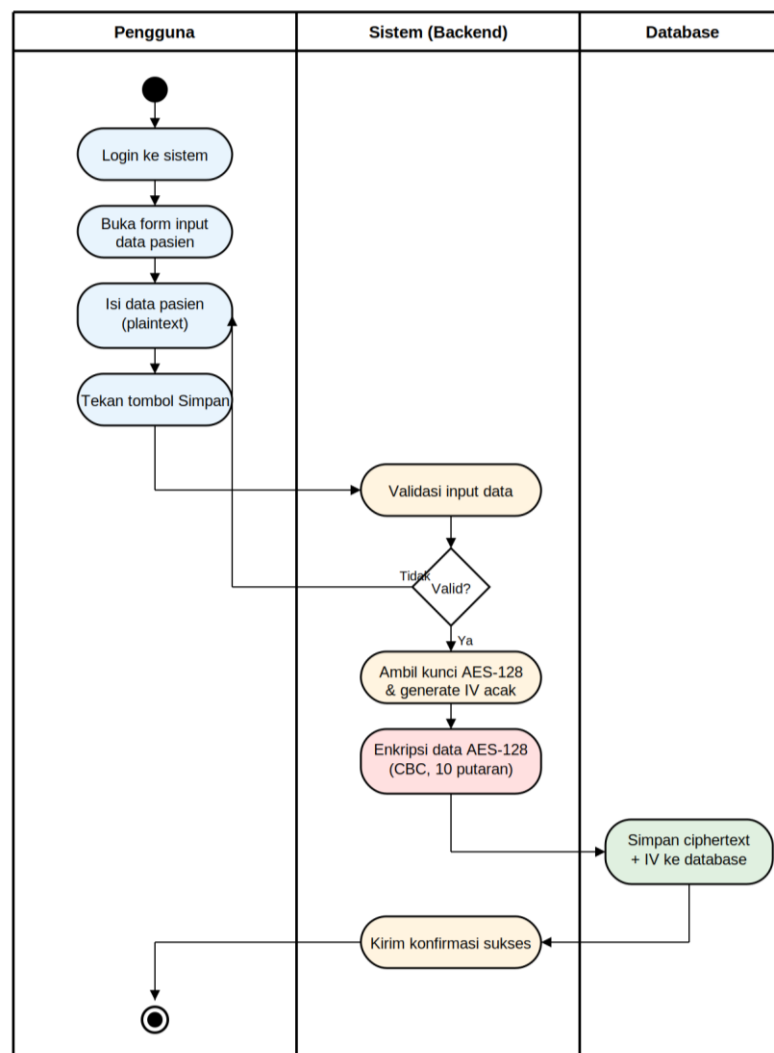
Penjelasan masing-masing use case dapat dilihat pada Tabel 3.4 berikut.

Tabel 3. 4 Deskripsi Use Case Sistem

No	Aktor	Use Case	Deskripsi
1	Admin	Kelola Data Pengguna	Menambah, mengubah, menonaktifkan akun pengguna sistem.
2	Admin	Kelola Kunci Enkripsi	Mengelola kunci AES-128 yang digunakan sistem.
3	Dokter	Lihat Rekam Medis	Mengakses rekam medis pasien dalam bentuk terdekripsi.
4	Dokter	Input Rekam Medis	Menginput data pemeriksaan yang akan terenkripsi sebelum disimpan.
5	Petugas	Daftarkan Pasien	Mendaftarkan pasien baru, data identitas akan terenkripsi.
6	Petugas	Cari Data Pasien	Mencari data pasien berdasarkan nama atau nomor RM.
7	Semua	Login/Logout	Autentikasi sebelum mengakses sistem.

3.7.2 Activity Diagram

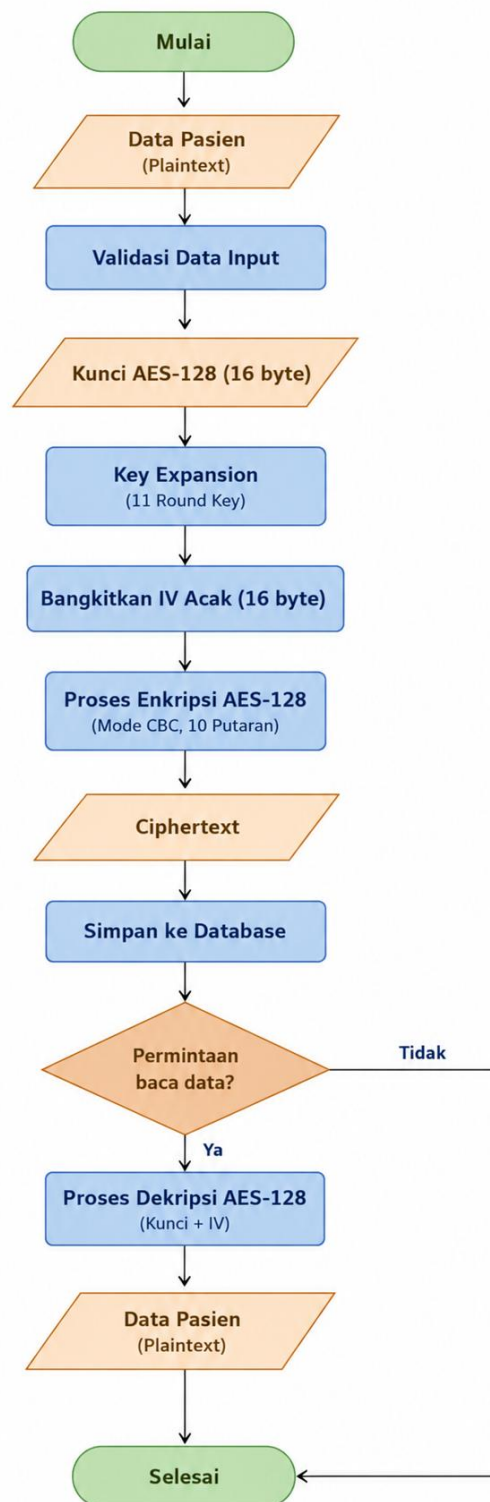
Activity diagram menggambarkan alur aktivitas dalam sistem. Alur utama dimulai dari autentikasi pengguna, input data, proses enkripsi AES-128, penyimpanan ciphertext ke database, dan dekripsi ketika data ditampilkan. Activity diagram untuk proses input dan penyimpanan data pasien dapat dilihat pada Gambar 3.2 berikut.



Gambar 3. 2 Activity Diagram Input dan Penyimpanan Data Pasien

3.7.3 Flowchart Proses Enkripsi dan Dekripsi AES-128

Flowchart berikut menggambarkan alur logika proses enkripsi dan dekripsi data pasien menggunakan algoritma AES-128.



Gambar 3. 3 Flowchart Proses Enkripsi dan Dekripsi AES-128

Alur Proses Enkripsi AES-128:

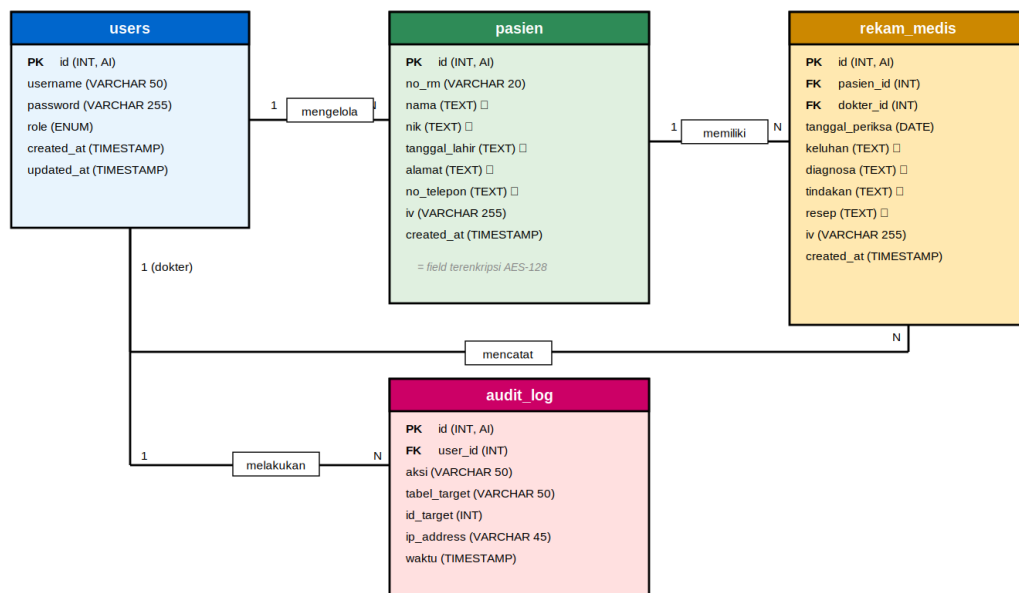
1. Pengguna memasukkan data pasien melalui formulir antarmuka sistem.
2. Sistem menerima data input sebagai plaintext dan memvalidasinya.
3. Sistem mengambil kunci AES-128 (16 byte) dari konfigurasi yang aman.
4. Sistem menjalankan Key Expansion untuk menghasilkan 11 round key.
5. Sistem membangkitkan Initialization Vector (IV) acak (16 byte).
6. Proses enkripsi dijalankan dalam mode CBC dengan 10 putaran transformasi.
7. Hasil enkripsi (ciphertext) disimpan ke database bersama IV-nya.

Alur Proses Dekripsi AES-128:

1. Pengguna meminta tampilan data pasien atau rekam medis.
2. Sistem mengambil ciphertext beserta IV dari database.
3. Sistem mengambil kunci AES-128 dan menjalankan Key Expansion.
4. Sistem menjalankan proses dekripsi (transformasi invers).
5. Hasil dekripsi (plaintext) ditampilkan ke pengguna.

3.7.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menggambarkan struktur dan hubungan antar entitas dalam database sistem. Sistem ini menggunakan empat entitas utama, yaitu `users`, `pasien`, `rekam_medis`, dan `audit_log`. ERD sistem ditampilkan pada Gambar 3.4 berikut.



Gambar 3. 4 Entity Relationship Diagram (ERD) Sistem

Penjelasan hubungan antar entitas:

1. users $\rightarrow$ pasien (1:N): satu petugas/dokter dapat mengelola banyak data pasien.
2. pasien $\rightarrow$ rekam_medis (1:N): satu pasien dapat memiliki banyak catatan rekam medis.
3. users $\rightarrow$ rekam_medis (1:N): satu dokter dapat mencatat banyak rekam medis.
4. users $\rightarrow$ audit_log (1:N): setiap aktivitas pengguna tercatat di audit_log.

3.8 Perancangan Database

Database sistem dirancang menggunakan MySQL 8.0 dengan mempertimbangkan prinsip normalisasi data hingga bentuk normal ketiga (3NF). Kolom yang menyimpan data sensitif pasien dirancang untuk menyimpan data dalam format ciphertext hasil enkripsi AES-128.

Tabel 3. 1 Struktur Tabel users

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Primary key
2	username	VARCHAR(50)	Username unik
3	password	VARCHAR(255)	Bcrypt hash
4	role	ENUM	admin, dokter, petugas
5	created_at	TIMESTAMP	Waktu pembuatan
6	updated_at	TIMESTAMP	Waktu pembaruan

Tabel 3. 2 Struktur Tabel pasien

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Primary key
2	no_rm	VARCHAR(20)	Nomor rekam medis (plaintext)
3	nama	TEXT	Nama pasien (ENKRIPSI AES-128)
4	nik	TEXT	NIK (ENKRIPSI AES-128)
5	tanggal_lahir	TEXT	Tanggal lahir (ENKRIPSI AES-128)
6	alamat	TEXT	Alamat (ENKRIPSI AES-128)

No	Nama Kolom	Tipe Data	Keterangan
7	no_telepon	TEXT	No telepon (ENKRIPSI AES-128)
8	iv	VARCHAR(255)	Initialization Vector
9	created_at	TIMESTAMP	Waktu pendaftaran

Tabel 3. 3 Struktur Tabel rekam_medis

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Primary key
2	pasien_id	INT (FK)	FK ke tabel pasien
3	dokter_id	INT (FK)	FK ke tabel users
4	tanggal_periksa	DATE	Tanggal pemeriksaan
5	keluhan	TEXT	Keluhan (ENKRIPSI AES-128)
6	diagnosa	TEXT	Diagnosa (ENKRIPSI AES-128)
7	tindakan	TEXT	Tindakan (ENKRIPSI AES-128)
8	resep	TEXT	Resep (ENKRIPSI AES-128)
9	iv	VARCHAR(255)	Initialization Vector
10	created_at	TIMESTAMP	Waktu pencatatan

Tabel 3. 4 Struktur Tabel *audit_log*

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Primary key
2	user_id	INT (FK)	FK ke tabel users
3	aksi	VARCHAR(50)	CREATE, READ, UPDATE, DELETE
4	tabel_target	VARCHAR(50)	Nama tabel yang diakses
5	id_target	INT	ID record yang diakses
6	ip_address	VARCHAR(45)	Alamat IP pengguna
7	waktu	TIMESTAMP	Waktu aktivitas

3.9 Perancangan Antarmuka (Interface)

Perancangan antarmuka sistem dirancang dengan mempertimbangkan prinsip usability bagi tenaga kesehatan yang mungkin tidak memiliki latar belakang teknis. Sistem memiliki beberapa halaman utama: Halaman Login, Dashboard, Manajemen Pasien, Rekam Medis, dan Manajemen Pengguna.

3.9.1 Rancangan Halaman Login

Halaman Login digunakan untuk autentikasi pengguna sebelum dapat mengakses sistem. Pengguna memasukkan username dan password yang akan divalidasi oleh sistem. Password disimpan dalam bentuk hash bcrypt di database.

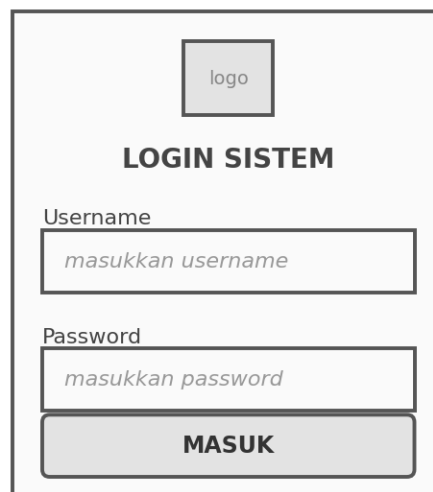


Diagram of a login system interface. At the top center is a box labeled "logo". Below it is the title "LOGIN SISTEM". There are two input fields: "Username" with placeholder text "masukkan username" and "Password" with placeholder text "masukkan password". Below the password field is a button labeled "MASUK".

Gambar 3. 5 Rancangan Antarmuka Halaman Login

3.9.2 Rancangan Halaman Manajemen Data Pasien

Halaman Manajemen Data Pasien menampilkan daftar seluruh pasien yang terdaftar. Pengguna dapat melakukan pencarian, menambah pasien baru, melihat detail, mengubah, atau menghapus data. Data yang ditampilkan merupakan hasil dekripsi dari ciphertext yang tersimpan di database.

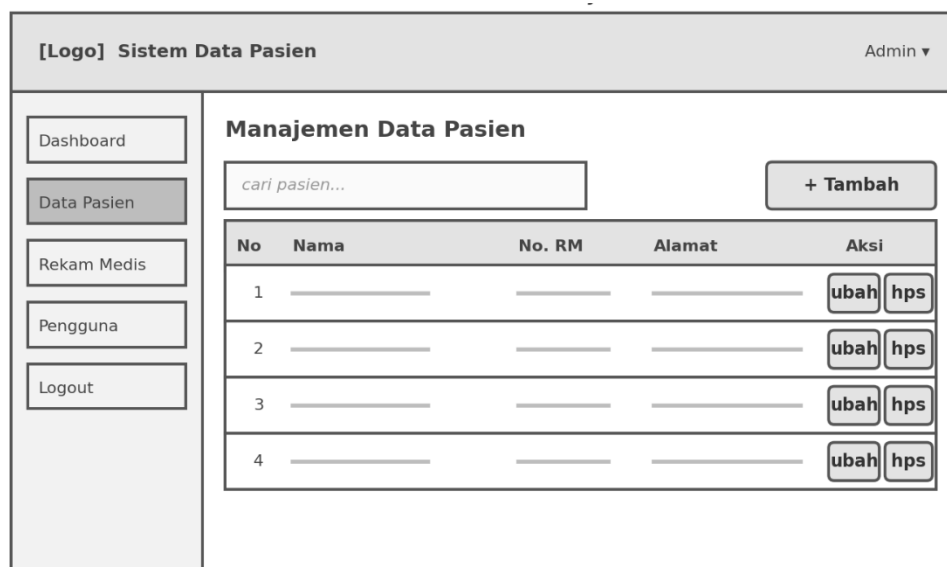


Diagram of a patient data management interface. The header bar contains "[Logo] Sistem Data Pasien" on the left and "Admin ▼" on the right. The left sidebar has buttons for "Dashboard", "Data Pasien" (highlighted), "Rekam Medis", "Pengguna", and "Logout". The main content area is titled "Manajemen Data Pasien" and contains a search bar with placeholder text "cari pasien..." and a "+ Tambah" button. Below this is a table with columns: "No", "Nama", "No. RM", "Alamat", and "Aksi". The table contains four rows of data, each with a "ubah" (change) and "hps" (delete) button in the "Aksi" column.

No	Nama	No. RM	Alamat	Aksi
1				ubah hps
2				ubah hps
3				ubah hps
4				ubah hps

Gambar 3. 6 Rancangan Antarmuka Halaman Manajemen Data Pasien

3.9.3 Rancangan Halaman Input Rekam Medis

Halaman Input Rekam Medis digunakan oleh dokter untuk menginput data pemeriksaan pasien. Data yang dimasukkan (keluhan, diagnosa, tindakan, resep) akan dienkripsi menggunakan AES-128 sebelum disimpan ke database.

Gambar 3. 7 Rancangan Antarmuka Halaman Input Rekam Medis

3.10 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa sistem yang dikembangkan telah memenuhi kebutuhan yang ditetapkan. Metode pengujian yang digunakan adalah Black Box Testing, pengujian khusus enkripsi/dekripsi, dan pengujian performa.

3.10.1 Black Box Testing

Black Box Testing menguji fungsionalitas sistem dari perspektif pengguna tanpa mempertimbangkan implementasi internal kode program. Setiap fungsi diuji dengan memberikan input tertentu dan memverifikasi apakah output yang dihasilkan sesuai harapan (Utomo et al., 2020).

Tahapan Black Box Testing yang dilaksanakan dalam penelitian ini terdiri dari lima tahap berikut:

1. Identifikasi Fitur Pengujian — Menentukan fitur-fitur sistem yang akan diuji, mencakup login, pendaftaran pasien, input rekam medis, pencarian data, serta mekanisme hak akses berbasis peran.
2. Penyusunan Skenario Pengujian (Test Case) — Setiap fitur dijabarkan menjadi skenario uji yang memuat: ID skenario, deskripsi skenario, input yang diberikan, output yang diharapkan.
3. Eksekusi Pengujian — Pengujian dilakukan dengan menjalankan setiap skenario pada sistem yang sudah diimplementasikan, lalu mencatat output aktual yang dihasilkan sistem.
4. Perbandingan Output — Membandingkan output aktual dengan output yang diharapkan. Status pengujian ditetapkan: Valid (jika sesuai) atau Tidak Valid (jika tidak sesuai).
5. Dokumentasi dan Evaluasi — Seluruh hasil pengujian didokumentasikan dalam tabel pengujian yang memuat kolom: skenario, input, output yang diharapkan, output aktual, dan status. Hasil yang Tidak Valid akan dianalisis penyebabnya dan dilakukan perbaikan, kemudian diuji ulang.

Contoh format tabel skenario pengujian Black Box ditampilkan pada Tabel 3.9 berikut.

Tabel 3. 5 Contoh Skenario Pengujian Black Box

ID	Skenario	Input	Output Diharapkan	Status
TC-01	Login dengan kredensial valid	username & password benar	Berhasil masuk ke Dashboard	Valid

ID	Skenario	Input	Output Diharapkan	Status
TC-02	Login dengan password salah	username benar, password salah	Pesan error 'Login gagal'	Valid
TC-03	Daftar pasien baru	Data pasien lengkap	Data tersimpan terenkripsi	Valid
TC-04	Akses fitur Admin oleh Petugas	Login sebagai petugas	Akses ditolak (403)	Valid

3.10.2 Pengujian Enkripsi dan Dekripsi AES-128

Pengujian kriptografi dilakukan untuk memverifikasi kebenaran dan konsistensi implementasi AES-128. Pengujian memastikan bahwa: (1) ciphertext berbeda dari plaintext, (2) dekripsi dengan kunci benar menghasilkan plaintext yang identik, (3) dekripsi dengan kunci salah tidak menghasilkan data yang dapat dibaca, dan (4) setiap enkripsi pada data yang sama menghasilkan ciphertext berbeda karena IV acak. Pengujian dilakukan dengan membandingkan hasil terhadap test vector NIST FIPS PUB 197 (Donald L. Evans, Phillip J. Bond, 2022).

Selain pengujian fungsional enkripsi dan dekripsi, dilakukan juga pengujian kualitas algoritma menggunakan parameter Avalanche Effect (AE) untuk mengukur sensitivitas algoritma terhadap perubahan input. Avalanche Effect dihitung dengan rumus:

$$AE = (\text{Jumlah bit yang berubah} / \text{Total bit}) \times 100\%$$

Nilai ideal Avalanche Effect mendekati 50%, yang berarti perubahan satu bit pada plaintext atau kunci akan mengubah sekitar setengah bit pada ciphertext. Berdasarkan penelitian (Candra Aditya Pinuyut et al., 2024), AES standar menghasilkan Avalanche Effect sebesar 49,63%, sedangkan (Prabowo & Nizirwan, 2025) membuktikan bahwa mode CBC pada AES-128 menghasilkan avalanche

effect yang lebih stabil mendekati 50% dibandingkan mode ECB, sehingga mode CBC digunakan pada implementasi sistem ini.

3.10.3 Pengujian Performa

Pengujian performa dilakukan untuk mengukur dampak penerapan enkripsi AES-128 terhadap kecepatan respons sistem. Parameter yang diukur meliputi waktu enkripsi per record, waktu dekripsi per record, serta waktu total operasi CRUD dengan dan tanpa enkripsi.

Parameter pengujian performa yang digunakan meliputi waktu enkripsi dan dekripsi dalam satuan milidetik (ms) serta throughput dalam satuan KB/detik. Hasil pengujian akan dibandingkan dengan benchmark dari penelitian terdahulu, di mana AES-128 CBC mencatatkan waktu enkripsi rata-rata 0,382 ms dan waktu dekripsi 0,245 ms (Candra Aditya Pinuyut et al., 2024), sehingga sistem yang dibangun dapat dievaluasi apakah memiliki performa yang sebanding dengan implementasi AES-128 pada penelitian sebelumnya.

Cara Pengukuran Waktu Eksekusi:

Pengukuran waktu dilakukan langsung pada sisi server (backend) menggunakan API berikut:

1. `performance.now()` — fungsi bawaan Node.js dari modul `perf_hooks` yang menghasilkan timestamp dengan presisi tinggi (sampai dengan sub-milidetik / mikrodetik), sehingga akurat untuk mengukur durasi operasi singkat seperti enkripsi/dekripsi.
2. `process.hrtime.bigint()` — alternatif untuk pengukuran presisi nanodetik.

Mekanisme perhitungan waktu dilakukan dengan langkah-langkah berikut:

3. Mencatat waktu awal (`T_start`) tepat sebelum proses enkripsi atau dekripsi dijalankan.
4. Menjalankan proses kriptografi (enkripsi atau dekripsi).
5. Mencatat waktu akhir (`T_end`) segera setelah proses selesai.

6. Menghitung durasi eksekusi dengan rumus: $\text{Durasi} = T_{\text{end}} - T_{\text{start}}$ (dalam milidetik).
7. Mengulang pengukuran sebanyak n kali (minimal 30 iterasi per skenario) untuk mendapatkan nilai yang stabil.
8. Menghitung nilai rata-rata (mean), nilai minimum, dan nilai maksimum dari seluruh iterasi.

Contoh format hasil pengukuran ditampilkan pada Tabel 3.10 berikut.

Tabel 3. 6 Format Pengukuran Waktu Eksekusi

Operasi	Ukuran Data	Rata-rata (ms)	Min (ms)	Max (ms)
Enkripsi 1 record pasien	~250 byte	0.4 – 0.8	0.3	1.2
Dekripsi 1 record pasien	~250 byte	0.4 – 0.8	0.3	1.2
Operasi CRUD penuh + enkripsi	1 record	< 50	—	—

Hasil pengujian performa akan dianalisis untuk memastikan bahwa overhead yang dihasilkan tidak menurunkan pengalaman pengguna sistem secara signifikan. Menurut (Puteri et al., 2024). penerapan AES-128 secara umum menghasilkan overhead waktu yang dapat diterima dalam konteks system informasi Kesehatan berbasis web.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari seluruh tahapan yang sudah dirancang pada Bab III, mulai dari hasil pengumpulan data di lapangan, implementasi sistem, penerapan algoritma AES-128 pada data pasien, sampai hasil pengujian sistem. Pembahasan disusun berurutan supaya alurnya mudah diikuti: dari kebutuhan, ke wujud sistemnya, lalu ke bukti bahwa sistem tersebut benar-benar bekerja seperti yang dijanjikan.

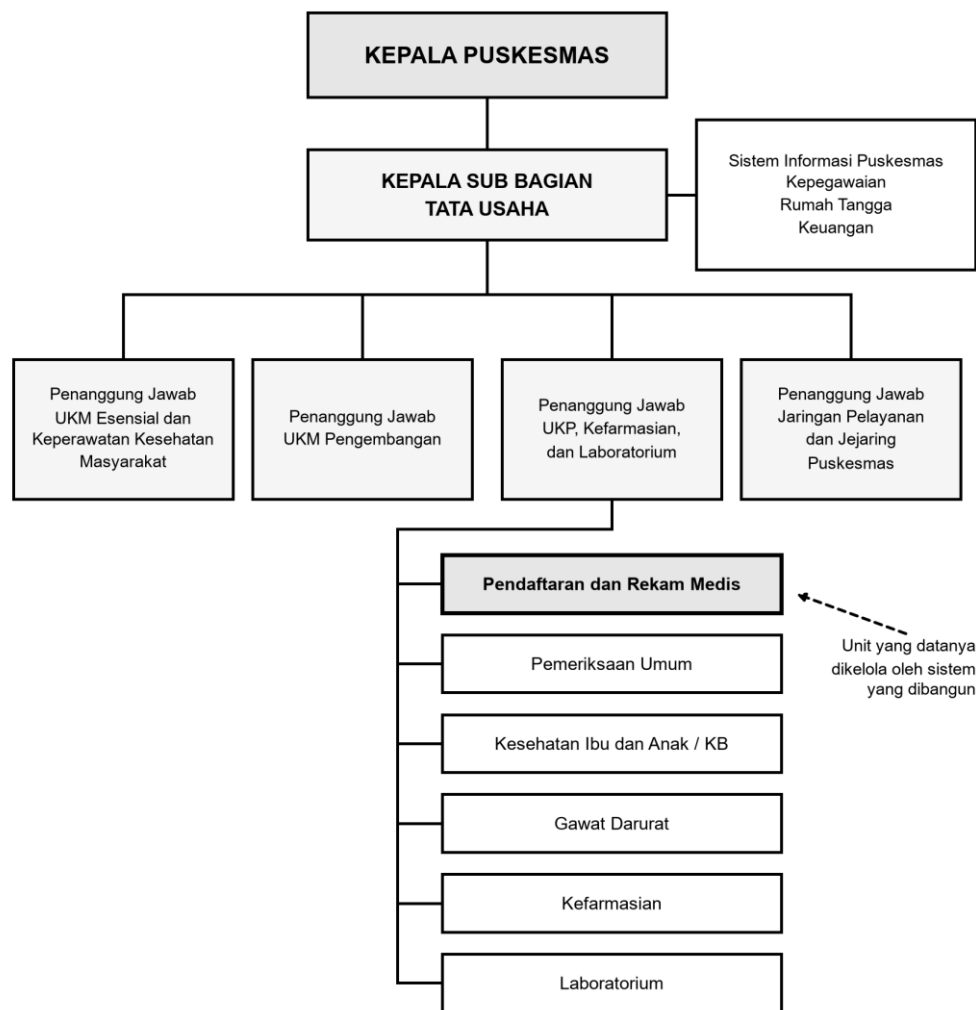
Seluruh angka yang ditampilkan pada bab ini merupakan hasil pengukuran langsung pada sistem yang sudah dibangun, bukan perkiraan. Perhitungan manual AES-128 pada sub-bab 4.4 juga sudah diverifikasi ulang menggunakan skrip agar tidak ada byte yang salah hitung.

4.1 Hasil Pengumpulan Data dan Analisis Kebutuhan

Pengumpulan data dilakukan dengan tiga cara seperti yang sudah dijelaskan pada sub-bab 3.3, yaitu observasi langsung ke Puskesmas Batahan, wawancara dengan pihak puskesmas, dan studi literatur. Hasilnya dipakai sebagai dasar untuk menentukan fitur apa saja yang harus ada di sistem dan data mana yang wajib dienkripsi.

4.1.1 Gambaran Umum Puskesmas Batahan

Puskesmas Batahan merupakan fasilitas kesehatan tingkat pertama (FKTP) milik Pemerintah Kabupaten Mandailing Natal yang melayani masyarakat di wilayah Kecamatan Batahan, Provinsi Sumatera Utara. Layanan yang diberikan mencakup pelayanan rawat jalan, pemeriksaan umum, kesehatan ibu dan anak, imunisasi, serta pelayanan kefarmasian dasar. Struktur organisasi Puskesmas Batahan ditampilkan pada Gambar 4.1.



Susunan organisasi mengikuti Peraturan Menteri Kesehatan Nomor 43 Tahun 2019 tentang Pusat Kesehatan Masyarakat.

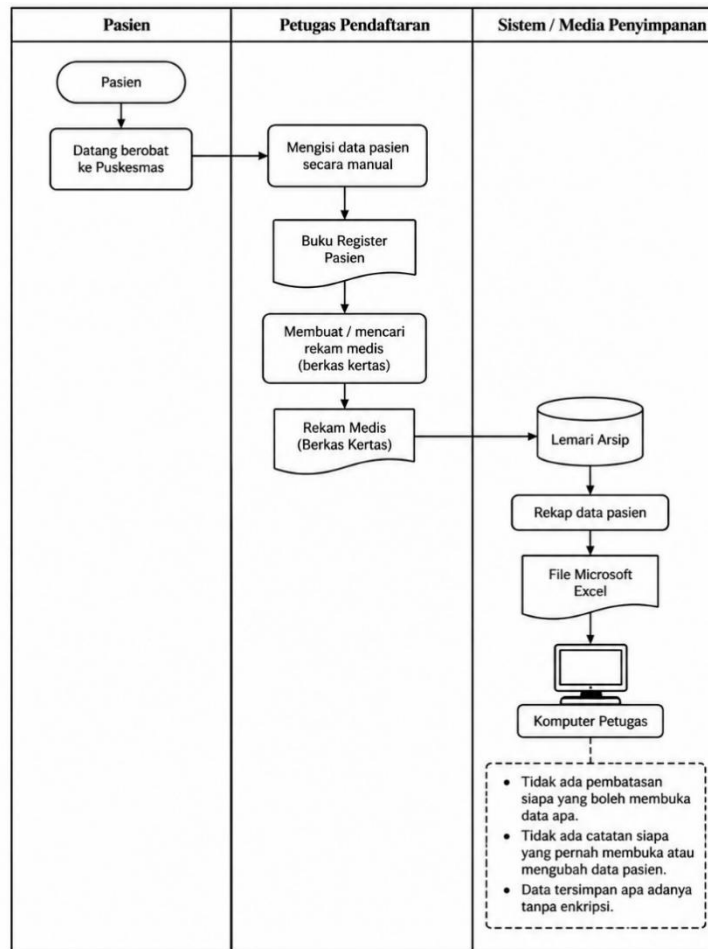
Gambar 4.1 Struktur Organisasi Puskesmas Batahan

Alur pelayanan pasien di Puskesmas Batahan secara umum berjalan seperti berikut. Pasien yang datang mendaftar terlebih dahulu di loket pendaftaran. Petugas pendaftaran mencari atau membuat kartu rekam medis pasien, lalu berkas tersebut diantar ke poli tujuan. Di poli, dokter atau tenaga medis melakukan pemeriksaan dan menuliskan keluhan, diagnosa, tindakan, serta resep pada lembar rekam medis. Setelah pelayanan selesai, berkas rekam medis dikembalikan ke ruang penyimpanan untuk diarsipkan.

4.1.2 Hasil Analisis Sistem yang Sedang Berjalan

Dari hasil observasi, pengelolaan data pasien di Puskesmas Batahan masih dilakukan secara manual menggunakan buku register dan berkas rekam medis berbentuk kertas yang disimpan di lemari arsip. Sebagian data rekapitulasi memang sudah dicatat dalam bentuk file Excel, tetapi file tersebut disimpan di komputer petugas tanpa pengamanan apa pun. Siapa saja yang bisa mengakses komputer tersebut otomatis bisa membaca seluruh isinya.

Ada beberapa hal yang langsung terlihat bermasalah dari sisi keamanan. Pertama, tidak ada pembatasan siapa yang boleh membuka data apa. Kedua, tidak ada catatan siapa yang pernah membuka atau mengubah data pasien, sehingga kalau terjadi kebocoran tidak bisa ditelusuri. Ketiga, data disimpan apa adanya dalam bentuk yang bisa langsung dibaca. Alur sistem yang sedang berjalan digambarkan pada Gambar 4.2.



Gambar 4.2 Flowmap Sistem yang Sedang Berjalan

4.1.3 Identifikasi Permasalahan Keamanan

Kelemahan yang ditemukan pada sistem berjalan kemudian dipetakan menjadi risiko dan ancaman yang mungkin terjadi. Pemetaan ini penting karena menjadi dasar kenapa fitur tertentu harus ada di sistem yang dibangun. Hasil pemetaannya ditampilkan pada Tabel 4.1.

Tabel 4.1 Identifikasi Permasalahan dan Risiko Keamanan

No	Kelemahan yang Ditemukan	Risiko yang Muncul	Solusi pada Sistem yang Dibangun
1	Data identitas dan rekam medis pasien disimpan dalam bentuk plaintext	Siapa pun yang berhasil masuk ke berkas atau basis data langsung bisa membaca seluruh data pasien	Enkripsi AES-128 mode CBC pada seluruh atribut sensitif sebelum data disimpan
2	Tidak ada pembatasan hak akses antar petugas	Petugas pendaftaran bisa membaca diagnosa pasien yang bukan wewenanganya	Kontrol akses berbasis peran (RBAC) dengan tiga peran: admin, dokter, dan petugas
3	Tidak ada pencatatan aktivitas pengguna	Kebocoran data tidak bisa ditelusuri siapa pelakunya	Tabel audit_log yang mencatat pengguna, aksi, tabel tujuan, alamat IP, dan waktu
4	Akses fisik ke berkas dan komputer tidak terkontrol	Berkas kertas bisa hilang, rusak, atau difoto oleh pihak yang tidak berhak	Pemindahan pengelolaan data ke sistem berbasis web dengan autentikasi wajib
5	Kata sandi (bila ada) disimpan	Kata sandi bisa terbaca langsung dan dipakai	Kata sandi disimpan sebagai hash bcrypt

No	Kelemahan yang Ditemukan	Risiko yang Muncul	Solusi pada Sistem yang Dibangun
	tanpa pengamanan	untuk menyamar sebagai pengguna lain	dengan cost factor 12, tidak bisa dibalik
6	Belum memenuhi ketentuan UU No. 27 Tahun 2022 tentang Perlindungan Data Pribadi	Puskesmas berpotensi terkena sanksi administratif karena data kesehatan tergolong data pribadi spesifik	Penerapan enkripsi, kontrol akses, dan pencatatan aktivitas sebagai bentuk upaya pengamanan data pribadi

4.1.4 Kebutuhan Fungsional dan Non-Fungsional

Berdasarkan hasil analisis di atas, kebutuhan sistem dirinci ulang dan diberi kode supaya nanti bisa dirujuk langsung oleh tabel pengujian black box pada sub-bab 4.6.1. Kode KF dipakai untuk kebutuhan fungsional dan KNF untuk kebutuhan non-fungsional. Daftar kebutuhan fungsional ditampilkan pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional Sistem

Kode	Kebutuhan Fungsional	Peran yang Berhak
KF-01	Sistem menyediakan proses login dengan verifikasi username dan kata sandi terhadap hash bcrypt yang tersimpan	Semua peran

Kode	Kebutuhan Fungsional	Peran yang Berhak
KF-02	Sistem menyediakan proses logout yang menghapus sesi pengguna	Semua peran
KF-03	Sistem membatasi akses halaman dan endpoint berdasarkan peran pengguna	Semua peran
KF-04	Sistem menyediakan manajemen pengguna (tambah, ubah peran, hapus)	Admin
KF-05	Sistem mengenkripsi atribut sensitif pasien dengan AES-128 sebelum disimpan ke basis data	Admin, Petugas
KF-06	Sistem menampilkan daftar pasien dalam bentuk yang sudah didekripsi	Semua peran
KF-07	Sistem menampilkan detail data seorang pasien	Semua peran
KF-08	Sistem dapat mengubah data pasien dan mengenkripsi ulang datanya	Admin, Petugas
KF-09	Sistem dapat menghapus data pasien	Admin
KF-10	Sistem menyediakan pencarian pasien berdasarkan nama atau nomor rekam medis	Semua peran
KF-11	Sistem mengenkripsi data rekam medis (keluhan, diagnosa, tindakan, resep) sebelum disimpan	Admin, Dokter

Kode	Kebutuhan Fungsional	Peran yang Berhak
KF-12	Sistem menampilkan riwayat rekam medis pasien dalam bentuk terdekripsi	Admin, Dokter
KF-13	Sistem dapat mengubah dan menghapus data rekam medis	Admin, Dokter
KF-14	Sistem mencatat seluruh aktivitas pengguna ke dalam tabel audit_log	Otomatis
KF-15	Sistem menampilkan log aktivitas pengguna	Admin
KF-16	Sistem menyediakan halaman demonstrasi enkripsi dan dekripsi AES-128	Semua peran
KF-17	Sistem menampilkan ringkasan data pada halaman dashboard	Semua peran

Sedangkan kebutuhan non-fungsional yang harus dipenuhi sistem ditampilkan pada Tabel 4.3.

Tabel 4.3 Kebutuhan Non-Fungsional Sistem

Kode	Aspek	Kebutuhan Non-Fungsional
KNF-01	Keamanan	Atribut sensitif pasien dan rekam medis dienkripsi dengan AES-128 mode CBC sebelum disimpan ke basis data

Kode	Aspek	Kebutuhan Non-Fungsional
KNF-02	Keamanan	Kunci AES-128 disimpan pada berkas .env, tidak ditulis langsung di dalam kode program, dan tidak diikutsertakan ke dalam repositori
KNF-03	Keamanan	Kata sandi pengguna disimpan sebagai hash bcrypt, bukan dienkripsi, sehingga tidak dapat dikembalikan ke bentuk asli
KNF-04	Keandalan	Hasil dekripsi harus identik dengan plaintext asli untuk seluruh data yang tersimpan
KNF-05	Kinerja	Waktu respons setiap operasi tidak lebih dari dua detik dan penambahan waktu akibat enkripsi tidak terasa oleh pengguna
KNF-06	Kegunaan	Antarmuka dapat digunakan tenaga kesehatan tanpa pelatihan teknis yang mendalam
KNF-07	Portabilitas	Sistem dapat diakses melalui peramban Chrome, Firefox, dan Edge tanpa instalasi tambahan
KNF-08	Integritas	Seluruh aksi terhadap data pasien dan rekam medis tercatat beserta identitas pengguna, waktu, dan alamat IP

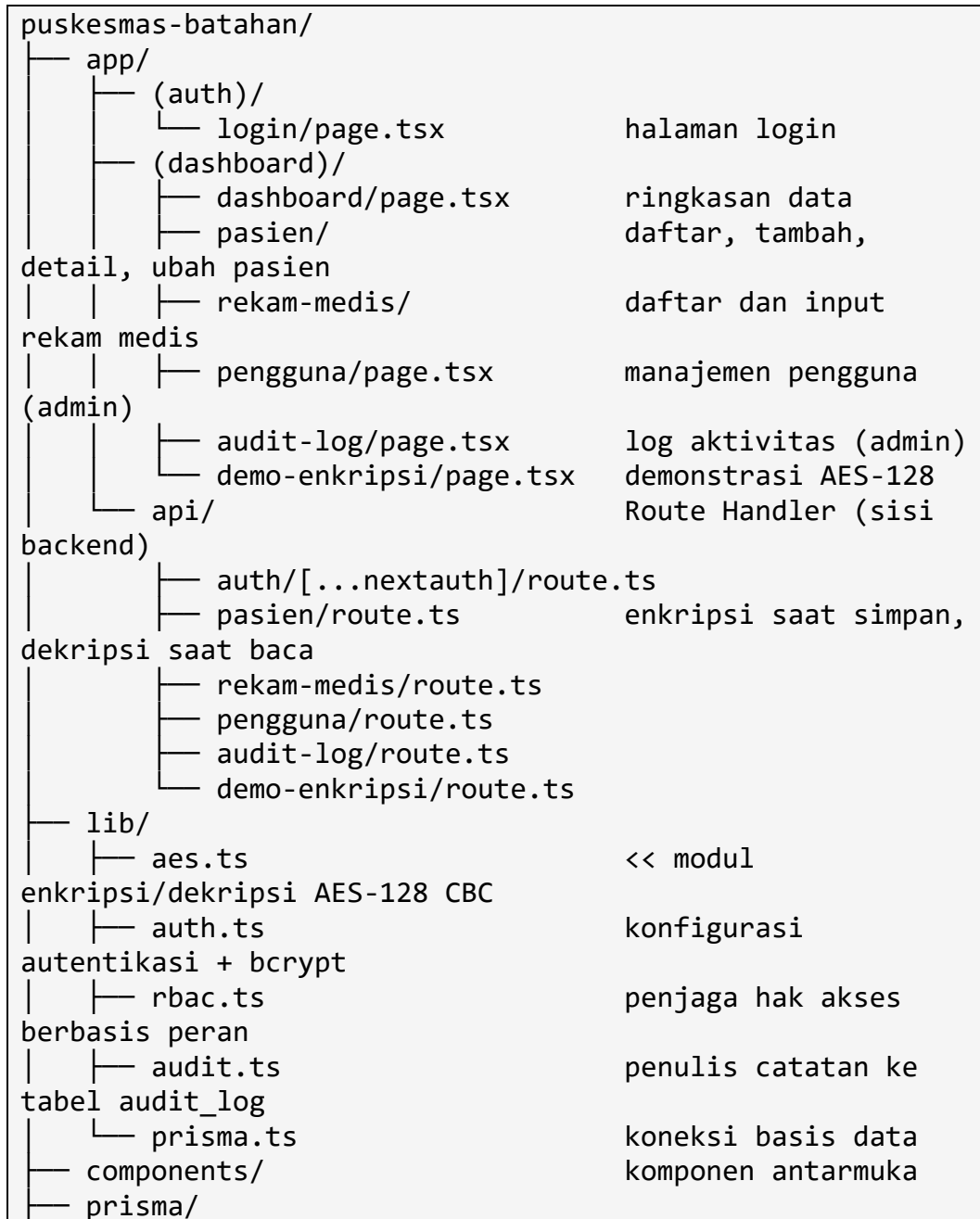
4.2 Implementasi Lingkungan Sistem

Sub-bab ini menjelaskan lingkungan tempat sistem dibangun dan diuji. Bagian ini bukan sekadar formalitas, karena hasil pengujian performa pada sub-bab

4.6.4 hanya bisa dibaca dengan benar kalau spesifikasi perangkat yang dipakai diketahui.

4.2.1 Struktur Direktori Aplikasi

Struktur direktori aplikasi disusun mengikuti konvensi Next.js App Router. Bagian yang berhubungan langsung dengan pengamanan data ditandai supaya mudah ditelusuri. Struktur direktorinya ditampilkan pada Gambar 4.3.



		schema.prisma	definisi struktur
basis data			
		migrations/	riwayat perubahan
struktur tabel			
		seed.ts	data awal pengguna
		middleware.ts	penjaga akses halaman
berdasarkan peran			
		.env	<< kunci AES-128 dan
kredensial basis data			
		package.json	

Gambar 4.3 Struktur Direktori Aplikasi

Dari struktur di atas terlihat bahwa seluruh logika kriptografi terkumpul pada satu berkas saja, yaitu `lib/aes.ts`. Pendekatan ini sengaja dipilih supaya kalau nanti algoritma atau modenya perlu diganti, perubahannya cukup dilakukan di satu tempat tanpa menyentuh berkas lain.

4.2.4 Konfigurasi Kunci dan Environment

Kunci AES-128 sepanjang 16 karakter tidak ditulis langsung di dalam kode program, melainkan disimpan pada berkas `.env` yang berada di luar repositori. Berkas ini sudah didaftarkan pada `.gitignore` sehingga tidak ikut terunggah ketika kode program dibagikan. Isi berkas `.env` dan cara pembacaannya ditampilkan pada Gambar 4.4.

```
# ----- berkas .env (tidak diikutsertakan ke
# repository) -----
DATABASE_URL="mysql://root:@localhost:3306/puskesmas_batahan"
AES_SECRET_KEY="PuskesmasBatahan"      # 16 karakter = 128 bit
AUTH_SECRET="<string acak untuk penandatanganan token sesi>"

# ----- lib/aes.ts : pembacaan dan validasi kunci -----
import CryptoJS from "crypto-js";

const SECRET_KEY = process.env.AES_SECRET_KEY;

if (!SECRET_KEY || SECRET_KEY.length !== 16) {
```

```

    throw new Error("AES_SECRET_KEY harus tepat 16 karakter
(128 bit)");
}

const KEY = CryptoJS.enc.Utf8.parse(SECRET_KEY);

```

Gambar 4.4 Konfigurasi Kunci AES-128 pada Berkas .env dan lib/aes.ts

Ada tiga hal yang perlu digarisbawahi dari potongan kode di atas. Pertama, panjang kunci diperiksa saat aplikasi dinyalakan. Kalau kuncinya bukan 16 karakter, aplikasi langsung berhenti dengan pesan kesalahan, bukan diam-diam memakai kunci yang salah panjang. Kedua, kunci diubah menjadi WordArray memakai CryptoJS.enc.Utf8.parse. Langkah ini wajib, karena kalau kunci dikirim ke crypto-js sebagai string biasa, pustaka tersebut akan memperlakukannya sebagai passphrase dan menurunkan kunci sendiri melalui fungsi derivasi, sehingga kunci yang benar-benar dipakai bukan 16 karakter yang kita tuliskan. Ketiga, Initialization Vector tidak ikut disimpan di .env karena IV memang dibangkitkan acak setiap kali proses enkripsi dijalankan.

4.3 Implementasi Basis Data

Basis data diimplementasikan pada MySQL 8.0 dengan nama puskesmas_batahan. Struktur tabel dibuat melalui berkas skema Prisma yang kemudian diterjemahkan menjadi perintah SQL melalui mekanisme migrasi. Empat tabel yang dirancang pada Bab III seluruhnya terwujud dengan nama kolom yang sama, yaitu users, pasien, rekam_medis, dan audit_log.

4.3.1 Struktur Tabel

Tabel users menyimpan data akun pengguna sistem beserta perannya. Strukturnya ditampilkan pada Tabel 4.4.

Tabel 4.4 Struktur Tabel users

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Kunci utama
2	username	VARCHAR(50)	Nama pengguna, bersifat unik
3	password	VARCHAR(255)	Hash bcrypt (cost 12), bukan hasil enkripsi
4	role	ENUM	admin, dokter, atau petugas

No	Nama Kolom	Tipe Data	Keterangan
5	created_at	TIMESTAMP	Waktu akun dibuat
6	updated_at	TIMESTAMP	Waktu akun terakhir diubah

Tabel pasien menyimpan data identitas pasien. Lima kolom di dalamnya menyimpan ciphertext, bukan data asli. Strukturnya ditampilkan pada Tabel 4.5.

Tabel 4.5 Struktur Tabel pasien

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Kunci utama
2	no_rm	VARCHAR(20)	Nomor rekam medis, unik, disimpan sebagai plaintext
3	nama	TEXT	Nama pasien — terenkripsi AES-128
4	nik	TEXT	Nomor Induk Kependudukan — terenkripsi AES-128
5	tanggal_lahir	TEXT	Tanggal lahir — terenkripsi AES-128
6	alamat	TEXT	Alamat tempat tinggal — terenkripsi AES-128
7	no_telepon	TEXT	Nomor telepon — terenkripsi AES-128
8	iv	VARCHAR(255)	Initialization Vector record ini (32 karakter heksadesimal)
9	created_by_id	INT (FK)	Referensi ke users.id, petugas yang mendaftarkan
10	created_at	TIMESTAMP	Waktu pendaftaran

Tabel rekam_medis menyimpan data hasil pemeriksaan pasien. Empat kolom di dalamnya berisi ciphertext. Strukturnya ditampilkan pada Tabel 4.6.

Tabel 4.6 Struktur Tabel rekam_medis

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Kunci utama
2	pasien_id	INT (FK)	Referensi ke pasien.id
3	dokter_id	INT (FK)	Referensi ke users.id
4	tanggal_periksa	DATETIME	Tanggal dan waktu pemeriksaan
5	keluhan	TEXT	Keluhan pasien — terenkripsi AES-128
6	diagnosa	TEXT	Diagnosa dokter — terenkripsi AES-128

No	Nama Kolom	Tipe Data	Keterangan
7	tindakan	TEXT	Tindakan medis — terenkripsi AES-128
8	resep	TEXT	Resep obat — terenkripsi AES-128
9	iv	VARCHAR(255)	Initialization Vector record ini
10	created_at	TIMESTAMP	Waktu pencatatan

Tabel audit_log menyimpan jejak aktivitas seluruh pengguna. Tabel ini tidak dienkripsi karena isinya memang metadata, bukan data pribadi pasien. Strukturnya ditampilkan pada Tabel 4.7.

Tabel 4.7 Struktur Tabel audit_log

No	Nama Kolom	Tipe Data	Keterangan
1	id	INT (PK, AI)	Kunci utama
2	user_id	INT (FK)	Referensi ke users.id, pelaku aktivitas
3	aksi	VARCHAR(50)	CREATE, READ, UPDATE, DELETE, LOGIN, atau LOGOUT
4	tabel_target	VARCHAR(50)	Nama tabel yang diakses
5	id_target	INT	Id record yang diakses, boleh kosong
6	ip_address	VARCHAR(45)	Alamat IP pengguna, mendukung IPv6
7	waktu	TIMESTAMP	Waktu aktivitas terjadi

Satu hal teknis yang perlu dijelaskan adalah pemilihan tipe data TEXT untuk kolom yang dienkripsi. Ciphertext AES yang disimpan dalam format Base64 jauh lebih panjang daripada data aslinya. Setiap 16 byte hasil enkripsi berubah menjadi 24 karakter Base64. Nama pasien sepanjang 11 karakter yang tadinya cukup disimpan pada VARCHAR(50), setelah dienkripsi menjadi 24 karakter. Alamat sepanjang 32 karakter membengkak menjadi 64 karakter. Kalau kolomnya tetap dibuat VARCHAR dengan panjang seperti data aslinya, data hasil enkripsi akan terpotong dan proses dekripsi otomatis gagal. Karena itu seluruh kolom terenkripsi dibuat bertipe TEXT yang mampu menampung sampai 65.535 byte.

4.3.2 Relasi Antar Tabel

Keempat tabel di atas saling berelasi seperti yang dirancang pada Gambar 3.4. Relasi yang terbentuk adalah users ke pasien (satu ke banyak), pasien ke rekam_medis (satu ke banyak), users ke rekam_medis (satu ke banyak), dan users ke audit_log (satu ke banyak).

Relasi antara pasien dan rekam_medis diberi aturan penghapusan berantai (ON DELETE CASCADE). Jadi ketika data seorang pasien dihapus, seluruh rekam medisnya ikut terhapus dan tidak meninggalkan data yatim di dalam basis data.

4.3.3 Penentuan Atribut yang Dienkripsi

Tidak semua kolom perlu dienkripsi. Kalau semuanya dienkripsi tanpa pertimbangan, sistem justru jadi tidak bisa dipakai karena kolom kunci dan kolom untuk pengurutan ikut teracak. Penentuan kolom mana yang dienkripsi dilakukan dengan melihat apakah kolom tersebut termasuk data pribadi menurut UU No. 27 Tahun 2022 dan apakah kolom tersebut dibutuhkan dalam bentuk terbaca untuk keperluan teknis. Hasil penentuannya ditampilkan pada Tabel 4.8.

Tabel 4.8 Penentuan Atribut yang Dienkripsi

Tabel	Atribut	Dienkripsi	Alasan
pasien	nama	Ya	Identitas langsung pasien, termasuk data pribadi umum
pasien	nik	Ya	Nomor identitas kependudukan, tergolong data pribadi spesifik
pasien	tanggal_lahir	Ya	Bila digabung dengan nama dapat mengidentifikasi seseorang secara langsung
pasien	alamat	Ya	Data pribadi yang menunjukkan lokasi tempat tinggal
pasien	no_telepon	Ya	Data pribadi yang dapat dipakai untuk menghubungi dan melacak pasien
pasien	no_rm	Tidak	Nomor internal puskesmas, dibutuhkan untuk pencarian cepat dan tidak menunjukkan identitas pasien

Tabel	Atribut	Dienkripsi	Alasan
pasien	id, created_at, created_by_id	Tidak	Kunci utama, kunci tamu, dan metadata; dibutuhkan untuk relasi, pengindeksan, dan pengurutan data
rekam_medis	keluhan, diagnosa, tindakan, resep	Ya	Data kesehatan yang tergolong data pribadi spesifik menurut Pasal 4 UU PDP
rekam_medis	tanggal_periksa	Tidak	Dibutuhkan untuk pengurutan riwayat dan penghitungan kunjungan harian
users	password	Tidak (di-hash bcrypt)	Verifikasi kata sandi tidak memerlukan bentuk aslinya, jadi cukup fungsi satu arah
users	username, role	Tidak	Dibutuhkan untuk pencarian akun saat login dan pemeriksaan hak akses
audit_log	seluruh kolom	Tidak	Berisi metadata aktivitas, bukan data pribadi pasien; harus tetap terbaca agar dapat diaudit

Bagian yang paling sering ditanyakan dari tabel di atas adalah kenapa kata sandi tidak ikut dienkripsi dengan AES padahal kata sandi jelas termasuk data sensitif. Jawabannya karena enkripsi dan hashing memang dibuat untuk tujuan yang berbeda.

Enkripsi adalah operasi dua arah. Data yang sudah dienkripsi bisa dikembalikan ke bentuk semula selama kuncinya diketahui. Sifat ini memang dibutuhkan untuk data pasien, karena dokter dan petugas harus bisa membaca nama, alamat, dan diagnosa pasien dalam bentuk aslinya. Sedangkan hashing adalah operasi satu arah. Keluaran fungsi hash tidak bisa dikembalikan menjadi masukan aslinya. Untuk kata sandi, sifat inilah yang justru diinginkan. Sistem sebenarnya tidak pernah perlu tahu kata sandi asli pengguna. Sistem hanya perlu tahu apakah kata sandi yang diketik saat login menghasilkan hash yang sama dengan yang tersimpan.

Kalau kata sandi disimpan dengan AES, maka siapa pun yang berhasil mendapatkan kunci AES otomatis bisa membaca seluruh kata sandi pengguna dalam bentuk asli. Dengan bcrypt hal itu tidak mungkin terjadi, karena bcrypt tidak

memakai kunci yang bisa dicuri, melainkan menambahkan salt acak pada setiap kata sandi dan mengulang proses hashing berkali-kali. Pada sistem ini bcrypt dijalankan dengan cost factor 12, artinya proses hashing diulang 2 pangkat 12 atau 4.096 kali. Perulangan ini membuat proses menebak kata sandi satu per satu menjadi sangat lambat bagi penyerang, sementara bagi pengguna yang login secara normal penambahan waktunya tidak terasa.

4.3.4 Perbandingan Data Sebelum dan Sesudah Enkripsi

Untuk membuktikan bahwa data benar-benar tersimpan dalam bentuk terenkripsi dan bukan sekadar disembunyikan di tampilan, satu record data uji pasien ditelusuri dari bentuk aslinya sampai bentuk yang tersimpan di basis data. Hasilnya ditampilkan pada Tabel 4.9. Seluruh atribut pada record ini dienkripsi memakai IV yang sama, yaitu f31edf3102ab651a5ffdf9501f11c5a4.

Tabel 4.9 Perbandingan Data Sebelum dan Sesudah Enkripsi

Kolom	Sebelum Enkripsi (plaintext)	Sesudah Enkripsi (ciphertext Base64)
nama	Siti Aminah	5PaRAeHEaZ24lA3YaaEbFA==
nik	1207054503010002	787qK+AxsLgj+skqbjURT7MueLXEG7URnoukvREJ3E8=
tanggal_lahir	1990-03-17	6aYOkf7eDjC5Rtnc4RqE7A==
alamat	Jl. Lintas Barat No. 21, Batahan	CAESU0C6lObgezMg/IS33I+WblgupmI2pGMo3+ORmeY4tIKdVqb5LgmWQt73KQ2c
no_telepon	081376542190	gDjPKcLI6xbWPUdwK8QDHA==

Terlihat jelas bahwa tidak ada satu pun potongan data asli yang masih bisa dikenali pada kolom ciphertext. Panjangnya pun tidak lagi mencerminkan panjang data asli, karena setiap data dibulatkan ke kelipatan 16 byte oleh padding sebelum dienkripsi. Nama Siti Aminah yang panjangnya 11 karakter dan tanggal lahir 1990-03-17 yang panjangnya 10 karakter sama-sama menghasilkan ciphertext sepanjang 24 karakter, sehingga panjang ciphertext tidak bisa dipakai untuk menebak isi datanya.

Isi tabel pasien pada basis data dilihat langsung melalui klien MySQL untuk memastikan yang tersimpan memang ciphertext. Hasilnya ditampilkan pada Gambar 4.5.

id	noRm	nama	nik	tanggalLahir	alamat	noTelepon	iv
1	RM-001	KNoS7d0srxWu42p8d9g==	KSYR1sga0pM1S1GnGh0jY57prjYvgpp+PbzrAKUCZY=	fRUwgyuZ+I1fA7G4/N/zkG==	LxKx5CI0oRH/2fcy6wV5Ng==	AJUVNqmp3LrKZMMUUCbw==	72ca664b14c75e8c3b778ea315733
2	RM-002	XuJootP5jXeeaujg0p8106F=	mXh4v7AvnDAt1v2gURFX/dQXvSnDC8oBfYodxSL4=	zDPVFSj50qbnx05/4BgBA=	Vu06MhrZc38818+p8ZA/FBy=	9LchL2ZrVINTZzm9fyVEw==	fCfFc8679bc1838c718e9ac3647c8ba4
3	RM-003	58Lfk2b2pa5wYxd1n/H10M=	ZsSHLP+3g7mKk21dRsyvSnKCK755740TnPP01WjCB=	zYCP2M04QRAu0LgcnZq8Iw=	Heyxb1d5gwCczjntFs7N8Q=	Dglw4+d8lpFeVkyKyrQRg=	97a88cdac9826f88c2485da269f52ea
4	RM-004	f761o+JzbP69BLXUJFxcgen=	q0pv5xXj+rSHeBOXzhewWPPYA+End2pw553KjvFKctnE=	LyFHBMatb/JK29VF+zYK9A=	rHfLXZ625U51+zVItzn+1t/_	39fZpYfpb8Knd/KYNNVbcg=	3da6687bea66b9d71d1e3b868275261
5	RM-005	I2dAlpG9pKKh9wzydwcl=	ApH6H7KnJR1n7ychu1DL5896AsfhL8GyUuMAJ6Lk=	XUJY6Kp25x0N708LjNysM4Q=	z9rU9MksbxUa9jDVer0W9=	48tZbwL/umfML8Dggy9EYQ=	8e5c65fefb8ddaca7193056a725f130
6	RM-006	WzJGMBL09AKFYjh7daZAlV=	YNT9SKqEYBONN+9fwJNxoX8p9hNNBoIJCD5z1Xzc8B=	kWM3MuFTc5D03amGxro0g=	qbkd9R/9phXp7Mec+rs58eT=	fnZBRF7U6M5kQYDf3wpf0w=	3e550f1e6dddea68b641be96838e5
7	RM-007	AzM286FShqbFSj0IsuarCIY=	x+m0J/H1531betfbFzTxd4XDdly+bVrgmXstcRRP1k=	a5odMP0ARmb+511w5L136Q=	898R92d+Tdxef0R/1117Q6=	puITovZAc1zhjzeEhNYMA=	3ebfc11fe4348e5f6dbc984838739596
8	RM-008	56fI6+19f6J3YS1y2Z/CY=	If2Ht18R5PxmVj0Jf/Rn1KaddyYqJGEIAIav52QYI=	21D18cX1C0e10K11866nlg=	nEUyq7t+HL8NVRr/6+n7G=	aj33jrn1bW6ldQn/na0Ww=	68933b7f65f01f33a78ec5ce79d3072e
9	RM-009	Fv2a/DLQ2P3Jtyg/yuxYhFJL=	6+hj0j0mgRd/1e8b+f6s1zudfAawjGCAnJ2f4t128=	158fYTYUZO/2zH0t1bEw=	9p5m0JGS+h06uz+3L11ek=	t00mj+0MBTfjYhnzr5w=	75bf77c6fef39758d44f4ccbe7a83dc
10	RM-010	1DwGqL3p5r+Ht12mkg0M=	hPanPB0dc11/W3LcSRWjyvJ8XUj3rPs0N24k808vA=	fu9RvyaYKEAbxQfV8QWg=	43CKLBTvYL0+haPgT4kPL=	8f66ryCnEa3QMBUj03Nw=	9045e4bc82da73f5496562736564d88e
11	RM-011	+W0mPNJt0C348zpv15ta=	z03fuJNChYH9U16Td1aHjmKf87LYqbaJVdFXCqQ=	TSj6HEU7t/kjbstWbYtA=	0T1x95940pYf50N4XB01Bw=	l6scwp+Hf20H15uueX8onw=	42be0c3804df1f87d37e2ab27d8136fb
12	RM-012	LKp2T5t1f5CUD3KLKTLV7R/87TnZeGw5J8ckdQ/zv06KqfNw=	UncVG1SLTKTLV7R/87TnZeGw5J8ckdQ/zv06KqfNw=	Y08zJx55Um17gK8KUUSQ=	59H22xR815gch7Lckso5C+P=	0+e5GA0oovs989G6NLLBa=	e0c61570a6b752f09a9f5f72b08f058a
13	RM-013	t5B7wL0hCq40d0mTo1WraJ=	cqhceU4Dvdpdoy8WnTdn1ST1rq3ev2bosm5p5Um=	K051D317qk1amXdmUKTA=	byfzeMPzb48XAKT1087TF8=	pxeQLzBTod3v9CpwQHEJw=	deac25a1a3a795c00b6143b00b9f7643
14	RM-014	4eAlu/uAE6hp0M/GoZIK4LW=	zsktVBEeEP/+C1AloguE7CjYpdlUMK3map0JaTu7TJU=	220dMqgFFtC85/7ehcSJ6w=	/X+SCLV6edtpgh8G6v1be3=	htKb/6TA3js98sR6ZJQJg=	cd9f89e3db83268d8eebaada7886986c
15	RM-015	Idpcfn0dduNA13LB6A6ZQ=	+HAY0+hCzMAks4pAZC1Kj2f6/PqcLbTPva9vgx9UFFB=	MDz0da3AQGLCYMGcQLmBhg=	/dyKaT8d10eq3UXSNEJ5yvx=	D0rG6V92Rn40W75otYoCa=	e314d726170513d9c939d2c75d69297f
16	RM-016	qGhqbH7EjJH0f5K5A8a5KQ=	Y48Ae+daQME0702Gn2byQLW6Bf8kgzFKAhX8J1D0H=	K1fZRYabn+QRpG028C4eA=	fepTHZ8jffn61nkhqIvxoL=	Muum+lbsS812qgFZ68Hpg=	918e32f3c1d3771af4d7ee143395f65
17	RM-017	rlgKbnZa4zJ7n08nZ4v0=	M+2oLmZwzavUHPJ20qH1W+4ggUmpL8avSXGfus=	1428CQ4w08F5B1YvCUJIA=	1uFnIda4k3eC2E6y1E3VtL=	uzunGLDPCnc/TWMS3s3Ng=	39e119b3f6b543247547f54ef68376c6
18	RM-018	MntSDmWraK8sef6E/ROH=	80/0tnJRS9Qd3cehW8QymSH35ty66G1CZVb0nldg=	7TR0Kf52M2LoWNEPFGnQ=	+2Kxf1ca3k3eC0T8FRhpqBL=	/aK1c300s48h4CtMhC1Q=	4707f67073cf68b77578c8ca3683c4
19	RM-019	oTL8IXj40a15zwfXCjZT6A=	CHR0q1wK2uma1am6jV1bCfXK8fjw7NWkaic1Qvck=	9ry7FKdx50Cfj3L5hya2w=	CYszsfo3EVCbgh07AZas8Xj=	ASzwX8hr1NpuU+10T8du1g=	a8e351e38eb177Cef79abccdb4966815c

Gambar 4.5 Isi Tabel pasien pada Basis Data MySQL

Kalau pun basis data ini bocor atau salinannya dicuri, yang didapat pihak tersebut hanya deretan karakter acak. Tanpa kunci AES-128 yang tersimpan terpisah pada berkas .env di server, data tersebut tidak bisa dikembalikan ke bentuk aslinya.

4.4 Implementasi Algoritma AES-128

Sub-bab ini merupakan inti dari penelitian. Bagian ini menjelaskan bagaimana algoritma AES-128 benar-benar bekerja pada data pasien, lengkap dengan perhitungan manual satu blok penuh sebanyak sepuluh putaran. Perhitungan manual ini dilakukan untuk membuktikan bahwa penerapan AES-128 pada sistem tidak sekadar memanggil fungsi dari pustaka, tetapi memang dipahami langkah demi langkahnya.

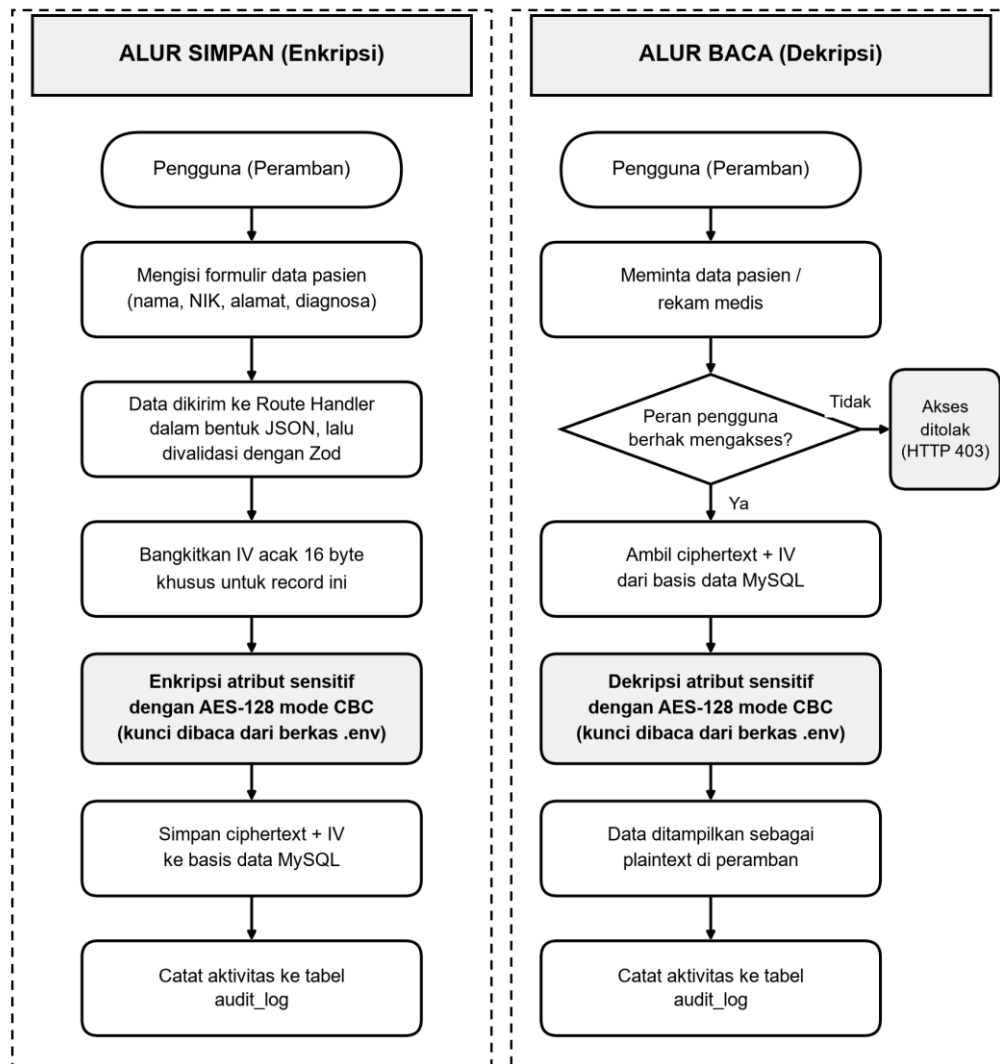
Seluruh angka pada sub-bab ini sudah diverifikasi ulang menggunakan skrip Python yang mengimplementasikan AES-128 dari nol. Skrip tersebut lebih dulu diuji terhadap vektor uji resmi NIST FIPS 197 dan hasilnya cocok, sehingga angka perhitungan manual di bawah ini dapat dipastikan benar.

4.4.1 Skema Umum Proses Pengamanan Data

Sebelum masuk ke perhitungan, ada baiknya dilihat dulu di titik mana proses enkripsi terjadi pada sistem. Ada dua alur utama, yaitu alur simpan dan alur baca.

Pada alur simpan, pengguna mengisi formulir data pasien di peramban. Data dikirim ke Route Handler dalam bentuk JSON, lalu divalidasi menggunakan Zod untuk memastikan formatnya benar, misalnya NIK harus tepat 16 digit. Setelah lolos validasi, sistem membangkitkan satu IV acak sepanjang 16 byte untuk record tersebut, mengenkripsi seluruh atribut sensitifnya dengan AES-128 mode CBC, lalu menyimpan ciphertext beserta IV-nya ke basis data. Terakhir, aktivitas ini dicatat ke tabel `audit_log`.

Pada alur baca, sistem mengambil record dari basis data beserta IV yang tersimpan, mendekripsi atribut yang terenkripsi memakai kunci dan IV tersebut, lalu mengirimkan hasilnya ke peramban. Sebelum data dikirim, sistem memeriksa dulu apakah peran pengguna yang meminta memang berhak mengakses data tersebut. Skema lengkap kedua alur ditampilkan pada Gambar 4.6.



Kunci AES-128 disimpan pada berkas .env di luar kode program dan tidak ikut tersimpan di dalam basis data, sehingga ciphertext yang bocor tidak dapat dikembalikan ke bentuk aslinya tanpa kunci tersebut.

Gambar 4.6 Skema Umum Proses Pengamanan Data pada Sistem

4.4.2 Pembangkitan Kunci (Key Expansion)

AES-128 tidak memakai kunci yang sama berulang-ulang pada setiap putaran. Kunci utama sepanjang 128 bit lebih dulu dikembangkan menjadi sebelas kunci putaran (round key), yaitu satu kunci untuk AddRoundKey awal ditambah sepuluh kunci untuk sepuluh putaran transformasi. Proses inilah yang disebut key expansion.

Kunci yang dipakai pada contoh perhitungan ini adalah **PuskesmasBatahan** yang panjangnya tepat 16 karakter atau 128 bit. Dalam bentuk heksadesimal, kunci tersebut menjadi:

50 75 73 6B 65 73 6D 61 73 42 61 74 61 68 61 6E

Ke-16 byte tersebut disusun menjadi matriks 4×4 yang diisi kolom per kolom, bukan baris per baris. Susunan inilah yang disebut state kunci awal dan sekaligus menjadi round key ke-0. Susunannya sebagai berikut.

50	65	73	61
75	73	42	68
73	6D	61	61
6B	61	74	6E

Gambar 4.7 Susunan State Kunci Awal (Round Key 0)

Setiap kolom pada matriks di atas disebut satu word dan diberi nama W0 sampai W3. Jadi W0 = 50 75 73 6B, W1 = 65 73 6D 61, W2 = 73 42 61 74, dan W3 = 61 68 61 6E. Dari keempat word ini akan dibangkitkan 40 word berikutnya sampai W43, sehingga totalnya 44 word atau sebelas round key.

Aturan pembangkitannya ada dua. Kalau nomor word habis dibagi empat, word sebelumnya harus melewati tiga operasi dulu yaitu RotWord, SubWord, dan XOR dengan Rcon, baru kemudian di-XOR dengan word empat posisi sebelumnya. Kalau nomor word tidak habis dibagi empat, cukup XOR biasa antara word sebelumnya dengan word empat posisi sebelumnya. Berikut perhitungan lengkap untuk W4 sampai W7 yang membentuk round key 1.

a. Perhitungan W4

Karena 4 habis dibagi 4, maka W3 harus melewati tiga operasi terlebih dahulu.

W3	=	61 68 61 6E
1) RotWord	:	geser satu byte ke kiri secara memutar 61 68 61 6E -> 68 61 6E 61
2) SubWord	:	substitusi tiap byte dengan tabel S-Box

```

68 -> baris 6 kolom 8 -> 45
61 -> baris 6 kolom 1 -> EF
6E -> baris 6 kolom E -> 9F
61 -> baris 6 kolom 1 -> EF
hasil          = 45 EF 9F EF

3) XOR Rcon : Rcon putaran 1 = 01 00 00 00
45 EF 9F EF XOR 01 00 00 00 = 44 EF 9F EF

4) XOR W0 : W4 = hasil langkah 3 XOR W0
44 EF 9F EF XOR 50 75 73 6B = 14 9A EC 84

>> W4 = 14 9A EC 84

```

Gambar 4.8 Perhitungan Word Keempat (W4) pada Key Expansion

b. Perhitungan W5, W6, dan W7

Nomor 5, 6, dan 7 tidak habis dibagi empat, jadi perhitungannya cukup XOR biasa tanpa melewati RotWord, SubWord, maupun Rcon.

```

W5 = W4 XOR W1 = 14 9A EC 84 XOR 65 73 6D 61 = 71 E9
81 E5
W6 = W5 XOR W2 = 71 E9 81 E5 XOR 73 42 61 74 = 02 AB
E0 91
W7 = W6 XOR W3 = 02 AB E0 91 XOR 61 68 61 6E = 63 C3
81 FF

```

Dengan demikian round key 1 adalah gabungan W4 sampai W7, yaitu 14 9A EC 84 71 E9 81 E5 02 AB E0 91 63 C3 81 FF.

Langkah yang sama diulang untuk W8 sampai W43. Nilai Rcon yang dipakai berbeda pada setiap kelipatan empat, yaitu 01, 02, 04, 08, 10, 20, 40, 80, 1B, dan 36 dalam heksadesimal. Karena langkahnya berulang, hasil lengkap kesebelas round key langsung disajikan pada Tabel 4.10 tanpa menuliskan seluruh perhitungannya.

Tabel 4.10 Hasil Key Expansion: Round Key 0 sampai Round Key 10

Round Key	Nilai (heksadesimal)	Rcon
RK0	50 75 73 6B 65 73 6D 61 73 42 61 74 61 68 61 6E	—
RK1	14 9A EC 84 71 E9 81 E5 02 AB E0 91 63 C3 81 FF	01
RK2	38 96 FA 7F 49 7F 7B 9A 4B D4 9B 0B 28 17 1A F4	02

Round Key	Nilai (heksadesimal)	Rcon
RK3	CC 34 45 4B 85 4B 3E D1 CE 9F A5 DA E6 88 BF 2E	04
RK4	00 3C 74 C5 85 77 4A 14 4B E8 EF CE AD 60 50 E0	08
RK5	C0 6F 95 50 45 18 DF 44 0E F0 30 8A A3 90 60 6A	10
RK6	80 BF 97 5A C5 A7 48 1E CB 57 78 94 68 C7 18 FE	20
RK7	06 12 2C 1F C3 B5 64 01 08 E2 1C 95 60 25 04 6B	40
RK8	B9 E0 53 CF 7A 55 37 CE 72 B7 2B 5B 12 92 2F 30	80
RK9	ED F5 57 06 97 A0 60 C8 E5 17 4B 93 F7 85 64 A3	1B
RK10	4C B6 5D 6E DB 16 3D A6 3E 01 76 35 C9 84 12 96	36

Terlihat bahwa mulai round key 1 tidak ada lagi potongan kunci asli yang bisa dikenali. Inilah salah satu alasan AES sulit diserang: mengetahui satu round key saja tidak langsung memberi tahu kunci utamanya, karena setiap round key sudah melewati substitusi non-linear melalui S-Box.

4.4.3 Proses Enkripsi

Plaintext yang dipakai pada contoh perhitungan ini adalah NIK data uji **1207054503010002**. NIK dipilih karena panjangnya tepat 16 karakter, sama dengan ukuran satu blok AES, sehingga perhitungannya bisa dilakukan dalam satu blok penuh tanpa perlu penambahan padding. Dalam bentuk heksadesimal, plaintext tersebut menjadi:

31 32 30 37 30 35 34 35 30 33 30 31 30 30 30 32

Sama seperti kunci, ke-16 byte plaintext disusun menjadi matriks state 4×4 secara kolom per kolom seperti berikut.

31	30	30	30
32	35	33	30
30	34	30	30
37	35	31	32

Gambar 4.9 Susunan State Plaintext Awal

a. AddRoundKey Putaran 0

Sebelum masuk ke putaran pertama, state plaintext lebih dulu di-XOR dengan round key 0. Operasi ini disebut initial round atau AddRoundKey putaran

0. Karena XOR dilakukan byte per byte pada posisi yang sama, perhitungannya cukup sederhana.

Tabel 4.11 Hasil AddRoundKey Putaran 0

Uraian	Nilai (heksadesimal)
State plaintext	31 32 30 37 30 35 34 35 30 33 30 31 30 30 30 32
Round key 0	50 75 73 6B 65 73 6D 61 73 42 61 74 61 68 61 6E
Hasil XOR	61 47 43 5C 55 46 59 54 43 71 51 45 51 58 51 5C

Sebagai contoh cara membacanya, byte pertama plaintext adalah 31 dan byte pertama round key 0 adalah 50. Bila keduanya di-XOR dalam bentuk biner, 0011 0001 XOR 0101 0000 menghasilkan 0110 0001 atau 61 dalam heksadesimal. Byte kedua, 32 XOR 75, menghasilkan 47. Begitu seterusnya sampai byte keenam belas.

b. Putaran 1 Secara Lengkap

Setiap putaran AES terdiri dari empat transformasi berurutan, yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Berikut penjabaran lengkap putaran 1 dengan state masukan 61 47 43 5C 55 46 59 54 43 71 51 45 51 58 51 5C.

1) SubBytes

Setiap byte pada state diganti dengan nilai lain berdasarkan tabel S-Box. Cara membacanya, nibble tinggi (digit heksadesimal pertama) dipakai sebagai nomor baris dan nibble rendah (digit kedua) sebagai nomor kolom. Byte 61 misalnya, dibaca pada baris 6 kolom 1, dan nilai pada posisi tersebut adalah EF. Byte 47 dibaca pada baris 4 kolom 7, hasilnya A0. Contoh lain yang sering dipakai, byte 53 dibaca pada baris 5 kolom 3 dan menghasilkan ED. Tabel S-Box lengkapnya dapat dilihat pada Gambar 2.9 dan pada bagian lampiran.

Tabel 4.12 Hasil Transformasi SubBytes pada Putaran 1

Uraian	Nilai (heksadesimal)
State sebelum SubBytes	61 47 43 5C 55 46 59 54 43 71 51 45 51 58 51 5C
State sesudah SubBytes	EF A0 1A 4A FC 5A CB 20 1A A3 D1 6E D1 6A D1 4A

2) ShiftRows

Transformasi ini menggeser baris-baris matriks state ke kiri secara memutar. Baris pertama tidak digeser, baris kedua digeser satu byte, baris ketiga digeser dua byte, dan baris keempat digeser tiga byte. Tujuannya supaya byte-byte yang tadinya berada dalam satu kolom menyebar ke kolom yang berbeda. Ilustrasi pergeserannya ditampilkan pada Gambar 4.10.

SEBELUM ShiftRows					SESUDAH							
ShiftRows												
baris 0		EF	FC	1A	D1		tetap		EF	FC	1A	D1
baris 1		A0	5A	A3	6A		<< 1 byte		5A	A3	6A	A0
baris 2		1A	CB	D1	D1		<< 2 byte		D1	D1	1A	CB
baris 3		4A	20	6E	4A		<< 3 byte		4A	4A	20	6E

Gambar 4.10 Ilustrasi Transformasi ShiftRows pada Putaran 1

Dibaca kembali secara kolom per kolom, state hasil ShiftRows menjadi EF 5A D1 4A FC A3 D1 4A 1A 6A 1A 20 D1 A0 CB 6E.

3) MixColumns

MixColumns mengalikan setiap kolom state dengan matriks tetap di dalam medan hingga $GF(2^8)$. Matriks pengalinya adalah baris 02 03 01 01, 01 02 03 01, 01 01 02 03, dan 03 01 01 02. Transformasi inilah yang paling berperan menyebarkan pengaruh satu byte ke seluruh kolom, sehingga perubahan satu bit pada masukan cepat menjalar ke banyak bit keluaran.

Perkalian di $GF(2^8)$ tidak sama dengan perkalian biasa. Perkalian dengan 02 dilakukan dengan menggeser bit ke kiri satu posisi, lalu bila bit paling kiri sebelum digeser bernilai 1, hasilnya di-XOR dengan 1B. Perkalian dengan 03 sama dengan hasil perkalian 02 di-XOR dengan bilangan aslinya, karena $03 = 02 \text{ XOR } 01$. Berikut perhitungan lengkap untuk kolom pertama.

Kolom pertama hasil ShiftRows : EF 5A D1 4A				
Perkalian yang dibutuhkan:				
02 x EF = C5	03 x EF =	02 x EF XOR EF =	C5 XOR EF =	2A
02 x 5A = B4	03 x 5A =	B4 XOR 5A =	EE	
02 x D1 = B9	03 x D1 =	B9 XOR D1 =	68	
02 x 4A = 94	03 x 4A =	94 XOR 4A =	DE	
Baris 0 : (02 x EF) XOR (03 x 5A) XOR (01 x D1) XOR (01 x 4A)				
	= C5	XOR	EE	XOR
4A	= B0		D1	XOR
Baris 1 : (01 x EF) XOR (02 x 5A) XOR (03 x D1) XOR (01 x 4A)				
	= EF	XOR	B4	XOR
4A	= 79		68	XOR
Baris 2 : (01 x EF) XOR (01 x 5A) XOR (02 x D1) XOR (03 x 4A)				
	= EF	XOR	5A	XOR
DE	= D2		B9	XOR
Baris 3 : (03 x EF) XOR (01 x 5A) XOR (01 x D1) XOR (02 x 4A)				
	= 2A	XOR	5A	XOR
94	= 35		D1	XOR
Kolom pertama hasil MixColumns : B0 79 D2 35				

Gambar 4.11 Perhitungan MixColumns pada Kolom Pertama Putaran 1

Sebagai contoh cara menghitung $02 \times EF$, nilai EF dalam biner adalah 1110 1111. Karena bit paling kiri bernilai 1, hasil pergeseran ke kiri harus di-XOR dengan 1B. Pergeseran menghasilkan 1101 1110, lalu $1101 \ 1110 \text{ XOR } 0001 \ 1011 = 1100 \ 0101$ atau C5 dalam heksadesimal.

Perhitungan yang sama diterapkan pada kolom kedua, ketiga, dan keempat. Hasil lengkap seluruh kolom setelah MixColumns adalah B0 79 D2 35 86 83 38 F9 B0 C0 24 1E E7 A2 4E DF.

4) AddRoundKey

Langkah terakhir pada setiap putaran adalah XOR antara state dengan round key putaran tersebut. Untuk putaran 1, round key yang dipakai adalah RK1 hasil key expansion pada Tabel 4.10.

Tabel 4.13 Hasil AddRoundKey Putaran 1

Uraian	Nilai (heksadesimal)
State hasil MixColumns	B0 79 D2 35 86 83 38 F9 B0 C0 24 1E E7 A2 4E DF
Round key 1	14 9A EC 84 71 E9 81 E5 02 AB E0 91 63 C3 81 FF
State keluaran putaran 1	A4 E3 3E B1 F7 6A B9 1C B2 6B C4 8F 84 61 CF 20

c. Putaran 2 sampai Putaran 9

Putaran 2 sampai 9 menjalankan urutan transformasi yang sama persis dengan putaran 1, hanya round key-nya yang berbeda mengikuti Tabel 4.10. Karena langkahnya berulang, penjabaran per operasi tidak diulang lagi dan hasil state tiap putaran langsung disajikan pada Tabel 4.14.

d. Putaran 10

Putaran terakhir punya satu perbedaan penting: transformasi MixColumns dihilangkan. Jadi putaran 10 hanya menjalankan SubBytes, ShiftRows, dan AddRoundKey. Alasannya bukan untuk menghemat waktu, melainkan supaya proses dekripsi bisa disusun simetris dengan proses enkripsi. Kalau MixColumns tetap dijalankan pada putaran terakhir, urutan operasi pada proses dekripsi akan menjadi tidak seragam dan implementasinya justru lebih rumit. Perlu ditegaskan bahwa penghilangan MixColumns di putaran akhir tidak mengurangi tingkat

keamanan AES, karena difusi sudah terbentuk penuh melalui sembilan putaran sebelumnya.

Tabel 4.14 State Hasil Setiap Putaran pada Proses Enkripsi

Putaran	State Keluaran (heksadesimal)
Putaran 0 (AddRoundKey awal)	61 47 43 5C 55 46 59 54 43 71 51 45 51 58 51 5C
Putaran 1	A4 E3 3E B1 F7 6A B9 1C B2 6B C4 8F 84 61 CF 20
Putaran 2	07 48 4B CF 5A A4 20 5C 21 77 83 2C 80 E3 6D B4
Putaran 3	E7 C1 86 BB 50 CA 88 CE E5 C4 EA F0 57 52 C6 65
Putaran 4	65 9F 56 0B 59 31 53 86 2D 76 38 F7 DB 10 6F B2
Putaran 5	38 89 48 29 03 6B 1A C4 15 2B 97 0A 39 C0 7B 5C
Putaran 6	CD 8F 3A 30 BF E3 74 12 E2 8B 1B AF 7B 1A 5B 03
Putaran 7	80 1C 48 8B A9 88 2F 15 97 77 7D 8D 5C 36 E2 1B
Putaran 8	3F 0B 55 FD 66 F9 7D B7 7D 9A E9 3D 71 99 34 DD

Putaran	State Keluaran (heksadesimal)
Putaran 9	68 4A DF 87 6E 84 27 95 7C 99 59 D2 0F 4D 60 C7
Putaran 10 (tanpa MixColumns)	09 E9 96 A8 44 F8 ED B1 2E E2 E8 1F BF 52 DE 23

e. Ciphertext Akhir

State keluaran putaran 10 adalah ciphertext dari blok tersebut, yaitu:

09 E9 96 A8 44 F8 ED B1 2E E2 E8 1F BF 52 DE 23

Ditulis rapat tanpa spasi menjadi 09E996A844F8EDB12EE2E81FBF52DE23. Ciphertext dalam bentuk heksadesimal ini kemudian dikodekan ke Base64 sebelum disimpan ke basis data, karena Base64 hanya memakai karakter yang aman untuk kolom teks dan untuk pengiriman melalui HTTP. Hasil pengodean Base64-nya adalah:

CemWqET47bEu4ugfv1LeIw==

Perlu dicatat bahwa perhitungan manual di atas dilakukan pada satu blok AES murni tanpa mode operasi, sehingga hasilnya bisa dibandingkan langsung dengan vektor uji standar. Pada sistem yang sebenarnya, blok ini masih akan di-XOR terlebih dahulu dengan Initialization Vector sesuai aturan mode CBC yang dijelaskan pada sub-bab 4.4.5, sehingga ciphertext yang tersimpan di basis data berbeda pada setiap record meskipun plaintext-nya sama.

4.4.4 Proses Dekripsi

Proses dekripsi menjalankan transformasi kebalikan dari enkripsi dengan urutan yang juga dibalik. Kalau enkripsi berjalan dari round key 0 menuju round key 10, maka dekripsi berjalan dari round key 10 mundur ke round key 0. Transformasi yang dipakai adalah InvShiftRows, InvSubBytes, AddRoundKey, dan InvMixColumns.

InvShiftRows menggeser baris ke kanan, bukan ke kiri, dengan jumlah pergeseran yang sama. InvSubBytes memakai tabel Inverse S-Box yang merupakan kebalikan dari S-Box, misalnya byte EF pada Inverse S-Box akan kembali menjadi 61. AddRoundKey tidak punya versi kebalikan karena operasi XOR memang sudah merupakan kebalikan dari dirinya sendiri, artinya $A \oplus B \oplus B$ akan kembali menjadi A . Sedangkan InvMixColumns memakai matriks pengali 0E 0B 0D 09 yang merupakan invers dari matriks 02 03 01 01 dalam $GF(2^8)$.

Proses dekripsi dimulai dengan AddRoundKey memakai round key 10 pada ciphertext.

Tabel 4.15 Hasil AddRoundKey Awal pada Proses Dekripsi

Uraian	Nilai (heksadesimal)
Ciphertext	09 E9 96 A8 44 F8 ED B1 2E E2 E8 1F BF 52 DE 23
Round key 10	4C B6 5D 6E DB 16 3D A6 3E 01 76 35 C9 84 12 96
Hasil XOR	45 5F CB C6 9F EE D0 17 10 E3 9E 2A 76 D6 CC B5

Putaran berikutnya menjalankan InvShiftRows, lalu InvSubBytes, kemudian AddRoundKey dengan round key 9, dan terakhir InvMixColumns. Urutan ini diulang sampai putaran kesepuluh dengan round key yang terus mundur. Sama seperti pada enkripsi, putaran terakhir dekripsi tidak menjalankan InvMixColumns. Hasil state setiap putaran ditampilkan pada Tabel 4.16.

Tabel 4.16 State Hasil Setiap Putaran pada Proses Dekripsi

Putaran	State Keluaran (heksadesimal)
AddRoundKey awal (RK10)	45 5F CB C6 9F EE D0 17 10 E3 9E 2A 76 D6 CC B5

Putaran	State Keluaran (heksadesimal)
Putaran 1 (RK9)	75 99 1E C1 33 B8 18 54 FF EE FC A9 A3 2B FF 27
Putaran 2 (RK8)	CD C4 FF AF D3 F5 98 3D 88 05 52 59 4A 9C 15 5D
Putaran 3 (RK7)	BD 11 AF 7B 08 3D 39 04 98 A2 80 C9 21 73 92 79
Putaran 4 (RK6)	07 7F 88 4A 7B F1 21 A5 59 BA 52 1C 12 A7 A2 67
Putaran 5 (RK5)	4D C7 07 37 CB 38 A8 2B D8 CA B1 44 B9 DB ED 68
Putaran 6 (RK4)	94 74 87 4D 53 1C B4 EA D9 00 44 8B 5B 78 C4 8C
Putaran 7 (RK3)	C5 49 EC 8D BE F5 3C 8A FD 11 B3 4A CD 52 B7 71
Putaran 8 (RK2)	49 02 1C B7 68 7F 8A C8 37 EF B2 9C 5F 11 56 73
Putaran 9 (RK1)	EF 5A D1 4A FC A3 D1 4A 1A 6A 1A 20 D1 A0 CB 6E
Putaran 10 (RK0, tanpa InvMixColumns)	31 32 30 37 30 35 34 35 30 33 30 31 30 30 30 32

State keluaran putaran 10 adalah 31 32 30 37 30 35 34 35 30 33 30 31 30 30 30 32. Kalau setiap byte tersebut diterjemahkan kembali menjadi karakter ASCII, hasilnya adalah 31 menjadi karakter 1, 32 menjadi 2, 30 menjadi 0, 37 menjadi 7, dan seterusnya, sehingga seluruhnya membentuk kembali NIK 1207054503010002 yang persis sama dengan plaintext semula. Ini membuktikan bahwa proses enkripsi dan dekripsi berjalan konsisten dan tidak ada informasi yang hilang.

Ada satu hal menarik yang bisa diamati dari Tabel 4.16. State pada putaran 9 dekripsi, yaitu EF 5A D1 4A FC A3 D1 4A 1A 6A 1A 20 D1 A0 CB 6E, nilainya sama persis dengan state hasil ShiftRows pada putaran 1 enkripsi. Kesamaan ini bukan kebetulan, melainkan bukti bahwa proses dekripsi memang benar-benar menelusuri balik jalur yang sama dengan proses enkripsi.

4.4.5 Penerapan Mode Operasi CBC

AES pada dasarnya hanya bisa mengenkripsi satu blok berukuran 16 byte. Padahal data yang disimpan sistem ini panjangnya bermacam-macam. Alamat pasien bisa 30 sampai 50 karakter, sedangkan diagnosa bisa lebih panjang lagi. Untuk itu dibutuhkan mode operasi yang mengatur bagaimana blok-blok tersebut dirangkai. Sistem ini memakai mode Cipher Block Chaining (CBC).

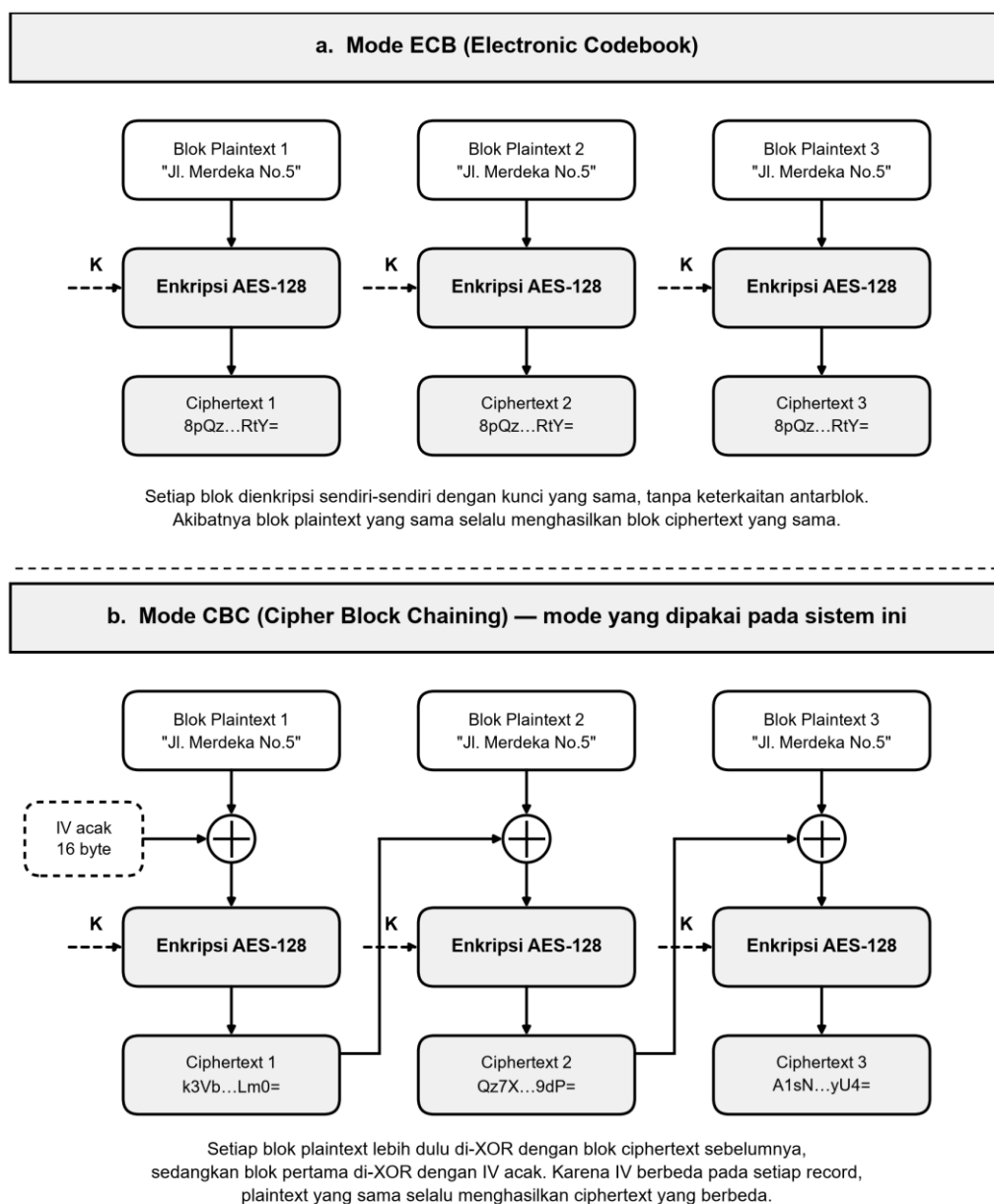
a. Alasan Memilih CBC dan Bukan ECB

Mode paling sederhana adalah Electronic Codebook (ECB), yang mengenkripsi setiap blok secara terpisah dengan kunci yang sama. Masalahnya, blok plaintext yang sama akan selalu menghasilkan blok ciphertext yang sama pula. Pada data pasien, kelemahan ini nyata sekali dampaknya.

Puskesmas Batahan melayani satu kecamatan, sehingga banyak pasien yang tinggal di desa dan jalan yang sama. Kalau memakai ECB, dua pasien yang alamatnya sama persis akan punya ciphertext yang sama persis juga. Orang yang melihat isi basis data memang tetap tidak tahu apa isi alamatnya, tetapi dia langsung tahu bahwa kedua pasien tersebut tinggal di tempat yang sama. Pola seperti ini bisa dipakai untuk mengelompokkan pasien, dan pada data rekam medis dampaknya

lebih berbahaya lagi karena pasien dengan diagnosa yang sama akan langsung terlihat berkelompok tanpa penyerang perlu memecahkan kuncinya sama sekali.

Mode CBC menutup celah ini dengan cara meng-XOR setiap blok plaintext dengan blok ciphertext sebelumnya sebelum dienkripsi. Blok pertama tidak punya blok sebelumnya, sehingga di-XOR dengan Initialization Vector yang dibangkitkan acak. Karena IV berbeda pada setiap record, maka plaintext yang sama akan selalu menghasilkan ciphertext yang berbeda. Perbandingan skema kedua mode ditampilkan pada Gambar 4.12.



Gambar 4.12 Perbandingan Skema Mode ECB dan Mode CBC

Untuk membuktikan penjelasan di atas, dilakukan pengujian kecil. Alamat yang sama persis dienkripsi dua kali dengan kunci yang sama, sekali memakai mode ECB dan sekali memakai mode CBC. Hasilnya ditampilkan pada Tabel 4.17.

Tabel 4.17 Perbandingan Ciphertext Mode ECB dan Mode CBC

Mode	Plaintext	Ciphertext (Base64)	Keterangan
ECB	Jl. Lintas	i8Lr3jR7IU5apYiGYCsY0BH	Pasien A dan pasien B menghasilkan ciphertext yang identik
	Barat No. 21,	X44BzTzLqycf5pi4QA8D1TN	
	Batahan	YjnAUx6Jid8MPAcxmH	
ECB	Jl. Lintas	i8Lr3jR7IU5apYiGYCsY0BH	(sama persis dengan baris di atas)
	Barat No. 21,	X44BzTzLqycf5pi4QA8D1TN	
	Batahan	YjnAUx6Jid8MPAcxmH	
CBC	Jl. Lintas	hzugCQJra/b5Awf7vYaMrMvk	Pasien A, memakai IV acak sendiri
	Barat No. 21,	80NT1PizcvL7UKFclJwQsc/E	
	Batahan	8xiFAemRn3rOYFE3	
CBC	Jl. Lintas	mjrIS+DUNaos3ptO1kmVwdN	Pasien B, ciphertext berbeda meski plaintext sama
	Barat No. 21,	hW/fVjflr03eGXw759kCllyiyi	
	Batahan	pEYdLitepyjF0Cg	

Hasilnya persis seperti yang diperkirakan. Pada mode ECB kedua ciphertext sama persis, sedangkan pada mode CBC keduanya sama sekali berbeda meskipun alamat, kunci, dan algoritmanya sama. Inilah alasan mode CBC dipilih untuk sistem ini.

b. Pengelolaan Initialization Vector

IV yang dipakai sistem ini panjangnya 16 byte, sama dengan ukuran satu blok AES. IV dibangkitkan secara acak menggunakan `CryptoJS.lib.WordArray.random(16)` setiap kali sebuah record disimpan, lalu diubah ke bentuk heksadesimal sepanjang 32 karakter dan disimpan pada kolom iv di tabel yang bersangkutan.

Satu pertanyaan yang wajar muncul adalah, kalau IV disimpan berdampingan dengan ciphertext di basis data yang sama, apakah itu tidak berbahaya? Jawabannya tidak, karena IV memang tidak dirancang untuk dirahasiakan. Fungsi IV bukan sebagai kunci, melainkan sebagai pembeda supaya plaintext yang sama tidak menghasilkan ciphertext yang sama. Tanpa mengetahui kunci AES-128, mengetahui IV sama sekali tidak membantu penyerang mendekripsi data. Yang justru berbahaya adalah kalau IV dipakai berulang untuk data yang berbeda, karena pola antar record bisa mulai terbaca. Karena itu sistem ini membangkitkan IV baru setiap kali data disimpan maupun diubah, tidak pernah memakai IV yang sama dua kali.

Pada implementasinya, satu IV dipakai bersama untuk seluruh atribut dalam satu record yang sama. Jadi nama, NIK, tanggal lahir, alamat, dan nomor telepon milik seorang pasien dienkripsi dengan IV yang sama, tetapi pasien lain memakai IV yang berbeda. Pilihan ini diambil supaya struktur tabel tetap sederhana, cukup satu kolom iv per record, dan karena kelima atribut tersebut isinya memang berbeda satu sama lain sehingga tidak menimbulkan ciphertext kembar.

c. Penerapan Padding PKCS#7

AES hanya bisa memproses data yang panjangnya kelipatan 16 byte. Padahal data pasien jarang sekali panjangnya pas kelipatan 16. Untuk itu dipakai padding PKCS#7, yaitu penambahan byte di akhir data sampai panjangnya menjadi kelipatan 16.

Aturannya sederhana. Kalau kurang n byte lagi untuk mencapai kelipatan 16, maka ditambahkan n buah byte yang nilainya sama dengan n . Nama Siti Aminah yang panjangnya 11 byte kurang 5 byte lagi, jadi ditambahkan 5 byte bernilai 05. Saat dekripsi, sistem cukup membaca byte terakhir untuk tahu berapa byte padding

yang harus dibuang. Yang menarik, kalau panjang data kebetulan sudah pas kelipatan 16, padding tetap ditambahkan satu blok penuh berisi 16 byte bernilai 10 heksadesimal. Ini dilakukan supaya tidak ada kebingungan saat dekripsi antara byte data asli dan byte padding. Contoh penerapannya ditampilkan pada Tabel 4.18.

Tabel 4.18 Contoh Penerapan Padding PKCS#7 pada Data Pasien

Data Asli	Panjang (byte)	Byte Padding	Panjang Setelah Padding	Jumlah Blok	Panjang Ciphertext Base64
Siti Aminah	11	5 byte bernilai 05	16 byte	1	24 karakter
081376542190	12	4 byte bernilai 04	16 byte	1	24 karakter
1990-03-17	10	6 byte bernilai 06	16 byte	1	24 karakter
Muhammad Rizky Pratama	22	10 byte bernilai 0A	32 byte	2	44 karakter
120705450301000 2	16	16 byte bernilai 10	32 byte	2	44 karakter
Jl. Lintas Barat No. 21, Batahan	32	16 byte bernilai 10	48 byte	3	64 karakter

Baris kelima pada tabel di atas adalah contoh yang menjelaskan kenapa ciphertext NIK selalu 44 karakter. NIK panjangnya tepat 16 byte, tetapi karena aturan PKCS#7 mewajibkan padding selalu ditambahkan, satu blok penuh berisi 16 byte bernilai 10 tetap disisipkan sehingga totalnya menjadi 32 byte atau dua blok.

4.4.6 Potongan Kode Implementasi

Bagian ini menampilkan potongan kode inti yang menjalankan proses enkripsi dan dekripsi pada sistem. Yang ditampilkan hanya bagian yang berhubungan langsung dengan pengamanan data, sedangkan kode lengkapnya dapat dilihat pada lampiran.

a. Fungsi `encrypt()` dan `decrypt()`

Kedua fungsi ini adalah pembungkus paling dasar yang dipakai seluruh bagian sistem. Kodenya ditampilkan pada Gambar 4.13.

```
// lib/aes.ts
export function encrypt(
  plaintext: string,
): { ciphertext: string; iv: string } {
  const iv = CryptoJS.lib.WordArray.random(16);
  const encrypted = CryptoJS.AES.encrypt(plaintext, KEY, {
    iv,
    mode: CryptoJS.mode.CBC,
    padding: CryptoJS.pad.Pkcs7,
  });
  return {
    ciphertext: encrypted.toString(),
    iv: iv.toString(CryptoJS.enc.Hex),
  };
}

export function decrypt(ciphertext: string, ivHex:
string): string {
  const iv = CryptoJS.enc.Hex.parse(ivHex);
  const decrypted = CryptoJS.AES.decrypt(ciphertext, KEY,
  {
    iv,
    mode: CryptoJS.mode.CBC,
    padding: CryptoJS.pad.Pkcs7,
  });
  return decrypted.toString(CryptoJS.enc.Utf8);
}
```

Gambar 4.13 Potongan Kode Fungsi `encrypt()` dan `decrypt()`

Pada fungsi `encrypt()`, baris pertama membangkitkan IV acak sepanjang 16 byte. Pemanggilan `CryptoJS.AES.encrypt` kemudian dijalankan dengan tiga

parameter penting: IV yang baru dibangkitkan, mode CBC, dan padding PKCS#7. Ketiganya ditulis eksplisit dan tidak dibiarkan memakai nilai bawaan, supaya tidak ada keraguan mode apa yang sebenarnya berjalan. Nilai yang dikembalikan berupa ciphertext dalam format Base64 dan IV dalam format heksadesimal, karena kedua format inilah yang disimpan di basis data. Pada fungsi decrypt(), IV yang tersimpan sebagai teks heksadesimal diubah kembali menjadi WordArray sebelum dipakai. Hasil dekripsi kemudian dibaca sebagai UTF-8 supaya kembali menjadi teks yang bisa dibaca manusia.

b. Fungsi encryptRecord() dan decryptRecord()

Karena satu record punya beberapa atribut yang harus dienkrpsi dengan IV yang sama, dibuat dua fungsi tambahan yang bekerja pada tingkat record. Kodenya ditampilkan pada Gambar 4.14.

```
// lib/aes.ts
export function encryptRecord<T extends Record<string,
string>>(<
  data: T,
  fields: (keyof T)[],
): { encrypted: T; iv: string } {
  const iv = CryptoJS.lib.WordArray.random(16);
  const ivHex = iv.toString(CryptoJS.enc.Hex);
  const encrypted = { ...data };
  for (const field of fields) {
    const value = data[field];
    if (value !== undefined && value !== null && value !==
    "") {
      const result = CryptoJS.AES.encrypt(String(value),
KEY, {
        iv,
        mode: CryptoJS.mode.CBC,
        padding: CryptoJS.pad.Pkcs7,
      });
      encrypted[field] = result.toString() as T[keyof T];
    }
  }
  return { encrypted, iv: ivHex };
}
```

```

export function decryptRecord<T extends Record<string,
string>>>(
  data: T, ivHex: string, fields: (keyof T)[],
): T {
  const decrypted = { ...data };
  for (const field of fields) {
    const value = data[field];
    if (value) {
      try {
        decrypted[field] = decrypt(String(value), ivHex)
as T[keyof T];
      } catch {
        decrypted[field] = "" as T[keyof T];
      }
    }
  }
  return decrypted;
}

```

Gambar 4.14 Potongan Kode Fungsi encryptRecord() dan decryptRecord()

Fungsi encryptRecord() menerima objek data beserta daftar nama kolom mana saja yang harus dienkripsi. Dengan cara ini, penentuan kolom yang dienkripsi tidak ditanamkan di dalam fungsi kriptografinya, melainkan ditentukan oleh pemanggilnya. Kolom yang nilainya kosong sengaja dilewati supaya sistem tidak menyimpan ciphertext dari string kosong yang tidak ada gunanya. Pada decryptRecord(), setiap proses dekripsi dibungkus try-catch. Ini penting karena kalau ada satu kolom yang gagal didekripsi, misalnya karena IV-nya tidak cocok, sistem tetap mengembalikan kolom lainnya dan tidak langsung berhenti dengan galat.

c. Pemanggilan pada Proses Penyimpanan Data Pasien

Potongan kode berikut memperlihatkan bagaimana fungsi enkripsi dipanggil ketika data pasien baru disimpan. Kodenya ditampilkan pada Gambar 4.15.

```

// app/api/pasien/route.ts
const ENCRYPTED_FIELDS =
  ["nama", "nik", "tanggalLahir", "alamat", "noTelepon"]
as const;

```

```

export async function POST(req: NextRequest) {
  const s = await requireRole(["admin", "petugas"]); //
1. periksa hak akses
  if (!s.ok) return s.response;

  const body = await req.json();
  const parsed = createPasienSchema.safeParse(body); //
2. validasi masukan
  if (!parsed.success) {
    return NextResponse.json({ error:
parsed.error.flatten() }, { status: 400 });
  }

  const { noRm, ...rest } = parsed.data;
  // 3. enkripsi seluruh atribut sensitif memakai satu IV
  const { encrypted, iv } = encryptRecord(rest,
[...ENCRYPTED_FIELDS]);

  const pasien = await prisma.pasien.create({ //
4. simpan ciphertext
    data: {
      noRm,
      nama: encrypted.nama,
      nik: encrypted.nik,
      tanggalLahir: encrypted.tanggalLahir,
      alamat: encrypted.alamat,
      noTelepon: encrypted.noTelepon,
      iv,
      createdById: parseInt(s.session.user.id),
    },
  });

  // 5. catat aktivitas ke tabel audit_log
  await logAudit(s.session.user.id, "CREATE", "pasien",
pasien.id, req);
  return NextResponse.json({ id: pasien.id, noRm:
pasien.noRm },
                                { status: 201 });
}

```

Gambar 4.15 Potongan Kode Penyimpanan Data Pasien dengan Enkripsi AES-128

Ada lima langkah yang berjalan berurutan pada kode di atas. Pertama, sistem memeriksa apakah pengguna yang mengirim permintaan punya peran admin atau petugas. Kalau tidak, permintaan langsung ditolak dengan status 403 sebelum

data apa pun diproses. Kedua, data yang masuk divalidasi. Ketiga, seluruh atribut sensitif dienkripsi sekaligus dengan satu IV. Keempat, yang dikirim ke basis data sudah berupa ciphertext, bukan data asli. Kelima, aktivitas ini dicatat ke `audit_log` lengkap dengan siapa pelakunya dan record mana yang dibuat. Yang perlu digarisbawahi adalah bahwa data asli tidak pernah menyentuh basis data sama sekali, karena enkripsi dilakukan di sisi server sebelum perintah penyimpanan dijalankan.

d. Penjaga Hak Akses Berbasis Peran

Enkripsi saja belum cukup, karena data yang sudah didekripsi tetap harus dibatasi siapa yang boleh melihatnya. Pembatasan ini ditangani oleh dua lapisan, yaitu middleware untuk halaman dan fungsi penjaga untuk endpoint API. Kodenya ditampilkan pada Gambar 4.16.

```
// lib/rbac.ts – penjaga endpoint API
export type Role = "admin" | "dokter" | "petugas";

export async function requireRole(roles: Role[]) {
  const r = await requireSession();
  if (!r.ok) return r; // belum login -> 401
  if (!roles.includes(r.session.user.role)) {
    return {
      ok: false as const,
      response: NextResponse.json({ error: "Forbidden" },
    { status: 403 })),
    };
  }
  return r;
}

// middleware.ts – penjaga halaman
export default auth((req) => {
  const { pathname } = req.nextUrl;
  const isLoggedIn = !!req.auth;
  const role = req.auth?.user?.role;

  if (!isLoggedIn) {
    return NextResponse.redirect(new URL("/login",
req.url));
  }
});
```

```

    }
    if (pathname.startsWith("/pengguna") && role !==
"admin") {
        return NextResponse.redirect(new URL("/dashboard",
req.url));
    }
    if (pathname.startsWith("/audit-log") && role !==
"admin") {
        return NextResponse.redirect(new URL("/dashboard",
req.url));
    }
    return NextResponse.next();
});

```

Gambar 4.16 Potongan Kode Kontrol Akses Berbasis Peran

Dua lapisan ini sengaja dibuat terpisah karena tugasnya memang berbeda. Middleware mencegah pengguna membuka halaman yang bukan haknya melalui peramban, sedangkan `requireRole` mencegah pengguna memanggil endpoint API secara langsung, misalnya lewat Postman atau perintah curl. Kalau hanya mengandalkan middleware saja, pengguna yang tahu alamat endpoint-nya masih bisa mengambil data tanpa melewati halaman. Karena itu setiap endpoint tetap memeriksa peran sendiri. Pemeriksaan ini juga selalu diletakkan di baris paling awal fungsi, sebelum badan permintaan dibaca, supaya permintaan yang tidak berhak ditolak secepat mungkin.

4.5 Implementasi Antarmuka Sistem

Sub-bab ini menampilkan hasil implementasi antarmuka sistem beserta penjelasan fungsi setiap halaman dan siapa saja yang berhak mengaksesnya. Seluruh data yang terlihat pada tangkapan layar merupakan data uji dengan nama, NIK, dan alamat fiktif. Data pasien yang sebenarnya sengaja tidak ditampilkan, karena akan bertentangan dengan tujuan penelitian ini sendiri.

Sistem memiliki tujuh halaman utama dengan pembagian hak akses seperti yang ditampilkan pada Tabel 4.19.

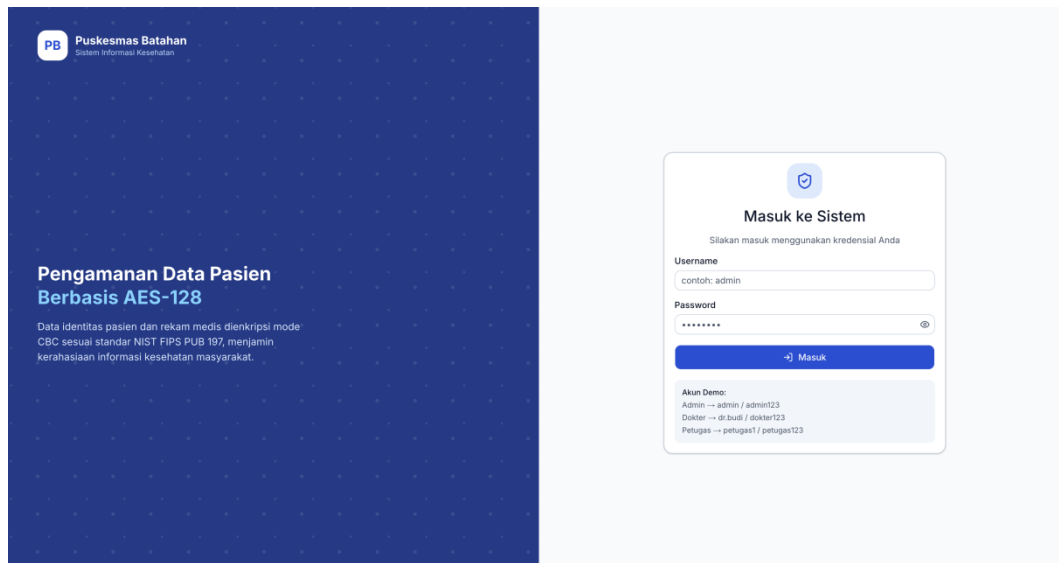
Tabel 4.19 Daftar Halaman Sistem dan Hak Aksesnya

No	Halaman	Alamat	Admin	Dokter	Petugas
1	Login	/login	✓	✓	✓
2	Dashboard	/dashboard	✓	✓	✓
3	Data Pasien	/pasien	✓	✓	✓
4	Rekam Medis	/rekam-medis	✓	✓	—
5	Manajemen Pengguna	/pengguna	✓	—	—
6	Log Aktivitas	/audit-log	✓	—	—
7	Demo Enkripsi	/demo-enkripsi	✓	✓	✓

Menu yang tampil di bilah samping menyesuaikan peran pengguna yang sedang login. Petugas pendaftaran tidak akan melihat menu Rekam Medis, Manajemen Pengguna, maupun Log Aktivitas. Penyembunyian menu ini hanya untuk kenyamanan pengguna dan bukan merupakan pengamanan utama, karena pengamanan yang sebenarnya dilakukan di sisi server melalui middleware dan pemeriksaan peran pada setiap endpoint.

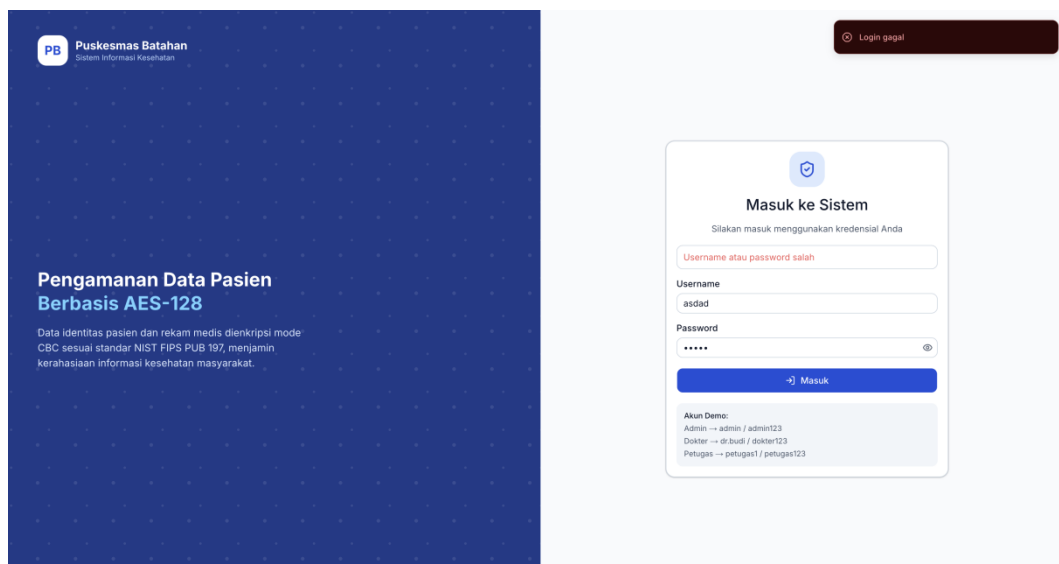
4.5.1 Halaman Login

Halaman login merupakan pintu masuk sistem. Pengguna memasukkan username dan kata sandi, lalu sistem mencari akun tersebut di tabel users dan membandingkan kata sandi yang diketik dengan hash bcrypt yang tersimpan menggunakan fungsi `bcrypt.compare()`. Kalau cocok, sistem membuat token sesi berbasis JWT yang berisi id pengguna dan perannya, lalu mengarahkan pengguna ke dashboard. Tampilan halaman login ditampilkan pada Gambar 4.17.



Gambar 4.17 Tampilan Halaman Login

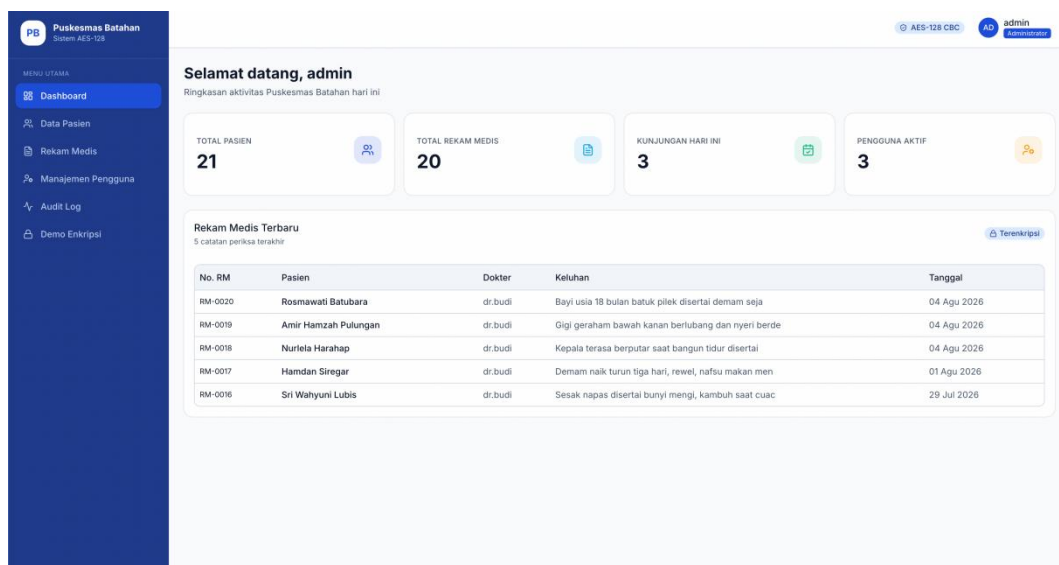
Ketika kata sandi yang dimasukkan salah, 89system menampilkan pesan kesalahan yang sifatnya umum, yaitu bahwa username atau kata sandi tidak sesuai. Pesan sengaja dibuat tidak spesifik dan tidak menyebutkan apakah yang salah itu username-nya atau kata sandinya. Kalau pesannya spesifik, penyerang bisa memakainya untuk menebak username mana saja yang terdaftar di 89system. Tampilan penanganan kesalahan login ditampilkan pada Gambar 4.18.



Gambar 4.18 Penanganan Kesalahan pada Proses Login

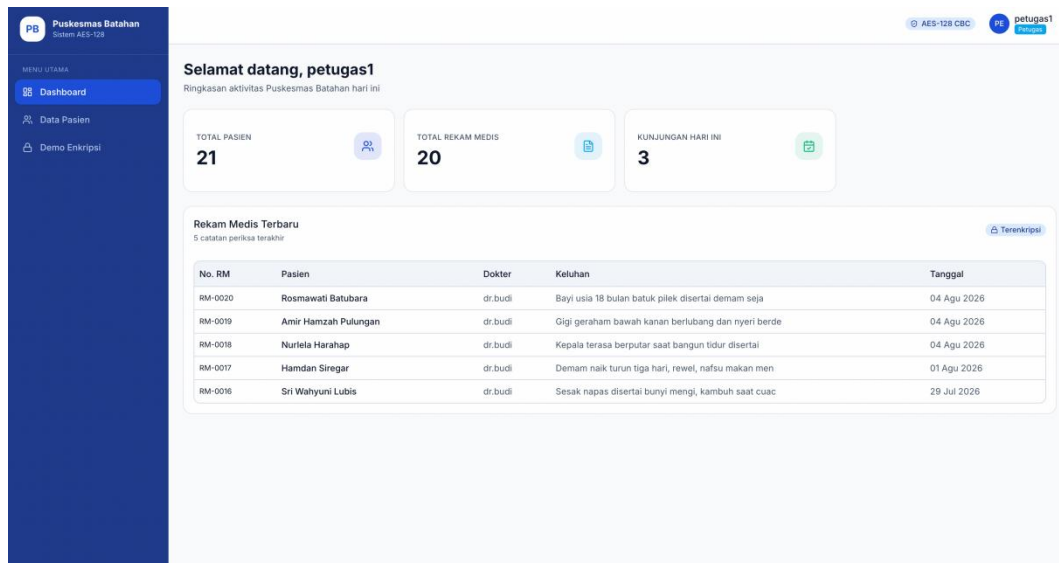
4.5.2 Halaman Dashboard

Dashboard menampilkan ringkasan data berupa jumlah pasien terdaftar, jumlah rekam medis, jumlah kunjungan hari ini, dan jumlah pengguna 90system, ditambah daftar lima pemeriksaan terakhir. Data pada daftar pemeriksaan terakhir tersebut merupakan hasil dekripsi, baik nama pasiennya maupun keluhanannya, sehingga bagian ini sekaligus membuktikan bahwa proses dekripsi berjalan otomatis saat data ditampilkan. Tampilan dashboard untuk peran admin ditampilkan pada Gambar 4.19.



Gambar 4.19 Tampilan Dashboard Peran Admin

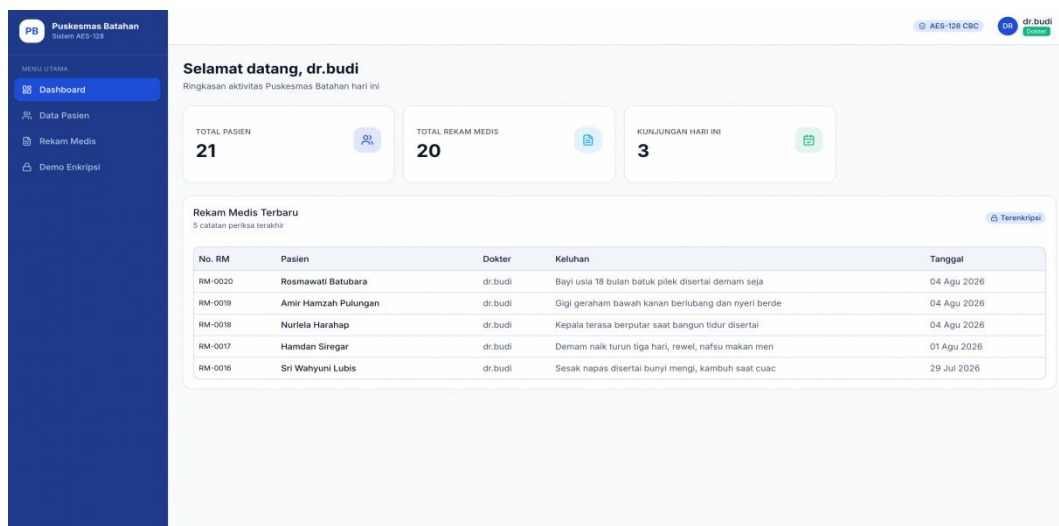
Ketika login menggunakan akun dengan peran petugas, menu yang tersedia di bilah samping berkurang. Perbedaan tampilan inilah bukti pertama bahwa control akses berbasis peran bekerja. Tampilannya ditampilkan pada Gambar 4.20.



Gambar 4.20 Tampilan Dashboard Peran Petugas

4.5.3 Halaman Manajemen Pengguna

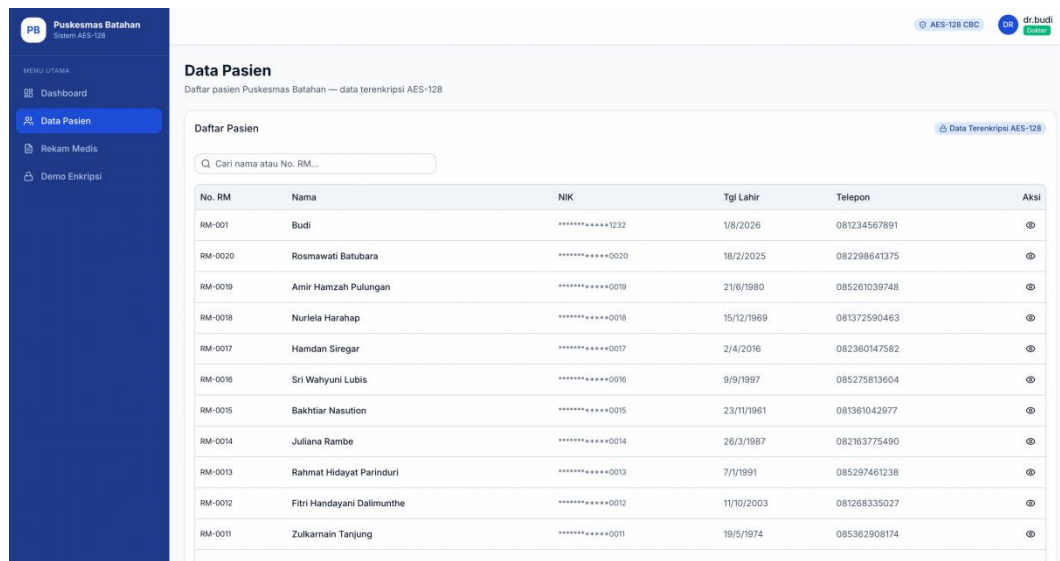
Halaman ini hanya bisa diakses admin dan dipakai untuk menambah akun baru, mengubah peran akun yang sudah ada, dan menghapus akun. Ketika akun baru dibuat, kata sandi yang diketik admin langsung di-hash menggunakan bcrypt dengan cost factor 12 sebelum disimpan. Admin sendiri tidak pernah bisa melihat kembali kata sandi pengguna lain, karena yang tersimpan memang bukan kata sandinya melainkan hash-nya. Tampilan halaman manajemen pengguna ditampilkan pada Gambar 4.21.



Gambar 4.21 Tampilan Halaman Manajemen Pengguna

4.5.4 Halaman Data Pasien

Halaman data pasien menampilkan daftar seluruh pasien yang terdaftar. Perlu dicatat bahwa data yang terlihat pada tabel ini sudah melewati proses dekripsi di sisi server. Yang tersimpan di basis data tetap berupa ciphertext seperti yang sudah ditunjukkan pada Gambar 4.5. Tampilan daftar pasien ditampilkan pada Gambar 4.22.



Data Pasien
Daftar pasien Puskesmas Batahan — data terenkripsi AES-128

Daftar Pasien Data Terenkripsi AES-128

Q Cari nama atau No. RM...

No. RM	Nama	NIK	Tgl Lahir	Telepon	Aksi
RM-001	Budi	*****1232	1/8/2026	081234567891	ⓘ
RM-0020	Rosmawati Batubara	*****0020	18/2/2025	082298641375	ⓘ
RM-0019	Amir Hamzah Pulungan	*****0019	21/6/1980	085261039748	ⓘ
RM-0018	Nuriela Harahap	*****0018	15/12/1969	081372590463	ⓘ
RM-0017	Hamdan Siregar	*****0017	2/4/2016	082360147582	ⓘ
RM-0016	Sri Wahyuni Lubis	*****0016	9/9/1997	085275813604	ⓘ
RM-0015	Bakhtiar Nasution	*****0015	23/11/1961	081381042977	ⓘ
RM-0014	Juliana Rambe	*****0014	26/3/1987	082163775480	ⓘ
RM-0013	Rahmat Hidayat Parinduri	*****0013	7/1/1991	085297461238	ⓘ
RM-0012	Fitri Handayani Dalimunthe	*****0012	11/10/2003	081268335027	ⓘ
RM-0011	Zulkarnain Tanjung	*****0011	19/5/1974	085362908174	ⓘ

Gambar 4.22 Tampilan Halaman Daftar Data Pasien

Formulir pendaftaran pasien baru memuat kolom nomor rekam medis, nama, NIK, tanggal lahir, alamat, dan nomor telepon. Seluruh kolom kecuali nomor rekam medis akan dienkripsi sebelum disimpan. Validasi dilakukan di dua tempat, yaitu di sisi peramban supaya pengguna langsung mendapat umpan balik, dan di sisi server memakai Zod supaya data yang tidak valid tetap tertolak walaupun dikirim langsung ke endpoint tanpa melewati 92ormular. Tampilan formulirnya ditampilkan pada Gambar 4.23.

Gambar 4.23 Tampilan Formulir Pendaftaran Pasien Baru

Halaman detail pasien menampilkan seluruh data identitas pasien beserta riwayat rekam medisnya. Tampilannya ditampilkan pada Gambar 4.24.

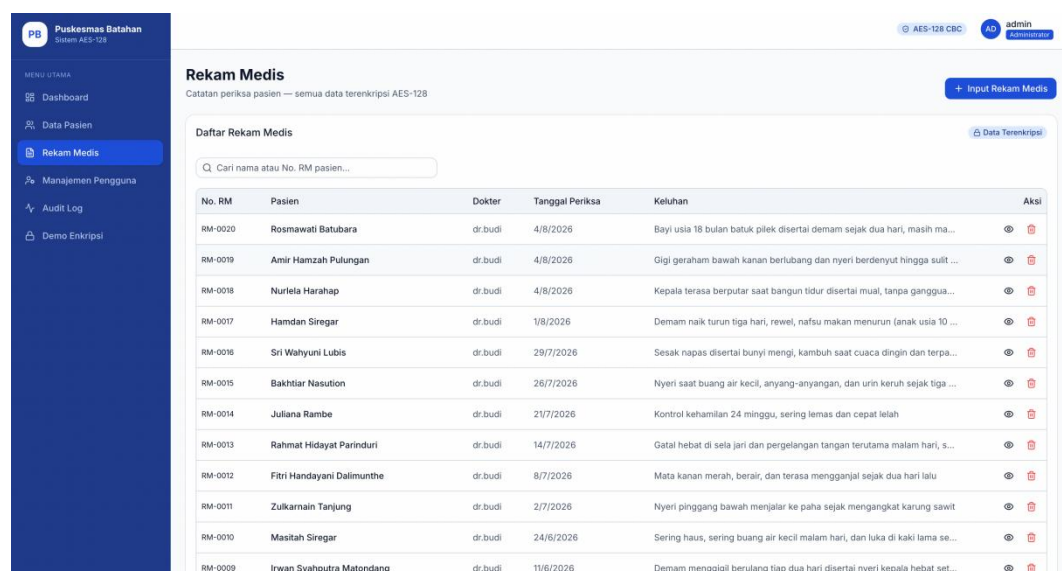
Gambar 4.24 Tampilan Halaman Detail Data Pasien

Ada satu hal teknis yang perlu dijelaskan mengenai fitur pencarian pada halaman ini. Karena kolom nama tersimpan dalam bentuk ciphertext, pencarian tidak bisa dilakukan dengan perintah LIKE di SQL. Ciphertext dari kata Siti tidak akan pernah cocok dengan potongan ciphertext dari nama Siti Aminah, sebab enkripsi mengacak seluruh blok. Karena itu pencarian dilakukan setelah data

didekripsi di sisi server, baru kemudian hasilnya disaring. Konsekuensinya, seluruh record harus diambil dan didekripsi terlebih dahulu sebelum disaring. Untuk jumlah pasien di tingkat puskesmas 94indakan masih cukup cepat, seperti yang ditunjukkan hasil pengujian performa pada sub-bab 4.6.4, tetapi keterbatasan ini tetap dicatat dan dibahas pada sub-bab 4.7.4.

4.5.5 Halaman Rekam Medis

Halaman rekam medis hanya bisa diakses admin dan dokter. Petugas pendaftaran sengaja tidak diberi akses karena diagnosa dan tindakan medis bukan bagian dari tugasnya. Pembatasan ini merupakan penerapan prinsip hak akses seperlunya, yaitu pengguna hanya diberi akses sebatas yang dibutuhkan untuk menjalankan tugasnya. Tampilan daftar rekam medis ditampilkan pada Gambar 4.25.



No. RM	Pasien	Dokter	Tanggal Periksa	Keluhan	Aksi
RM-0020	Rosmawati Batubara	dr.budi	4/8/2026	Bayi usia 18 bulan batuk pilek disertai demam sejak dua hari, masih ma...	
RM-0019	Amir Hamzah Pulungan	dr.budi	4/8/2026	Gigi geraham bawah kanan berlubang dan nyeri berdenyut hingga sulit ...	
RM-0018	Nuriela Harahap	dr.budi	4/8/2026	Kepala terasa berputar saat bangun tidur disertai mual, tanpa ganggua...	
RM-0017	Hamdan Siregar	dr.budi	1/8/2026	Demam naik turun tiga hari, rewel, nafsu makan menurun (anak usia 10 ...	
RM-0016	Sri Wahyuni Lubis	dr.budi	29/7/2026	Sesak napas disertai bunyi mengi, kambuh saat cuaca dingin dan terpa...	
RM-0015	Bakhtiar Nasution	dr.budi	26/7/2026	Nyeri saat buang air kecil, anyang-anyangan, dan urin keruh sejak tiga ...	
RM-0014	Juliana Rambe	dr.budi	21/7/2026	Kontrol kehamilan 24 minggu, sering lemas dan cepat lelah	
RM-0013	Rahmat Hidayat Parinduri	dr.budi	14/7/2026	Gatal hebat di sela jari dan pergelangan tangan terutama malam hari, s...	
RM-0012	Fitri Handayani Dalimunthe	dr.budi	8/7/2026	Mata kanan merah, berair, dan terasa mengganjal sejak dua hari lalu	
RM-0011	Zulkarnain Tanjung	dr.budi	2/7/2026	Nyeri pinggang bawah menjalar ke paha sejak mengangkat karung sawit	
RM-0010	Masitah Siregar	dr.budi	24/6/2026	Sering haus, sering buang air kecil malam hari, dan luka di kaki lama se...	
RM-0009	Irwan Syahputra Matondang	dr.budi	11/6/2026	Demam menggigil berulang tiap dua hari disertai nyeri kepala hebat set...	

Gambar 4.25 Tampilan Halaman Daftar Rekam Medis

Pada formulir input rekam medis, dokter memilih pasien lalu mengisi keluhan, diagnosa, tindakan, dan resep. Keempat kolom tersebut dienkripsi dengan satu IV yang sama sebelum disimpan. Nama dokter tidak perlu diisi manual karena diambil otomatis dari sesi pengguna yang sedang login, sehingga tidak bisa dipalsukan dari sisi peramban. Tampilan formulirnya ditampilkan pada Gambar 4.26.

Gambar 4.26 Tampilan Formulir Input Rekam Medis

4.5.6 Halaman Log Aktivitas

Halaman log aktivitas hanya bisa dibuka admin dan menampilkan 200 catatan terakhir dari tabel `audit_log`. Setiap baris memuat nama pengguna, jenis aksi, tabel yang diakses, id record yang disentuh, alamat IP, dan waktu kejadian. Halaman ini menjawab kelemahan nomor tiga pada Tabel 4.1, yaitu tidak adanya jejak siapa yang mengakses data pasien. Tampilannya ditampilkan pada Gambar 4.27.

Waktu	Pengguna	Aksi	Target	ID Target	IP
4/8/2026, 21:30.43	admin	READ	users	-	192.168.1.22
4/8/2026, 21:30.07	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:29.33	admin	READ	rekam_medis	-	192.168.1.22
4/8/2026, 21:27.11	admin	READ	pasien	1	192.168.1.22
4/8/2026, 21:27.08	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:26.08	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:24.26	dr.budi	READ	pasien	-	192.168.1.22
4/8/2026, 21:19.12	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:16.15	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:04.15	admin	READ	pasien	-	192.168.1.22
4/8/2026, 21:04.15	admin	CREATE	pasien	1	192.168.1.22
4/8/2026, 21:03.37	admin	READ	pasien	-	192.168.1.22
4/8/2026, 19:14.38	dr.budi	READ	pasien	22	192.168.1.32
4/8/2026, 18:14.38	dr.budi	CREATE	rekam_medis	21	192.168.1.32
4/8/2026, 14:14.38	petugas1	CREATE	pasien	22	192.168.1.22
4/8/2026, 13:14.38	petugas1	LOGIN	users	-	192.168.1.22

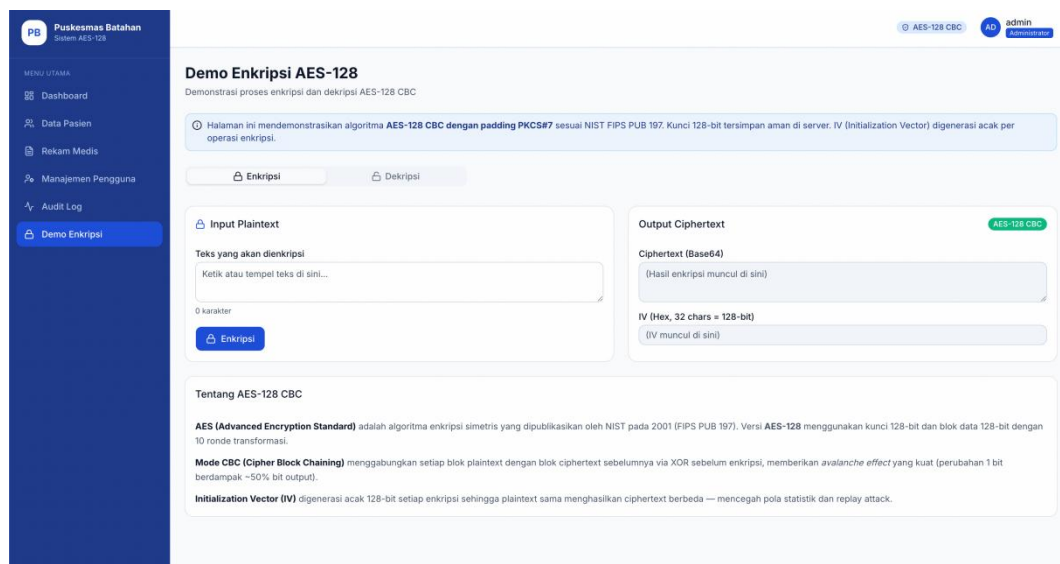
Gambar 4.27 Tampilan Halaman Log Aktivitas

Pencatatan log dilakukan otomatis oleh fungsi `logAudit()` yang dipanggil di setiap endpoint setelah operasi berhasil dijalankan. Fungsi ini dibungkus try-catch, sehingga kalau penulisan log gagal karena satu dan lain hal, operasi utamanya tetap berhasil dan pengguna tidak terganggu. Alamat IP diambil dari header `x-forwarded-for` atau `x-real-ip` supaya tetap terbaca dengan benar walaupun aplikasi nanti dijalankan di belakang reverse proxy.

4.5.7 Halaman Demo Enkripsi

Halaman ini bukan bagian dari alur kerja pelayanan pasien, melainkan sarana untuk memperagakan cara kerja AES-128 secara langsung. Pengguna bisa mengetik teks apa saja, lalu sistem menampilkan ciphertext dalam format Base64 beserta IV dalam format heksadesimal sepanjang 32 karakter. Ciphertext dan IV tersebut kemudian bisa dikembalikan lagi menjadi teks aslinya melalui tab dekripsi.

Halaman ini berguna untuk dua hal. Pertama, untuk memperlihatkan kepada pihak puskesmas bahwa data mereka memang berubah bentuk, bukan sekadar disembunyikan di tampilan. Kedua, untuk memperagakan bahwa mengenkripsi teks yang sama dua kali akan menghasilkan ciphertext yang berbeda, karena IV-nya dibangkitkan ulang setiap kali. Tampilannya ditampilkan pada Gambar 4.28.



Gambar 4.28 Tampilan Halaman Demo Enkripsi AES-128

4.6 Pengujian Sistem

Pengujian dilakukan dengan empat metode sesuai yang direncanakan pada sub-bab 3.10, ditambah satu kelompok pengujian keamanan. Keempat metode tersebut adalah pengujian black box untuk memastikan fungsionalitas berjalan, pengujian validitas algoritma terhadap vektor uji NIST FIPS 197, pengujian avalanche effect untuk mengukur kualitas difusi, dan pengujian performa untuk mengukur dampak enkripsi terhadap kecepatan sistem.

4.6.1 Pengujian Black Box

Pengujian black box dilakukan dengan menjalankan 38 skenario yang mencakup seluruh kebutuhan fungsional pada Tabel 4.2. Setiap skenario diberi kode kebutuhan supaya bisa ditelusuri kembali ke kebutuhan mana yang sedang diuji. Skenario dibagi ke dalam lima kelompok, yaitu autentikasi dan hak akses, manajemen pengguna, data pasien, rekam medis, serta log aktivitas dan fitur pendukung.

Yang perlu ditekankan, skenario yang diuji tidak hanya skenario berhasil. Justru skenario gagal seperti login dengan kata sandi salah, petugas membuka halaman admin, atau NIK yang kurang dari 16 digit itulah yang membuktikan bahwa validasi dan kontrol akses benar-benar bekerja. Kalau hanya skenario berhasil yang diuji, sistem yang sama sekali tidak punya pengamanan pun akan lolos semua pengujian. Hasil lengkapnya ditampilkan pada Tabel 4.20.

Tabel 4.20 Hasil Pengujian Black Box

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
1	KF-01	Login dengan kredensial yang benar	admin / admin123	Berhasil masuk dan diarahkan ke dashboard	Sesuai	Valid
2	KF-01	Login dengan kata sandi salah	admin / salah123	Muncul pesan kesalahan, tetap di halaman login	Sesuai	Valid

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
3	KF-01	Login dengan username tidak terdaftar	userpalsu / 123456	Muncul pesan kesalahan yang sama, akun tidak terungkap	Sesuai	Valid
4	KF-01	Login dengan kolom kosong	(kosong) / (kosong)	Validasi formulir menahan, permintaan tidak dikirim	Sesuai	Valid
5	KF-02	Logout dari sistem	Klik menu keluar	Sesi dihapus dan pengguna diarahkan ke /login	Sesuai	Valid
6	KF-03	Membuka /dashboard tanpa login	Akses URL langsung	Dialihkan otomatis ke halaman /login	Sesuai	Valid
7	KF-03	Petugas membuka halaman /pengguna	Login petugas1, akses /pengguna	Dialihkan kembali ke /dashboard	Sesuai	Valid
8	KF-03	Dokter membuka halaman /audit-log	Login dr.budi, akses /audit-log	Dialihkan kembali ke /dashboard	Sesuai	Valid
9	KF-03	Petugas memanggil GET /api/pengguna	Postman + sesi petugas	Ditolak dengan HTTP 403 Forbidden	Sesuai	Valid

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
1 0	KF- 03	Memanggil GET /api/pasien tanpa sesi	Postman tanpa cookie sesi	Ditolak dengan HTTP 401 Unauthorized	Sesua i	Vali d
1 1	KF- 04	Admin menambah pengguna baru	perawat1 / 123456 / petugas	HTTP 201, kata sandi tersimpan sebagai hash bcrypt	Sesua i	Vali d
1 2	KF- 04	Menambah pengguna dengan username yang sudah ada	admin	Ditolak dengan HTTP 409 dan pesan username sudah ada	Sesua i	Vali d
1 3	KF- 04	Menambah pengguna dengan kata sandi kurang dari 6 karakter	123	Ditolak dengan HTTP 400 oleh validasi Zod	Sesua i	Vali d
1 4	KF- 04	Admin mengubah peran pengguna	petugas1 menjadi dokter	Peran berubah dan menu ikut menyesuaikan	Sesua i	Vali d
1 5	KF- 04	Admin menghapus pengguna	Pilih akun perawat1	Akun terhapus dari daftar pengguna	Sesua i	Vali d

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
1 6	KF- 05	Menambah pasien dengan data lengkap	RM-0011, Siti Aminah, NIK 16 digit	HTTP 201, data tersimpan	Sesua i	Vali d
1 7	KF- 05	Memeriksa hasil penyimpanan di basis data	SELECT * FROM pasien	Kolom nama, nik, alamat berisi ciphertext Base64	Sesua i	Vali d
1 8	KF- 05	Menambah pasien dengan NIK kurang dari 16 digit	NIK 12070545	Ditolak dengan pesan NIK harus 16 digit	Sesua i	Vali d
1 9	KF- 05	Menambah pasien dengan nomor RM yang sudah ada	RM-0001	Ditolak dengan HTTP 409	Sesua i	Vali d
2 0	KF- 05	Menambah pasien dengan kolom nama kosong	nama = (kosong)	Ditolak dengan HTTP 400	Sesua i	Vali d
2 1	KF- 06	Menampilkan daftar pasien	Buka halaman /pasien	Data tampil terbaca hasil dekripsi otomatis	Sesua i	Vali d

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
2 2	KF- 07	Membuka detail seorang pasien	Klik salah satu baris	Seluruh atribut tampil sesuai data asli	Sesuai	Valid
2 3	KF- 08	Mengubah alamat pasien	Alamat diubah lalu disimpan	Data tersimpan ulang dengan IV yang baru	Sesuai	Valid
2 4	KF- 09	Admin menghapus data pasien	Pilih hapus pada satu pasien	Data pasien dan rekam medisnya ikut terhapus	Sesuai	Valid
2 5	KF- 09	Petugas mencoba menghapus data pasien	DELETE /api/pasien/ 1	Ditolak dengan HTTP 403	Sesuai	Valid
2 6	KF- 10	Mencari pasien berdasarkan nama	Kata kunci: Siti	Hanya pasien yang namanya cocok yang tampil	Sesuai	Valid
2 7	KF- 10	Mencari dengan kata kunci yang tidak ada	Kata kunci: zzz	Daftar kosong tanpa galat	Sesuai	Valid
2 8	KF- 11	Dokter menyimpan rekam medis baru	Keluhan, diagnosa, tindakan, resep	HTTP 201, keempat kolom tersimpan terenkripsi	Sesuai	Valid

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
29	KF-11	Petugas mencoba menyimpan rekam medis	POST /api/rekam-medis	Ditolak dengan HTTP 403	Sesuai	Valid
30	KF-11	Menyimpan rekam medis untuk pasien yang tidak ada	pasienId = 999	Ditolak dengan HTTP 404	Sesuai	Valid
31	KF-12	Menampilkan daftar rekam medis	Buka halaman /rekam-medis	Data terbaca, nama pasien ikut terdekripsi	Sesuai	Valid
32	KF-13	Dokter mengubah diagnosa	Diagnosa diubah lalu disimpan	Data tersimpan ulang dengan IV yang baru	Sesuai	Valid
33	KF-13	Dokter mencoba menghapus rekam medis	DELETE /api/rekam-medis/1	Ditolak dengan HTTP 403, hanya admin yang boleh	Sesuai	Valid
34	KF-14	Pencatatan aktivitas otomatis	Tambah satu data pasien	Muncul baris baru di audit_log dengan aksi CREATE	Sesuai	Valid
35	KF-15	Admin membuka log aktivitas	Buka halaman /audit-log	200 catatan terakhir tampil beserta IP dan waktu	Sesuai	Valid

N o	Kod e	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Uji	Ket.
3 6	KF- 16	Demo enkripsi sebuah teks	Teks: Siti Aminah	Muncul ciphertext Base64 dan IV heksadesimal 32 karakter	Sesua i	Vali d
3 7	KF- 16	Demo dekripsi dengan IV yang salah	Satu karakter IV diubah	Hasil kosong, sistem tidak berhenti dengan galat	Sesua i	Vali d
3 8	KF- 17	Ringkasan data pada dashboard	Login lalu buka /dashboard	Jumlah pasien, rekam medis, dan kunjungan tampil benar	Sesua i	Vali d

Dari 38 skenario yang diuji, seluruhnya menghasilkan keluaran yang sesuai dengan yang diharapkan, sehingga tingkat keberhasilan pengujian black box mencapai 100%. Setiap kebutuhan fungsional KF-01 sampai KF-17 sudah terwakili oleh minimal satu skenario, jadi tidak ada kebutuhan yang lolos tanpa diuji.

Yang paling penting dari tabel di atas adalah sepuluh skenario negatifnya, yaitu nomor 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 18, 19, 20, 25, 29, 30, 33, dan 37. Skenario nomor 9 dan 10 khususnya membuktikan bahwa kontrol akses tidak hanya bekerja di tampilan. Endpoint API dipanggil langsung melalui Postman tanpa melewati antarmuka sama sekali, dan sistem tetap menolaknya.

4.6.2 Pengujian Validitas Algoritma terhadap NIST FIPS 197

Pengujian ini bertujuan memastikan bahwa implementasi AES-128 pada sistem memang menghasilkan keluaran yang sesuai standar, bukan sekadar menghasilkan teks acak yang kebetulan bisa didekripsi kembali. Caranya dengan

menguji implementasi terhadap vektor uji resmi yang dipublikasikan NIST pada dokumen FIPS PUB 197 Appendix C.1 untuk AES-128.

Vektor uji ini menggunakan mode ECB satu blok tanpa padding, karena yang diuji adalah fungsi blok cipher AES itu sendiri, bukan mode operasinya. Hasil pengujian ditampilkan pada Tabel 4.21.

Tabel 4.21 Hasil Pengujian terhadap Vektor Uji NIST FIPS 197

Parameter	Nilai
Kunci (128 bit)	000102030405060708090a0b0c0d0e0f
Plaintext (128 bit)	00112233445566778899aabbccddeeff
Ciphertext menurut FIPS 197	69c4e0d86a7b0430d8cdb78070b4c55a
Ciphertext hasil sistem	69c4e0d86a7b0430d8cdb78070b4c55a
Kesimpulan	COCOK — implementasi sesuai standar
Hasil dekripsi ciphertext	00112233445566778899aabbccddeeff
Kesimpulan dekripsi	COCOK — kembali ke plaintext semula

Ada satu hal teknis yang perlu ditulis apa adanya di sini, karena kalau dilewatkan justru akan menghasilkan kesimpulan yang keliru. Pustaka `crypto-js` secara bawaan memakai mode CBC dengan padding PKCS#7, dan yang lebih penting, kalau kunci diberikan dalam bentuk string biasa maka `crypto-js` tidak memakai string tersebut sebagai kunci melainkan memperlakukannya sebagai passphrase. Dari passphrase itu `crypto-js` menurunkan kunci sendiri dengan bantuan salt acak.

Akibatnya, kalau kunci vektor uji NIST dikirim sebagai string biasa, ciphertext yang dihasilkan adalah `f9c9708f3e01cfbad79d22f393d84f6d...` yang sama sekali tidak cocok dengan nilai yang diharapkan, dan hasilnya akan berbeda lagi setiap kali dijalankan karena salt-nya acak. Untuk pengujian ini kunci wajib diuraikan sebagai heksadesimal memakai `CryptoJS.enc.Hex.parse` dan mode operasinya harus ditetapkan secara eksplisit sebagai ECB tanpa padding. Potongan kode pengujiannya sebagai berikut.

```

const nistKey =
CryptoJS.enc.Hex.parse("000102030405060708090a0b0c0d0e0f")
;
const nistPt =
CryptoJS.enc.Hex.parse("00112233445566778899aabbccddeeff")
;

const hasil = CryptoJS.AES.encrypt(nistPt, nistKey, {
  mode: CryptoJS.mode.ECB,
  padding: CryptoJS.pad.NoPadding,
}).ciphertext.toString(CryptoJS.enc.Hex);

// hasil -> 69c4e0d86a7b0430d8cdb78070b4c55a (cocok
dengan FIPS 197)

```

Gambar 4.29 Potongan Kode Pengujian Vektor Uji NIST FIPS 197

Pada sistem yang sebenarnya, kunci juga tidak pernah dikirim sebagai string. Kunci 16 karakter dari berkas .env selalu diuraikan lebih dulu memakai CryptoJS.enc.Utf8.parse seperti yang sudah ditunjukkan pada Gambar 4.4, sehingga yang dipakai memang benar-benar kunci 128 bit tersebut.

Setelah validitas algoritma terbukti, pengujian dilanjutkan dengan uji bolak-balik pada sepuluh data uji yang mewakili seluruh jenis atribut yang dienkripsi sistem. Setiap data dienkripsi lalu langsung didekripsi kembali, dan hasilnya dibandingkan dengan data aslinya. Hasilnya ditampilkan pada Tabel 4.22.

Tabel 4.22 Hasil Pengujian Bolak-balik Enkripsi dan Dekripsi

N o	Atribut	Plaintext	Pan jang	Ciphertext (Base64)	Pan jang	Hasi l
1	nik	1207054503010002	16	5xztgbffmdACnZ6Oyb Bd469gD1uB7DrIt03 AzN5XLQ4=	44	Coco k
2	nama	Siti Aminah	11	9ADk+O2COx5EUoy UwYVaCA==	24	Coco k
3	nama	Muhammad Rizky Pratama	22	y3FtMTJDhX7Grsmo/ 8EITUeSDSf8TKcJdH OOPYUIvNg=	44	Coco k

N o	Atribut	Plaintext	Pan jang	Ciphertext (Base64)	Pan jang	Hasi l
4	alamat	Jl. Lintas Barat No. 21, Batahan	32	AVrVYxn3ZU+luXhS 2mYaRunCc1Op72kE xZUQS308X9GC...	64	Coco k
5	no_telep on	081376542190	12	/snpUr1KYFt6rukYkA 4mKw==	24	Coco k
6	tanggal_ lahir	1990-03-17	10	8/L16reuZwyjGm4Y+ D49Dg==	24	Coco k
7	keluhan	Demam tinggi sejak 3 hari, disertai batuk	41	/4fuCIZ/eSbOC+hMQ nXqUDqtfR9rATfk/jm WL8PPRHv5...	64	Coco k
8	diagnosa	ISPA (Infeksi Saluran Pernapasan Akut)	38	TFbnTgn9YR/rmkkr7t Y9J7frRcIutPI/5Bap7 U14FKWQ...	64	Coco k
9	tindakan	Pemberian obat penurun panas dan antibiotik	43	fmdD6HFyE/Tm31xhp a9NPYHPZnAQjsjRAj z13cyI/nZF...	64	Coco k
10	resep	Paracetamol 500mg 3x1, Amoxicillin 500mg 3x1	44	ZE3UBadISqQJhG+P Nv0IzpCrOGJEqBT7C gly5lNGVtX1...	64	Coco k

Seluruh sepuluh data uji berhasil dikembalikan ke bentuk aslinya secara utuh tanpa ada satu karakter pun yang berubah atau hilang, termasuk data yang mengandung spasi, tanda titik, tanda hubung, dan tanda kurung. Ini membuktikan bahwa kombinasi AES-128 mode CBC dengan padding PKCS#7 dan pengodean Base64 yang dipakai sistem berjalan konsisten, sehingga kebutuhan KNF-04 terpenuhi.

Dari kolom panjang pada tabel di atas juga terlihat pola perubahan ukuran data yang sudah dijelaskan pada Tabel 4.18. Data sepanjang 1 sampai 15 karakter menghasilkan ciphertext 24 karakter, data 16 sampai 31 karakter menghasilkan 44 karakter, dan data 32 sampai 47 karakter menghasilkan 64 karakter. Rata-rata, ukuran penyimpanan yang dibutuhkan menjadi sekitar dua kali lipat dari data aslinya. Penambahan ini merupakan konsekuensi yang tidak bisa dihindari dari penerapan enkripsi dan sudah diantisipasi dengan pemilihan tipe data TEXT pada rancangan basis data.

4.6.3 Pengujian Avalanche Effect

Avalanche effect mengukur seberapa besar perubahan pada ciphertext ketika masukan diubah sedikit saja. Algoritma kriptografi yang baik harus punya sifat ini, karena kalau perubahan satu bit pada plaintext hanya mengubah sedikit bit pada ciphertext, penyerang bisa mempelajari hubungan antara masukan dan keluaran. Nilai avalanche effect dihitung dengan rumus:

$$AE = (\text{jumlah bit yang berbeda} \div \text{total bit}) \times 100\%$$

Total bit yang dipakai adalah 128, karena satu blok AES panjangnya 128 bit. Nilai ideal avalanche effect adalah 50%, artinya perubahan satu bit pada masukan mengubah kira-kira separuh bit pada keluaran. Nilai dalam rentang 45% sampai 55% masih dianggap memenuhi sifat difusi yang baik.

Pengujian dilakukan dalam dua skenario. Pada skenario A, satu bit plaintext diubah sedangkan kuncinya tetap. Pada skenario B, satu bit kunci yang diubah sedangkan plaintext-nya tetap. Masing-masing skenario diuji sepuluh kali dengan posisi bit yang berbeda-beda. Hasil skenario A ditampilkan pada Tabel 4.23.

Tabel 4.23 Hasil Pengujian Avalanche Effect — Perubahan 1 Bit Plaintext

N o	Data Uji	Bit ke-	Ciphertext Awal	Ciphertext Setelah Diubah	Bit Berbe da	AE (%)
1	1207054503 010002	0	09e996a844f8 edb1...	8d0864533675 36f1...	62/128	48.44

No	Data Uji	Bit ke-	Ciphertext Awal	Ciphertext Setelah Diubah	Bit Berbe da	AE (%)
2	1207054503 010013	13	5e061b376144 92cc...	54fd4eec15f0c ddc...	64/128	50.00
3	1207051207 990004	26	329196a44810 5c97...	54797262968a cb3a...	65/128	50.78
4	Siti Aminah	39	5233f3cf519c d7c6...	e625483b1664 a060...	71/128	55.47
5	Budi Santoso	52	b64a335bef7a 623b...	d6096616b505 3308...	67/128	52.34
6	Ahmad Fauzi	65	98445ab3bdb7 ff64...	7aa98044d945 95ad...	75/128	58.59
7	0813765421 901234	78	e099a9b37cdf acdb...	8944393de2de d994...	65/128	50.78
8	1990-03-17	91	61bbd1a88b59 165c...	91157314920b 7a90...	61/128	47.66
9	Demam tinggi	104	29b7ce971d57 eafe...	0d2abadedcc2 86df...	55/128	42.97
10	ISPA ringan	117	87ee354dbb5d c124...	b29f1bf65c95 bbec...	64/128	50.00

Nilai rata-rata avalanche effect pada skenario A adalah 50.70%. Nilai terendah 42,97% pada pengujian kesembilan dan tertinggi 58,59% pada pengujian keenam. Sebaran ini wajar karena setiap pengujian hanya melibatkan satu blok, jadi wajar kalau ada yang sedikit di atas atau di bawah nilai ideal.

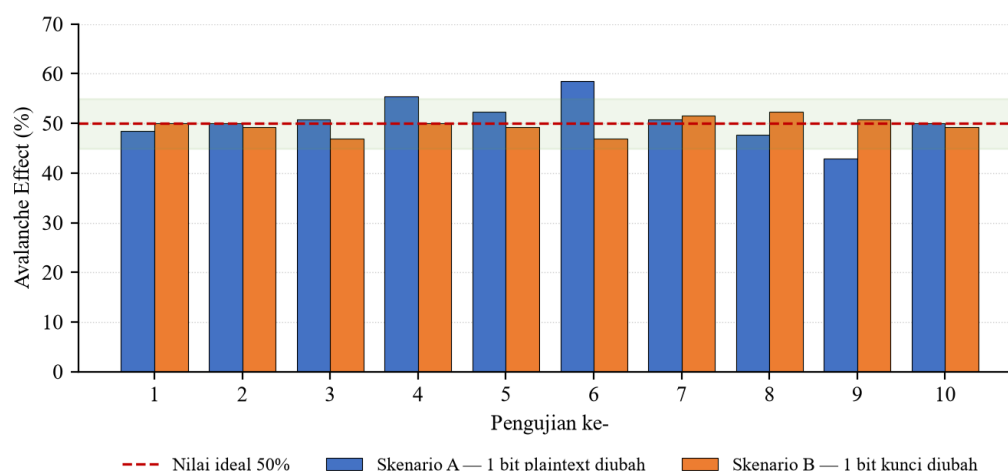
Hasil skenario B, yaitu ketika satu bit kunci yang diubah, ditampilkan pada Tabel 4.24.

Tabel 4.24 Hasil Pengujian Avalanche Effect — Perubahan 1 Bit Kunci

No	Bit Kunci ke-	Kunci Setelah Diubah	Ciphertext Setelah Diubah	Bit Berbeda	AE (%)
1	0	d075736b65736d6 1...	bf406f5847dd9ee1 ...	64/128	50.00
2	11	5065736b65736d6 1...	b8c536ed3f43ee72 ...	63/128	49.22
3	22	5075716b65736d6 1...	95c6c0816441030 5...	60/128	46.88
4	33	5075736b25736d6 1...	c456c198a08ed98c ...	64/128	50.00
5	44	5075736b657b6d6 1...	2bbaf5a3e96c89dc ...	63/128	49.22
6	55	5075736b65736c6 1...	09e5db23b09e680 3...	60/128	46.88
7	66	5075736b65736d6 1...	027eeb4c7ccffb27 ...	66/128	51.56
8	77	5075736b65736d6 1...	ff888ae98fee423f ...	67/128	52.34
9	88	5075736b65736d6 1...	6cd29a956c42d03 d...	65/128	50.78
10	99	5075736b65736d6 1...	1cfefa1af81f6f24 ...	63/128	49.22

Nilai rata-rata avalanche effect pada skenario B adalah 49.61%, dengan nilai terendah 46,88% dan tertinggi 52,34%. Sebarannya lebih rapat dibandingkan skenario A, yang masuk akal karena perubahan satu bit pada kunci akan menjalar ke seluruh sebelas round key melalui proses key expansion, sehingga pengaruhnya lebih merata sejak putaran-putaran awal.

Perbandingan kedua skenario ditampilkan dalam bentuk grafik batang pada Gambar 4.30.



Gambar 4.30 Grafik Hasil Pengujian Avalanche Effect

Kedua skenario menghasilkan rata-rata 50.70% dan 49.61%, keduanya berada di dalam rentang 45% sampai 55%, sehingga implementasi AES-128 pada sistem ini dinyatakan memenuhi sifat difusi. Hasil ini juga sejalan dengan penelitian terdahulu. Penelitian Candra Aditya Pinuyut et al. (2024) mencatat avalanche effect AES standar sebesar 49,63%, sedangkan Sidqon dan Nafis (2024) mencatat 50,15% pada sistem e-arsip berbasis web. Kedekatan angka ini menunjukkan implementasi pada penelitian ini berperilaku sebagaimana mestinya.

4.6.4 Pengujian Performa

Pengujian performa dilakukan untuk menjawab kekhawatiran yang paling sering muncul ketika enkripsi diterapkan pada sistem yang dipakai sehari-hari, yaitu apakah sistem jadi terasa lambat. Pengukuran dilakukan di sisi server memakai `performance.now()` dari modul `perf_hooks` bawaan Node.js, yang punya ketelitian sampai di bawah satu milidetik.

Setiap skenario dijalankan 30 kali, dan sebelum pengukuran dimulai dijalankan lima iterasi pemanasan yang hasilnya dibuang. Iterasi pemanasan ini perlu karena mesin JavaScript melakukan optimasi setelah sebuah fungsi dipanggil beberapa kali, sehingga pemanggilan pertama biasanya jauh lebih lambat dan akan

mengacaukan rata-rata kalau ikut dihitung. Yang dilaporkan bukan hanya rata-rata, tetapi juga nilai minimum, maksimum, dan simpangan baku, supaya sebaran datanya terlihat.

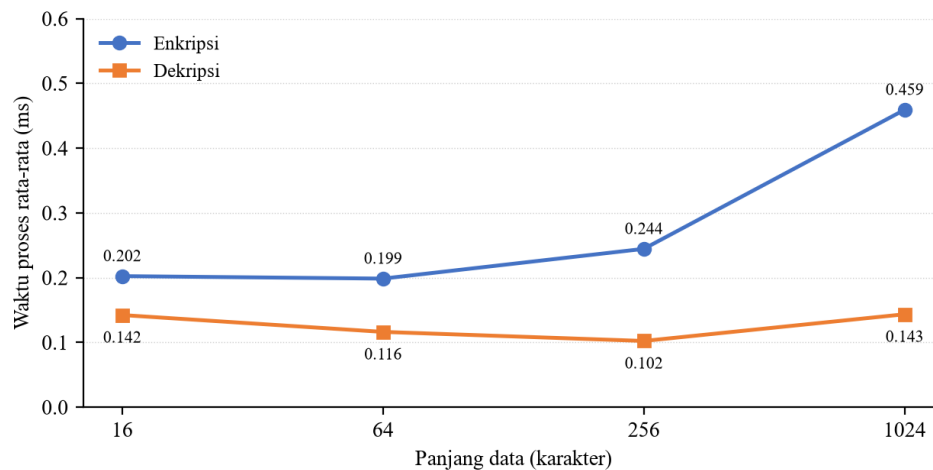
Hasil pengukuran untuk berbagai panjang data ditampilkan pada Tabel 4.25.

Tabel 4.25 Waktu Enkripsi dan Dekripsi Berdasarkan Panjang Data (30 iterasi)

Panjang Data	Operasi	Rata-rata (ms)	Minimum (ms)	Maksimum (ms)	Simpangan Baku (ms)
16 karakter	Enkripsi	0.2024	0.1453	0.5734	0.0841
16 karakter	Dekripsi	0.1423	0.0928	0.9313	0.1511
64 karakter	Enkripsi	0.1986	0.1453	0.9803	0.1522
64 karakter	Dekripsi	0.1163	0.1001	0.1917	0.0227
256 karakter	Enkripsi	0.2445	0.1680	1.1799	0.1897
256 karakter	Dekripsi	0.1023	0.0955	0.1168	0.0058
1024 karakter	Enkripsi	0.4593	0.3261	1.0875	0.1567
1024 karakter	Dekripsi	0.1435	0.1374	0.1982	0.0109

Terlihat bahwa waktu proses hampir tidak berubah untuk data 16 sampai 256 karakter, dan baru naik ketika data mencapai 1024 karakter. Pola ini masuk akal karena sebagian besar waktu justru habis untuk membangkitkan IV acak dan menyiapkan objek kriptografi, bukan untuk proses enkripsi blok itu sendiri. Data 16 karakter hanya menghasilkan dua blok setelah padding, sedangkan data 1024 karakter menghasilkan 65 blok, tetapi waktunya hanya naik sekitar dua kali lipat. Waktu dekripsi konsisten lebih cepat daripada enkripsi karena tidak perlu membangkitkan IV.

Grafik hubungan antara panjang data dan waktu proses ditampilkan pada Gambar 4.31.



Gambar 4.31 Grafik Panjang Data terhadap Waktu Proses

Pengukuran berikutnya dilakukan pada tingkat record, yaitu kondisi yang sebenarnya terjadi ketika sistem dipakai. Satu record pasien berisi lima atribut yang harus dienkripsi, sedangkan halaman daftar pasien perlu mendekripsi banyak record sekaligus. Hasilnya ditampilkan pada Tabel 4.26.

Tabel 4.26 Waktu Proses pada Tingkat Record Data Pasien (30 iterasi)

Skenario	Rata-rata (ms)	Minimum (ms)	Maksimum (ms)	Simpangan Baku (ms)
Enkripsi 1 record pasien (5 atribut)	0.5329	0.4580	0.9899	0.1094
Dekripsi 1 record pasien (5 atribut)	0.3953	0.3390	1.6478	0.2377
Dekripsi 50 record pasien (tampil daftar)	17.5758	14.7433	24.2393	2.4694

Hasil ini menjawab pertanyaan tentang dampak enkripsi terhadap kenyamanan penggunaan. Menyimpan satu data pasien hanya menambah waktu sekitar 0,53 milidetik, sedangkan menampilkan daftar berisi 50 pasien menambah sekitar 17,58 milidetik. Sebagai perbandingan, batas waktu di mana manusia mulai merasakan adanya jeda pada antarmuka adalah sekitar 100 milidetik, dan kebutuhan

non-fungsional KNF-05 menetapkan batas dua detik. Angka yang diperoleh masih jauh di bawah keduanya.

Kalau dibandingkan dengan waktu yang dibutuhkan untuk satu kali kueri ke basis data dan pengiriman data melalui jaringan yang biasanya berkisar puluhan milidetik, porsi waktu yang dipakai untuk enkripsi dan dekripsi sebenarnya kecil sekali. Dengan kata lain, penambahan waktu akibat penerapan AES-128 pada sistem ini tidak akan disadari oleh petugas puskesmas saat memakai sistem.

Hasil ini juga selaras dengan benchmark yang disebutkan pada sub-bab 3.10.3, di mana AES-128 mode CBC tercatat memiliki waktu enkripsi rata-rata 0,382 milidetik. Angka pada penelitian ini berada di kisaran yang sama, yaitu antara 0,10 sampai 0,53 milidetik tergantung skenarionya.

4.6.5 Pengujian Keamanan

Tiga pengujian tambahan dilakukan untuk memastikan bahwa pengamanan yang diterapkan benar-benar bekerja dalam skenario yang paling dikhawatirkan. Hasilnya ditampilkan pada Tabel 4.27.

Tabel 4.27 Hasil Pengujian Keamanan Sistem

N o	Skenario Pengujian	Cara Pengujian	Hasil	Ket.
1	Basis data bocor atau salinannya dicuri	Membuka tabel pasien dan rekam_medis langsung melalui klien MySQL memakai kredensial basis data yang sah	Seluruh kolom sensitif hanya menampilkan ciphertext Base64; tidak ada nama, NIK, alamat, atau diagnosa yang terbaca	Ama n
2	Dekripsi memakai kunci yang salah	Mendekripsi ciphertext yang sah memakai kunci PuskesmasBataha N yang hanya berbeda satu huruf dari kunci asli	Proses gagal dan mengembalikan string kosong; byte mentahnya berupa data acak yang tidak menyerupai data asli	Ama n
3	Akses endpoint milik	Memanggil GET /api/pengguna dan	Kedua permintaan ditolak dengan HTTP	Ama n

N o	Skenario Pengujian	Cara Pengujian	Hasil	Ket.
	admin oleh peran lain	GET /api/audit-log melalui Postman memakai sesi petugas dan sesi dokter	403 Forbidden sebelum data apa pun diambil dari basis data	
4	Kata sandi terbaca di basis data	Membuka kolom password pada tabel users	Yang tersimpan berupa hash bcrypt berawalan \$2b\$12\$ yang tidak dapat dikembalikan ke kata sandi aslinya	Ama n
5	Akses halaman tanpa autentikasi	Membuka /dashboard, /pasien, dan /pengguna tanpa sesi	Seluruhnya dialihkan otomatis ke halaman login oleh middleware	Ama n

a. Pengujian Akses Langsung ke Basis Data

Pengujian pertama menjawab skenario terburuk, yaitu apa yang terjadi kalau basis data berhasil diakses pihak lain. Tabel pasien dibuka langsung melalui klien MySQL memakai kredensial yang sah, sehingga posisi penguji setara dengan penyerang yang sudah berhasil mendapatkan akses penuh ke basis data. Hasilnya sudah ditampilkan pada Gambar 4.5, di mana seluruh kolom sensitif hanya berisi deretan karakter Base64. Tanpa kunci AES-128 yang tersimpan pada berkas .env di server, data tersebut tidak berguna bagi penyerang.

b. Pengujian Dekripsi dengan Kunci yang Salah

Pengujian kedua membuktikan bahwa kunci benar-benar menjadi penentu. Ciphertext yang sah dicoba didekripsi memakai kunci yang hanya berbeda satu huruf, yaitu huruf n terakhir diubah menjadi huruf N besar. Hasilnya ditampilkan berikut ini.

```

Plaintext asli   : Muhammad Rizky Pratama
Ciphertext      :
2ArMnihFhgETZIJUu2bvJzYIeKNGXnq3OuJt0XWZg/k=

Kunci benar  "PuskesmasBatahan" -> "Muhammad Rizky
Pratama"

```

```
Kunci salah "PuskesmasBatahaN" -> "" (string kosong,
dekripsi gagal)
```

```
Byte mentah hasil dekripsi dengan kunci salah (tanpa
pembuangan padding):
```

```
d95afbc6bc1b3c401aac655fdb188130
be3ea2b3316f169492c10adaea314e51
```

Gambar 4.32 Hasil Pengujian Dekripsi dengan Kunci yang Salah

Sistem mengembalikan string kosong, bukan data asli dan bukan pula data yang sebagian benar. Ini terjadi karena hasil dekripsi dengan kunci yang salah menghasilkan byte padding yang tidak valid, sehingga proses pembuangan padding PKCS#7 gagal. Kalau pembuangan padding dilewati, yang terlihat hanyalah byte acak seperti pada baris terakhir gambar di atas. Tidak ada satu pun potongan nama Muhammad Rizky Pratama yang bisa dikenali. Perbedaan satu huruf pada kunci sudah cukup untuk membuat data tidak bisa dipulihkan sama sekali, yang sekaligus memperkuat hasil pengujian avalanche effect pada sub-bab sebelumnya.

c. Pengujian Kontrol Akses

Pengujian ketiga dilakukan dengan memanggil endpoint milik admin memakai sesi petugas melalui Postman, tanpa melewati antarmuka sama sekali. Pengujian ini penting karena penyembunyian menu di antarmuka bukan pengamanan yang sebenarnya. Siapa pun yang tahu alamat endpoint-nya bisa memanggilnya langsung.

Sistem mengembalikan status 403 Forbidden beserta pesan Forbidden. Yang perlu diperhatikan, penolakan terjadi sebelum sistem mengambil data apa pun dari basis data, karena pemeriksaan peran dijalankan di baris paling awal fungsi. Jadi tidak ada data yang sempat terbaca lalu baru disaring, melainkan permintaannya memang langsung dihentikan.

4.7 Pembahasan

Sub-bab ini membahas makna dari hasil-hasil pengujian di atas, mengaitkannya dengan teori pada Bab II, membandingkannya dengan penelitian

terdahulu, serta menjelaskan secara terbuka apa saja keterbatasan sistem yang dibangun.

4.7.1 Analisis Hasil Pengujian

Dari seluruh pengujian yang dilakukan, ada empat kesimpulan yang bisa ditarik.

Pertama, dari sisi fungsionalitas, seluruh 38 skenario pengujian black box menghasilkan keluaran yang sesuai harapan sehingga tingkat keberhasilannya 100%. Seluruh kebutuhan fungsional KF-01 sampai KF-17 sudah terwakili, termasuk 18 skenario negatif yang memang dirancang untuk gagal. Ini berarti sistem tidak hanya bisa menjalankan tugasnya, tetapi juga bisa menolak hal-hal yang seharusnya ditolak.

Kedua, dari sisi kebenaran algoritma, implementasi AES-128 pada sistem menghasilkan ciphertext yang sama persis dengan vektor uji resmi NIST FIPS 197. Ini berarti algoritma yang berjalan memang AES-128 sesuai standar, bukan sekadar fungsi pengacak yang kebetulan bisa dibalik. Uji bolak-balik pada sepuluh data uji juga seluruhnya berhasil tanpa satu pun karakter yang berubah.

Ketiga, dari sisi kualitas kriptografis, avalanche effect rata-rata mencapai 50.70% untuk perubahan bit plaintext dan 49.61% untuk perubahan bit kunci. Keduanya berada di dalam rentang wajar 45% sampai 55% dan sangat dekat dengan nilai ideal 50%. Ini berarti tidak ada hubungan sederhana yang bisa dipelajari penyerang antara masukan dan keluaran algoritma.

Keempat, dari sisi performa, penambahan waktu akibat enkripsi rata-rata hanya 0,53 milidetik untuk menyimpan satu data pasien dan 17,58 milidetik untuk menampilkan daftar 50 pasien. Angka ini jauh di bawah ambang 100 milidetik yang mulai terasa oleh pengguna, apalagi di bawah batas dua detik yang ditetapkan pada KNF-05. Artinya, keamanan yang diperoleh dari penerapan AES-128 tidak dibayar dengan penurunan kenyamanan penggunaan.

4.7.2 Analisis Keamanan Sistem

Hasil pengujian di atas dapat dipetakan ke tiga aspek keamanan informasi yang dijelaskan pada sub-bab 2.5, yaitu kerahasiaan, integritas, dan ketersediaan atau yang biasa disebut CIA Triad.

Aspek kerahasiaan dipenuhi melalui penerapan AES-128 mode CBC pada seluruh atribut sensitif. Buktinya ada pada dua tempat, yaitu Gambar 4.5 yang memperlihatkan isi basis data hanya berupa ciphertext, dan pengujian keamanan nomor 1 dan 2 pada Tabel 4.27 yang membuktikan data tidak bisa dipulihkan tanpa kunci yang tepat. Penggunaan mode CBC dengan IV acak per record juga menutup celah kebocoran pola yang akan terjadi kalau memakai mode ECB, sebagaimana dibuktikan pada Tabel 4.17. Ini penting karena kerahasiaan tidak hanya soal isi data, tetapi juga soal pola yang bisa disimpulkan dari data.

Aspek integritas dipenuhi melalui tabel `audit_log` yang mencatat setiap aksi terhadap data pasien dan rekam medis lengkap dengan identitas pelaku, waktu, dan alamat IP. Kalau ada data yang berubah tanpa sepengetahuan pihak puskesmas, catatan ini bisa dipakai untuk menelusuri siapa yang mengubahnya dan kapan. Perlu dicatat secara jujur bahwa mode CBC sendiri tidak menyediakan jaminan integritas ciphertext. Kalau seseorang mengubah byte ciphertext di basis data, sistem tidak akan tahu bahwa data tersebut sudah diubah. Keterbatasan ini dibahas lebih lanjut pada sub-bab 4.7.4 dan menjadi salah satu saran pengembangan pada Bab V.

Aspek ketersediaan dan kontrol akses dipenuhi melalui penerapan RBAC dengan tiga peran dan autentikasi berbasis hash bcrypt. Data tetap tersedia bagi pihak yang berhak, karena proses dekripsi berjalan otomatis dan cepat, tetapi tertutup bagi yang tidak berhak. Pengujian nomor 3, 4, dan 5 pada Tabel 4.27 membuktikan bahwa pembatasan ini berjalan di sisi server, bukan hanya di tampilan.

Selain itu, penerapan sistem ini juga mendukung pemenuhan kewajiban Puskesmas Batahan sebagai pengendali data pribadi menurut UU No. 27 Tahun 2022. Data kesehatan yang tergolong data pribadi spesifik sudah dilindungi dengan

enkripsi, akses terhadapnya dibatasi berdasarkan kebutuhan tugas, dan setiap akses tercatat.

4.7.3 Perbandingan dengan Penelitian Terdahulu

Perbandingan penelitian ini dengan penelitian terdahulu yang sudah dibahas pada sub-bab 2.14 ditampilkan pada Tabel 4.28.

Tabel 4.28 Perbandingan dengan Penelitian Terdahulu

Peneliti (Tahun)	Objek Penelitian	Algoritma dan Mode	Pengujian yang Dilakukan	Perbedaan dengan Penelitian Ini
Ramadhani & Cobantoro (2022)	Basis data sistem informasi umum	AES-128, mode tidak disebutkan spesifik	Pengujian fungsional dan waktu proses	Penelitian ini menambahkan uji vektor NIST, avalanche effect, dan diterapkan pada objek puskesmas
Hakim & Margolang (2025)	Rekam medis digital	AES	Analisis keamanan secara kualitatif	Penelitian ini menghasilkan sistem berbasis web yang berjalan, bukan hanya analisis, dan disertai pengukuran kuantitatif
Tamin & Hendrik (2025)	Berkas rekam medis pasien	AES-128	Uji enkripsi dan dekripsi berkas	Penelitian ini mengkripsi pada tingkat kolom basis data, bukan berkas, sehingga data tetap bisa dikelola secara terstruktur

Peneliti (Tahun)	Objek Penelitian	Algoritma dan Mode	Pengujian yang Dilakukan	Perbedaan dengan Penelitian Ini
Candra Aditya Pinuyut et al. (2024)	Data teks umum	AES termodifika si	Avalanche effect dan throughput	Penelitian ini memakai AES-128 standar tanpa modifikasi dan menambahkan verifikasi terhadap vektor uji NIST FIPS 197
Prabowo & Nizirwan (2025)	Simulasi algoritma	AES-128 mode ECB dan CBC	Perbandingan avalanche effect antar mode	Penelitian ini menerapkan mode CBC pada sistem nyata dan membuktikan kelemahan ECB memakai data pasien
Sidqon & Nafis (2024)	Sistem e- arsip KPU Sidoarjo	AES berbasis web	Avalanche effect dan uji keandalan	Penelitian ini menambahkan kontrol akses berbasis peran dan pencatatan audit log yang tidak dibahas pada penelitian tersebut

Dari perbandingan di atas terlihat bahwa penelitian ini mengisi beberapa celah sekaligus. Penelitian terdahulu umumnya berhenti pada salah satu dari tiga hal, yaitu hanya menganalisis tanpa membangun sistem, hanya menguji algoritma tanpa menerapkannya pada kasus nyata, atau membangun sistem tetapi tanpa

pengujian kriptografis yang terukur. Penelitian ini menggabungkan ketiganya, yaitu membangun sistem yang berjalan, menerapkan AES-128 mode CBC pada data pasien nyata di puskesmas, dan mengujinya dengan empat metode sekaligus mulai dari black box sampai verifikasi terhadap standar NIST.

Selain itu, sejauh penelusuran literatur yang dilakukan, belum ditemukan penelitian yang menerapkan AES-128 pada sistem informasi puskesmas dengan menggabungkan tiga lapisan pengamanan sekaligus, yaitu enkripsi data, kontrol akses berbasis peran, dan pencatatan jejak audit. Kebanyakan penelitian hanya berfokus pada lapisan enkripsinya saja.

4.7.4 Kelebihan dan Keterbatasan Sistem

Bagian ini menyampaikan secara terbuka apa yang sudah bisa dilakukan sistem dan apa yang belum. Keterbatasan yang disebutkan di sini bukan berarti sistem gagal, melainkan batas ruang lingkup yang memang sudah ditetapkan sejak Bab I, dan sebagian di antaranya menjadi bahan saran pengembangan pada Bab V. Ringkasannya ditampilkan pada Tabel 4.29.

Tabel 4.29 Kelebihan dan Keterbatasan Sistem

Aspek	Kelebihan	Keterbatasan
Kerahasiaan data	Seluruh atribut sensitif pasien dan rekam medis terenkripsi AES-128 CBC dengan IV acak per record; terverifikasi terhadap vektor uji NIST FIPS 197	Kolom no_rm dan tanggal_periksa sengaja tidak dienkripsi karena dibutuhkan untuk pencarian dan pengurutan
Pengelolaan kunci	Kunci tidak ditulis di dalam kode, disimpan pada berkas .env di luar repositori, dan panjangnya divalidasi saat aplikasi dijalankan	Hanya ada satu kunci untuk seluruh data dan belum ada mekanisme pergantian kunci berkala; kalau kunci bocor, seluruh data ikut terdampak

Aspek	Kelebihan	Keterbatasan
Integritas data	Setiap aksi tercatat di audit_log lengkap dengan pelaku, waktu, dan alamat IP	Mode CBC tidak menyediakan verifikasi keutuhan ciphertext, sehingga perubahan byte ciphertext secara langsung di basis data tidak terdeteksi sistem
Kontrol akses	RBAC tiga peran diterapkan di dua lapisan, yaitu middleware halaman dan pemeriksaan peran pada setiap endpoint API	Belum ada autentikasi dua faktor dan belum ada pembatasan jumlah percobaan login yang gagal
Pencarian data	Pencarian berdasarkan nama dan nomor rekam medis tetap dapat dilakukan meskipun data tersimpan terenkripsi	Pencarian pada kolom terenkripsi tidak bisa memakai perintah LIKE di SQL, sehingga seluruh record harus didekripsi dulu; cara ini akan melambat kalau jumlah pasien mencapai puluhan ribu
Performa	Penambahan waktu akibat enkripsi hanya sekitar 0,53 ms per record, tidak terasa oleh pengguna	Pengujian dilakukan pada satu lingkungan lokal dengan satu pengguna, belum diuji pada kondisi banyak pengguna bersamaan
Penyimpanan	Tipe data TEXT dipilih sejak awal sehingga ciphertext tidak pernah terpotong	Ukuran penyimpanan menjadi sekitar dua kali lipat dibanding data asli akibat padding dan pengodean Base64
Cadangan data	Data pada basis data utama sudah tersimpan dalam keadaan terenkripsi	Belum ada mekanisme pencadangan basis data terjadwal yang otomatis

Keterbatasan yang paling perlu diperhatikan pihak puskesmas adalah soal pengelolaan kunci. Seluruh keamanan sistem ini pada akhirnya bertumpu pada satu

kunci sepanjang 16 karakter yang tersimpan di berkas .env pada server. Sekuat apa pun algoritmanya, kalau berkas tersebut bisa diakses pihak lain maka seluruh data pasien bisa dipulihkan. Karena itu pengamanan akses ke server dan pergantian kunci secara berkala menjadi hal yang tidak bisa diabaikan, dan hal ini menjadi salah satu saran pada Bab V.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, kesimpulan penelitian ini disusun untuk menjawab ketiga rumusan masalah yang telah ditetapkan sebelumnya secara berurutan.

1. Algoritma AES-128 berhasil diterapkan untuk melakukan proses enkripsi dan dekripsi data pasien pada Puskesmas Batahan. Enkripsi dijalankan pada mode CBC dengan padding PKCS#7 dan Initialization Vector acak sepanjang 16 byte yang dibangkitkan ulang setiap kali data disimpan. Atribut yang dienkrpsi meliputi nama, NIK, tanggal lahir, alamat, dan nomor telepon pada tabel pasien, serta keluhan, diagnosa, tindakan, dan resep pada tabel rekam_medis. Kebenaran implementasi dibuktikan melalui dua cara. Pertama, hasil enkripsi sistem diuji terhadap vektor uji resmi NIST FIPS PUB 197 dan menghasilkan ciphertext 69c4e0d86a7b0430d8cdb78070b4c55a yang cocok persis dengan nilai yang ditetapkan standar. Kedua, uji bolak-balik pada sepuluh data uji berhasil mengembalikan seluruh data ke bentuk aslinya tanpa satu pun karakter yang berubah. Perhitungan manual satu blok penuh sebanyak sepuluh putaran juga sudah diuraikan pada sub-bab 4.4 dan hasilnya konsisten dengan keluaran sistem.
2. Sistem keamanan data pasien berbasis web berhasil dirancang dan dibangun menggunakan Next.js, React, Prisma, dan MySQL 8.0. Sistem mencakup tujuh halaman utama, yaitu login, dashboard, data pasien, rekam medis, manajemen pengguna, log aktivitas, dan demo enkripsi, serta empat tabel basis data yaitu users, pasien, rekam_medis, dan audit_log. Pengamanan tidak hanya bertumpu pada enkripsi, tetapi disusun berlapis: enkripsi AES-128 untuk kerahasiaan data, hash bcrypt dengan cost factor 12 untuk kata sandi pengguna, kontrol akses berbasis peran dengan tiga peran admin,

dokter, dan petugas yang diterapkan di sisi halaman maupun endpoint API, serta pencatatan jejak audit untuk setiap aksi terhadap data. Seluruh 38 skenario pengujian black box yang mencakup ketujuh belas kebutuhan fungsional menghasilkan keluaran sesuai harapan, sehingga tingkat keberhasilan pengujian mencapai 100%.

3. Penerapan algoritma AES-128 terbukti meningkatkan keamanan data pasien di Puskesmas Batahan. Data identitas dan rekam medis yang sebelumnya tersimpan dalam bentuk yang bisa langsung dibaca kini tersimpan sebagai ciphertext, sehingga pengujian akses langsung ke basis data hanya menampilkan deretan karakter acak tanpa satu pun data yang terbaca. Pengujian dekripsi dengan kunci yang hanya berbeda satu karakter juga gagal memulihkan data. Dari sisi kualitas algoritma, avalanche effect rata-rata mencapai 50.70% pada perubahan satu bit plaintext dan 49.61% pada perubahan satu bit kunci, keduanya berada dalam rentang wajar 45% sampai 55% sehingga memenuhi sifat difusi. Peningkatan keamanan ini tidak mengorbankan kenyamanan penggunaan, karena penambahan waktu akibat enkripsi hanya sekitar 0,53 milidetik untuk menyimpan satu data pasien dan 17,58 milidetik untuk menampilkan daftar 50 pasien, jauh di bawah batas dua detik yang ditetapkan pada kebutuhan non-fungsional.

5.2 Saran

Berdasarkan keterbatasan sistem yang sudah diuraikan pada hasil penelitian, saran dibagi menjadi dua kelompok, yaitu saran yang bisa langsung dijalankan pihak Puskesmas Batahan dan saran untuk penelitian selanjutnya.

a. Saran bagi Puskesmas Batahan

1. Sebelum sistem dipakai, sebaiknya diadakan pelatihan singkat bagi petugas pendaftaran dan tenaga medis. Materi pelatihan cukup mencakup cara login, cara menginput data, dan yang paling penting pemahaman bahwa akun tidak boleh dipakai bersama-sama. Sistem sudah mencatat setiap aktivitas beserta pelakunya, tetapi catatan itu jadi tidak berguna kalau satu akun dipakai banyak orang.

2. Berkas .env yang berisi kunci AES-128 harus disimpan dengan pengamanan yang setara dengan data pasien itu sendiri. Akses ke server tempat berkas tersebut berada perlu dibatasi hanya untuk petugas yang benar-benar berwenang, dan kunci sebaiknya diganti secara berkala, misalnya setiap enam bulan atau setiap ada pergantian petugas yang memegang akses server.
3. Pencadangan basis data perlu dijadwalkan secara rutin. Karena data di dalam basis data sudah tersimpan dalam keadaan terenkripsi, berkas cadangannya pun otomatis ikut terenkripsi. Namun perlu diingat bahwa berkas cadangan tidak ada gunanya kalau kuncinya hilang, sehingga cadangan kunci juga harus disimpan terpisah di tempat yang aman.
4. Log aktivitas pada halaman audit log sebaiknya diperiksa secara berkala oleh admin, bukan hanya dibuka ketika sudah terjadi masalah. Pemeriksaan rutin bisa membantu mengenali pola akses yang tidak wajar sejak awal, misalnya akses di luar jam pelayanan atau dari alamat IP yang tidak dikenal.
5. Ketika sistem nantinya dijalankan pada jaringan yang bisa diakses dari luar puskesmas, penggunaan HTTPS menjadi keharusan. Enkripsi pada penelitian ini melindungi data yang tersimpan di basis data, sedangkan data yang sedang berpindah melalui jaringan dilindungi oleh HTTPS. Keduanya saling melengkapi dan tidak bisa saling menggantikan.

b. Saran bagi Penelitian Selanjutnya

1. Penggunaan mode operasi yang menyediakan autentikasi data seperti AES-GCM dapat dipertimbangkan. Mode CBC yang dipakai pada penelitian ini hanya menjamin kerahasiaan data, sedangkan AES-GCM sekaligus menghasilkan tag autentikasi sehingga sistem bisa mengetahui apabila ciphertext diubah orang lain. Ini akan menutup keterbatasan pada aspek integritas yang sudah disebutkan pada sub-bab 4.7.4. Peningkatan panjang kunci ke AES-256 juga dapat diteliti, meskipun perlu diukur ulang dampaknya terhadap performa.
2. Pengelolaan kunci yang lebih baik perlu diteliti lebih lanjut, misalnya penerapan rotasi kunci otomatis, penggunaan kunci berbeda untuk setiap record dengan kunci utama sebagai pembungkus, atau pemanfaatan layanan

pengelolaan kunci seperti Key Management Service. Sistem pada penelitian ini masih memakai satu kunci tunggal untuk seluruh data.

3. Penerapan searchable encryption atau indeks terenkripsi dapat diteliti untuk mengatasi keterbatasan pencarian. Pada sistem ini seluruh record harus didekripsi dulu sebelum bisa disaring, dan cara ini akan melambat kalau jumlah pasien sudah mencapai puluhan ribu. Teknik seperti blind index atau deterministic encryption pada kolom tertentu bisa menjadi alternatif yang layak diuji.
4. Pengujian performa pada kondisi banyak pengguna bersamaan perlu dilakukan. Pengujian pada penelitian ini masih dijalankan di lingkungan lokal dengan satu pengguna. Pengujian beban memakai alat seperti k6 atau Apache JMeter akan memberikan gambaran yang lebih mendekati kondisi nyata ketika beberapa petugas memakai sistem pada waktu yang sama.
5. Penambahan autentikasi dua faktor dan pembatasan jumlah percobaan login yang gagal dapat diteliti untuk memperkuat lapisan autentikasi, mengingat sistem saat ini masih mengandalkan kombinasi username dan kata sandi saja.
6. Integrasi dengan SIMPUS atau platform SATUSEHAT milik Kementerian Kesehatan dapat menjadi arah pengembangan berikutnya, dengan catatan perlu diteliti terlebih dahulu bagaimana data yang tersimpan dalam keadaan terenkripsi dapat dipertukarkan dengan sistem lain tanpa mengurangi tingkat keamanannya.

DAFTAR PUSTAKA

- Alaikha Annan. (2024). Tinjauan Yuridis Perlindungan Data Pribadi Pada Sektor Kesehatan Berdasarkan Undang-Undang No. 27 Tahun 2022. *Jurnal Ilmiah Multidisiplin*, 1(4), 247–254.
- Bakri, S. N., Irwan, M., Nasution, P., Islam, U., & Sumatera, N. (2024). Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem. *Jurnal Sistem Informasi Dan Teknologi Informasi Komputer*, 3(1), 53–66. <https://doi.org/10.53624/jsitik.v3i1.542>
- Basri, M., & Sulaiman, K. (2024). Encryption of URL Address (Uniform Resource Locator) on PHP Web Base with MD5 Algorithm. *Proceeding of International Conference on Science and Technology UISU*, 7929, 12–16.
- Candra Aditya Pinuyut, Ema Utami, & Alva Hendi Muhammad. (2024). Analisis Kinerja Algoritma Advanced Encryption Standard (Aes) Termodifikasi Dalam Enkripsi Dan Dekripsi Data. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 5(2), 132–137. <https://doi.org/10.46764/teknimedia.v5i2.200>
- Donald L. Evans, Phillip J. Bond, K. H. B. (2022). Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication 4*, 197. <https://doi.org/10.6028/NIST.FIPS.197-upd1.ipd11>
- Ganesh, R., Khan, B. U. I., Khan, A. R., & Kamsin, A. Bin. (2025). A panoramic survey of the advanced encryption standard: from architecture to security analysis, key management, real-world applications, and post-quantum challenges. In *International Journal of Information Security* (Vol. 24, Issue 5). <https://doi.org/10.1007/s10207-025-01116-x>
- Hakim, A. R., & Margolang, K. F. (2025). Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Analisis Keamanan Data Rekam Medis Digital Menggunakan Algoritma Kriptografi AES Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD. *Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD*, 8, 108–114. <https://doi.org/10.53513/jsk.v8i2.11461>
- Hidayatuloh, F., Amila, Y., Naufal, M., Akbar, N., & Maesaroh, S. (2026). Komparasi Performa dan Keamanan Algoritma AES-128 dan Blowfish pada

- Enkripsi Berkas Teks. *Jurnal Ilmiah Multidisiplin*, 2(1), 1674–1677.
- Indriyajati, F., Margarith, M., Damiana, S., Utomo, H., Mada, U. G., Mada, U. G., & Mada, U. G. (2024). Analisis Keamanan Data Electronic Medical Record Digital Transformation Office (DTO) Kementerian Kesehatan Indonesia. *Sanskara Manajemen Dan Bisnis*, 02(01), 59–66. <https://doi.org/10.58812/smb.v2i01>
- Prabowo, W., & Nizirwan, A. (2025). Pengujian Model Simulasi Efek Avalanche Kriptografi Simetris Algoritma AES 128-bit, Mode ECB dan CBC. *Jurnal Ikraith-Infomatika*, 9(1), 178–186.
- Puteri, D., Pramesti, A., Ayuningtyas, D., & Verdi, R. (2024). Keamanan Dan Kerahasiaan Data Medis Pasien Dalam Implementasi Rekam Medis Elektronik : *Jurnal Kesehatan Masyarakat*, 8, 7691–7702. <https://doi.org/10.31004/prepotif.v8i3.38445>
- Ramadhani, T. A., Cobantoro, A. F., Informatika, T., Teknik, F., & Ponorogo, U. M. (2022). Implementasi Algoritma Advanced Encryption Standard 128 Untuk Pengamanan Database Sistem. *Jurnal Informatika Polinema*, 521–526. <https://doi.org/10.33795/jip.v10i4.5619>
- Rospita Adelina Siregar, H. S. R. S. (2025). Aspek Hukum Perlindungan Data Pasien Dalam. *Jurnal Hukum To-Ra : Hukum Untuk Mengatur Dan Melindungi Masyarakat*, 11(1), 106–116.
- Saravanos, A. (2023). Simulating the Software Development Lifecycle : The Waterfall Model. *Applied System Innovation*. <https://doi.org/10.3390/asi6060108>
- Sidqon, M., & Nafis, R. O. (2024). Rancang Bangun Sistem E-Arsip Berbasis Website Menggunakan Metode Enkripsi Aes (Advanced Encryption Standard) Studi Kasus Kpu Sidoarjo. *Jurnal Rekayasa Sistem Informasi Dan Teknologi*, 2(1), 488–497.
- Siregar, F. A., Informasi, T., & Utara, M. S. (2024). Implementation of the Arnold Catmap on a Combination of Symmetric and Asymmetric Cryptography. *Jurnal Dan Penelitian Teknik Informatika*, 8(1), 174–182.
- Tamin, Z., Hendrik, B., Komputer, F. I., Putra, U., & Yptk, I. (2025). Jurnal

- KomtekInfo Penerapan Algoritme Advanced Encryption Standard (AES-128) untuk Mengamankan File Rekam Medis Pasien. *Jurnal KomtekInfo*, 12, 22–30. <https://doi.org/10.35134/komtekinfo.v12i1.592>
- Utomo, A., Sutanto, Y., Tiningrum, E., & Susilowati, E. M. (2020). Menggunkan Black Box Testing Boundary. *Jurnal Bisnis Terapan*, 04, 133–140. <https://doi.org/10.24123/jbt.v4i2.2170>
- Vaiyapuri, T., Binbusayyis, A., & Varadarajan, V. (2021). Security , Privacy and Trust in IoMT Enabled Smart Healthcare System : A Systematic Review of Current and Future Trends. *International Journal of Advanced Computer Science and Applications*, 12(2)
- Barker, E. (2020). Recommendation for Key Management: Part 1 – General (NIST Special Publication 800-57 Part 1, Revision 5). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-57pt1r5>
- Dworkin, M. J. (2001). Recommendation for Block Cipher Modes of Operation: Methods and Techniques (NIST Special Publication 800-38A). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-38A>
- International Organization for Standardization. (2025). ISO 27799:2025 Health informatics—Information security controls in health based on ISO/IEC 27002. ISO. <https://www.iso.org/standard/84647.html>
- Joint Task Force. (2020). Security and Privacy Controls for Information Systems and Organizations (NIST Special Publication 800-53, Revision 5). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-53r5>



UMSU
Unggul | Cerdas | Terpercaya

Wala membebas surat ini agar disebarkan
nomor dan tanggapan

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi Unggul Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 174/SK/BAN-PT/Ak.Ppy/PT/2024

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

<https://fidi.umsu.ac.id>

fidi@umsu.ac.id

[fumsu](#) [umsu](#) [umsu](#)

[umsu](#) [umsu](#) [umsu](#)

[umsu](#) [umsu](#) [umsu](#)

[umsu](#) [umsu](#) [umsu](#)

FORMULIR PERBAIKAN UJIAN SKRIPSI

Pada hari ini, Selasa 1 September 2026 telah dilaksanakan Ujian Skripsi bagi mahasiswa

Fakultas Ilmu Komputer Dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.

Sbb:

Nama Mahasiswa : Charisya Silvara Maligan

NPM : 2209010203

Program Studi : Sistem Informasi

Judul Proposal : Pengamanan Data Pasien Pada Puskesmas Batahan Menggunakan
Algoritma AES-128 Berbasis Web

Materi/Point yang Diperbaiki :

		Paraf
Mhd. Basri, S.Si. M.Kom	OK	
Dr. Al-Khowarizmi, M.Kom.		
Farid Akbar Siregar S.Kom., M.Kom.	Kesimpulan Tambahkan perforan	

Berita acara ini ditandatangani setelah skripsi diperbaiki sesuai petunjuk/arahan dari Pembimbing dan Penguji/Pembahas.



UMSU

Unggul | Cerdas | Terpercaya

Umsu merupakan surat yang diterbitkan
nomor dan tanggal

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi Unggul Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 174/SK/BAN-PT/Ak.Ppy/PT/BB/2024

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

<http://fkip.umsumed.ac.id>

fkip@umsumed.ac.id

[f umsumedan](#)

[@umsumedan](#)

[umsumedan](#)

[umsumedan](#)

Nomor : 836/IL.3-AU/UMSU-09/F/2026
Lampiran : -
Perihal : **IZIN PENELITIAN**

Medan, 20 Shafar 1448 H
03 Agustus 2026 M

Kepada Yth.

Bapak/Ibu Pimpinan

Puskesmas Batahan

Batahan, Kecamatan Batahan, Kabupaten Mandailing Natal, Provinsi Sumatera Utara

Di Tempat

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan hormat, sehubungan mahasiswa kami akan menyelesaikan studi, untuk itu kami memohon kesediaan Bapak / Ibu untuk memberikan kesempatan pada mahasiswa kami melakukan riset di **Perusahaan / Instansi** yang Bapak / Ibu pimpin, guna untuk penyusunan skripsi yang merupakan salah satu persyaratan dalam menyelesaikan Program Studi Strata Satu (S-1)

Adapun Mahasiswa/i di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara tersebut adalah:

Nama : Charisya Silvara Maligan
Npm : 2209010203
Jurusan : Sistem Informasi
Semester : VIII (Delapan)
Judul : Pengamanan Data Pasien pada Puskesmas Batahan Menggunakan Algoritma AES-128 Berbasis Web
Email : charisyamaligan@gmail.com
Hp/Wa : 082183307372
Dosen
Pembimbing : Mhd. Basri, S.Si., M.Kom.

Demikianlah surat kami ini, atas perhatian dan kerjasamanya yang Bapak / Ibu berikan kami ucapkan terimakasih

Wassalamu 'alaikum Warahmatullahi Wabarakatuh



Dekan

Asst. Prof. Dr. Al-Khowarizmi, M.Kom.
NIDN. 0127099201

Cc:File



CV. kampus Akademik Publishing

PUSAT PUBLIKASI JURNAL ILMIAH

Berdasarkan Akta Notaris NOOR MUJAHID ABABILLYANTO, SH, MKn
No AHU-0054285-AH.01.14 Tanggal 30 Agustus Tahun 2023
Jl Pedurungan Kidul IV RT 03 RW 01 No 62 Kel .Pedurungan Kidul , Kec Pedurungan
Kota Semarang 50192 HP 089669609417, HP 089652776467
Email : office@kampusakademik.co.id
website : ejurnal.kampusakademik.co.id

LETTER OF ACCEPTANCE

No.12917/L/JSSR/VI/2026

Dear Author : Charisya Silvara Maligan, Mhd. Basri

Di Universitas Negeri Surabaya

Dengan hormat,

Kami dari Redaksi Jurnal Sains Student Research menyampaikan bahwa artikel bapak/ibu dengan judul:

PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN MENGUNAKAN ALGORITMA AES-128 BERBASIS WEB

telah diterima dan sudah direview dan dinyatakan diterima (ACCEPTED) dan akan diterbitkan di Volume 4 Nomor 6 Edisi Desember 2026 dengan e-ISSN : 3025-9851 p-ISSN : 3025-986X.

Kami mengucapkan terimakasih banyak atas kepercayaan bapak/ibu untuk menerbitkan artikel terbaik, kami akan kembali menginformasikan tahap proses berikutnya sampai publish (terbit). dan untuk seterusnya kami masih menunggu artikel terbaik saudara selanjutnya.

Demikianlah surat keterangan ini kami perbuat untuk dapat dipergunakan sebagaimana perlunya.

Semarang, 05 Oktober 2026
Editor Chief Journal

Hendri Rasmino, S.Kom., M.Si

JOURNAL SAINS STUDENT RESEARCH

← Back to Submissions

12917 / Charisya Silvara Mailigan et al. / PENGAMANAN DATA PASIEN PADA PUSKESMAS BATAHAN MENGGUNAKAN ALGORITMA AES-

Library

Workflow

Publication

Submission

Review

Copyediting

Production

Submission Files

43648

Charisya Silvara Mailigan et al DOI AKTIF.docx

October 4, 2026

Article Text

Download All Files

Search

Pre-Review Discussions

Add discussion

Name	From	Last Reply	Replies	Closed
No Items				

skripsi-charisya-new-45_1789222193538.pdf

ORIGINALITY REPORT

14%

SIMILARITY INDEX

PRIMARY SOURCES

1	repository.umsu.ac.id Internet	217 words — 1%
2	dspace.uil.ac.id Internet	158 words — 1%
3	kc.umn.ac.id Internet	143 words — 1%
4	repository.ub.ac.id Internet	114 words — < 1%
5	text-id.123dok.com Internet	80 words — < 1%
6	Irwan Irwan, Alfiansyah Hasibuan, Kristofel Santa. "Implementasi Algoritma AES-128 Untuk Keamanan Dokumen Soal Ujian di SMA Negeri 1 Kabawo", RIGGS: Journal of Artificial Intelligence and Digital Business, 2026 Crossref	76 words — < 1%
7	eprints.polsri.ac.id Internet	74 words — < 1%
8	www.coursehero.com Internet	73 words — < 1%
9	repository.dinamika.ac.id Internet	71 words — < 1%
10	repositori.usu.ac.id Internet	69 words — < 1%
11	cdn.juris.id Internet	62 words — < 1%



PEMERINTAH KABUPATEN MANDAILING NATAL

DINAS KESEHATAN
UPTD PUSKESMAS BATAHAN



Jl. Raya Batahan – Natal Desa Sari Kenanga Kode Pos 229s88

Telepon : 082272571267 Faksimile :

Pos-el : pkmbatahan@gmail.com, Laman :

Nomor : 440/ /PB/VIII/2026

Kepada Yth. Bapak/Ibu Pimpinan
Universitas Muhammadiyah Sumatera Utara
Fakultas Ilmu Komputer dan teknologi Informasi
Jurusan Sitem Informasi

Di Medan

Perihal : Surat Keterangan Selesai Penelitian

Dengan ini menyatakan bahwa :

Nama : Charisya Silvara Maligan
NPM : 2209010203
Jurusan : Sistem Informasi
Judul Penelitian : Pengamanan Data Pasien Pada Puskesmas Batahan
Menggunakan Algoritma AES-128 Berbasis Web

Telah Menyelesaikan Penelitian, Mahasiswa ditempatkan di UPTD Puskesmas Batahan.
Demikian kami sampaikan surat keterangan ini, Atas Perhatiannya kami ucapkan
terimakasih.

Kepala Puskesmas Batahan



WIRA MURTI, S. Keb, Bdn.M.K.M
NIP.19770828 2007 01 2005



UMSU

Unggul | Cerdas | Terpercaya
Bila mengabdikan suri agar disebarkan
nama dan tanggapnya

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
<https://fiki.umsu.ac.id> fiki@umsu.ac.id [umsu.medan](https://www.facebook.com/umsu.medan) [umsu.medan](https://www.instagram.com/umsu.medan) [umsu.medan](https://www.youtube.com/channel/UC...) [umsu.medan](https://www.tiktok.com/@umsu.medan)

Berita Acara Pembimbingan Proposal

Nama Mahasiswa : Charisya Silvara Maligar
NPM : 2209010203
Nama Dosen Pembimbing : Mhd. Basri, S.Si, M.Kom
Program Studi : Sistem Informasi
Judul Penelitian : pengamanan data pasien
pada puskesmas Batahan menggunakan
kan algoritma AES-128 Berbasis
Web.

Tanggal Bimbingan	Hasil Evaluasi	Paraf Dosen
25 Mei 2026	Bimbingan Bab 1, revisian pada latar belakang	
29 Mei 2026	Bimbingan bab 11, revisian	
3 Juni 2026	Bimbingan bab 1-3, revisian	
15 Juni 2026	acc sempro	
3 Juli 2026	bimbingan bab 4	
10 Juli 2026	bimbingan bab 5	
21 Juli 2026	Bimbingan bab 1-5, exhib, dan revisian	

Diketahui oleh :
Ketua Program Studi

(.....)

Medan,.....

Disetujui oleh :
Dosen Pembimbing

(.....)





UMSU
Unggul | Cerdas | Terpercaya
Bila mendapat surat ini agar disebutkan nomor dan tanggalnya

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
<http://fkti.umsu.ac.id> fkti@umsu.ac.id [umsumedan](https://www.facebook.com/umsumedan) [umsumedan](https://www.instagram.com/umsumedan) [umsumedan](https://www.youtube.com/channel/UCumsumedan) [umsumedan](https://www.linkedin.com/company/umsumedan)

Berita Acara Pembimbingan Proposal

Nama Mahasiswa : Charisya Silvan Maligan Program Studi : sistem Informasi
NPM : 2209010203 Judul Penelitian : Pengamanan data pasien
Nama Dosen Pembimbing : Mhd. Bari S.Si, M.Kom pada pustkesmas Ratahan
menggunakan algoritma AES-128
Berbasis web.

Tanggal Bimbingan	Hasil Evaluasi	Paraf Dosen
27 Juli 2026	acc sidang,	

Medan, ...

Diketahui oleh :
Ketua Program Studi

(.....)

Disetujui oleh :
Dosen Pembimbing

(.....)

