BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi di era digital telah memberikan dampak signifikan di berbagai sektor, termasuk kesehatan. Digitalisasi data melalui penerapan rekam medis elektronik (*Electronic Medical Records*/EMR) telah menggantikan metode tradisional berbasis kertas, meningkatkan efisiensi layanan kesehatan, dan memudahkan tenaga medis dalam mengakses serta memperbarui data pasien (Ikawati, 2024). Namun, penerapan teknologi dalam pengelolaan data medis juga menghadirkan tantangan besar terkait keamanan dan privasi data pasien. Rekam medis mengandung informasi pribadi yang sensitif, sehingga perlu dilindungi dari ancaman kebocoran data. Tanpa perlindungan memadai, kebocoran data dapat menimbulkan pelanggaran privasi, mengurangi kepercayaan pasien terhadap layanan kesehatan, serta merusak reputasi institusi kesehatan (Ardianto & Nurjanah, 2024)

Kasus kebocoran data medis di Indonesia mempertegas pentingnya penerapan sistem keamanan yang lebih baik. Pada Januari 2022, dilaporkan kebocoran data sebesar 720 GB berisi catatan medis pasien dari beberapa rumah sakit di Indonesia, dan data tersebut diduga dijual di forum online (Ardianto & Nurjanah, 2024). Insiden ini menunjukkan bahwa institusi kesehatan di Indonesia masih menghadapi tantangan besar dalam memastikan keamanan data pasien. Selain merusak reputasi, kebocoran data juga dapat mengurangi tingkat kepercayaan masyarakat terhadap layanan kesehatan. Oleh karena itu, sistem pengelolaan data yang dilengkapi teknologi keamanan seperti enkripsi sangat penting untuk meminimalkan risiko kebocoran data.

Kriptografi, khususnya algoritma Advanced Encryption Standard (AES), telah terbukti sebagai metode yang andal dalam menjaga kerahasiaan data.

AES adalah algoritma enkripsi blok simetris yang mengubah data asli (*plaintext*) menjadi data terenkripsi (*ciphertext*) menggunakan kunci enkripsi dengan panjang 128-bit, 192-bit, atau 256-bit (Khoirunnisa & Djuniadi, 2021). Penelitian oleh (Ramadhani et al., 2024) menunjukkan bahwa AES-128 berhasil dalam mengamankan data rekam medis dalam database. (Khoirunnisa & Djuniadi, 2021) menegaskan bahwa enkripsi AES menghasilkan data yang hanya dapat diakses melalui proses dekripsi, memastikan keamanan data. Selain itu, studi oleh (Alhamdi & Siahaan, 2021) membahas implementasi kriptografi menggunakan AES-128 pada platform Android yang berhasil menjaga keamanan pesan teks, terutama dalam mencegah akses tidak sah. Penelitian oleh (Muttaqin & Rahmadoni, 2020) menganalisis desain sistem keamanan file berbasis kriptografi AES, menunjukkan bahwa algoritma ini tidak hanya aman tetapi juga hemat waktu dalam proses enkripsi dan dekripsi data, menjadikannya cocok untuk diterapkan di berbagai platform digital. Dengan demikian, AES-128 menjadi pilihan yang tepat untuk memastikan perlindungan data yang aman di lingkungan kesehatan.

Puskesmas Cilebak, Kabupaten Kuningan, menghadapi tantangan dalam pengelolaan data rekam medis karena masih menggunakan sistem manual seperti data pasien dicatat di atas kertas dan direkap menggunakan aplikasi seperti Microsoft Word dan Excel, yang tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan dan kehilangan data. Sistem yang tidak terintegrasi ini menyulitkan proses pencarian dan pembaruan data. Saat ini, aplikasi PCare yang digunakan hanya mencakup pengelolaan administrasi pasien BPJS, sehingga data pasien umum belum tercatat dalam sistem yang aman. Aplikasi PCare adalah sistem informasi yang dikembangkan oleh Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan untuk mendukung pelayanan kesehatan bagi peserta BPJS. Aplikasi ini digunakan oleh fasilitas kesehatan, termasuk Puskesmas, untuk mengelola data administrasi pasien BPJS, seperti pendaftaran, pencatatan riwayat kunjungan, dan pelaporan layanan kesehatan. Dengan rata-rata kunjungan harian sebanyak 15 pasien,

baik peserta BPJS maupun pasien umum, digitalisasi sistem pengelolaan data menjadi kebutuhan untuk meningkatkan kelancaran layanan sekaligus menjaga kerahasiaan data pasien. Data rekam medis yang mencakup informasi sensitif pasien, seperti nomor NIK, nomor kartu BPJS dan riwayat rekam medis, saat ini belum terlindungi dengan baik dan berisiko diakses oleh pihak tidak berwenang.

Berdasarkan latar belakang di atas, penelitian ini bertujuan untuk merancang sebuah sistem pengelolaan data rekam medis internal berbasis web di Puskesmas Cilebak. Sistem ini dirancang dengan menerapkan aspek keamanan melalui algoritma *Advanced Encryption Standard (AES)* 128-bit guna memenuhi regulasi yang berlaku, termasuk Peraturan Menteri Kesehatan tentang Perlindungan Data Pasien. Tujuannya adalah untuk menjaga kerahasiaan informasi pasien, seperti nomor NIK, nomor kartu BPJS dan riwayat rekam medis. Oleh karena itu, penelitian ini mengusung judul: **"Implementasi Algoritma AES-128 untuk Pengamanan Data Rekam Medis (Studi Kasus: Puskesmas Cilebak)".**

1.2 Identifikasi Masalah

Berdasarkan pemaparan dari latar belakang di atas, dapat di identifikasi permasalahan pada penelitian yang akan dilakukan, yaitu sebagai berikut :

1. Sistem pengelolaan data rekam medis di Puskesmas Cilebak masih manual, menggunakan pencatatan di atas kertas dan perekapan dengan Microsoft Word dan Excel, sehingga tidak terstruktur, lambat, rentan kesalahan, dan menyulitkan pencarian serta pembaruan data.
2. Data rekam medis mengandung informasi pribadi dan sensitif. Seperti identitas pasien, riwayat penyakit dan hasil pemeriksaan, yang sangat rentan terhadap penyalahgunaan apabila tidak dilindungi dengan baik.

1.3 Rumusan Masalah

Adapun rumusan permasalahan yang akan dilakukan dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang sistem pengelolaan data rekam medis di Puskesmas Cilebak yang saat ini masih bersifat manual agar lebih terstruktur dan meminimalkan risiko kehilangan atau kesalahan data?
2. Bagaimana mengimplementasikan algoritma *Advanced Encryption Standard (AES)* 128-bit untuk meningkatkan keamanan data rekam medis di Puskesmas Cilebak guna melindungi informasi sensitif pasien dari risiko kebocoran atau akses tidak berwenang?

1.4 Batasan Masalah

Agar penelitian yang dilaksanakan terarah dan tersusun dengan baik. Adapun batasan masalah dalam penelitian ini, yaitu :

1. Penelitian mencakup perancangan dan penerapan sistem pengelolaan data rekam medis untuk seluruh pasien di Puskesmas Cilebak, baik pasien BPJS maupun umum.
2. Aplikasi yang dikembangkan adalah sistem client-server berbasis web yang digunakan oleh berbagai petugas di Puskesmas, termasuk resepsionis, petugas dokter/perawat, dan admin, di mana seluruh pengguna diwajibkan *login* untuk memastikan akses terbatas hanya kepada pihak berwenang. Setiap peran memiliki hak aksesnya masing-masing, yaitu:
    a. Petugas Resepsionis: Mengelola pendaftaran dan data pasien.
    b. Petugas Dokter/Perawat: Mengelola pelayanan dan data riwayat rekam medis pasien.
    c. Admin: Memiliki hak akses penuh untuk mengelola data pasien, pelayanan dan riwayat rekam medis pasien.
3. Sistem ini menggunakan enkripsi AES-128 pada data sensitif seperti nomor NIK, nomor kartu BPJS, riwayat rekam medis serta memastikan kerahasiaan dan keamanan data rekam medis pasien di setiap akses dan pengelolaan.

4. Sistem ini mengenkripsi data pasien dan riwayat rekam medis, data yang akan dienkripsi pada data pasien adalah NIK dan nomor kartu BPJS, contoh NIK: 3208277105020002 dan contoh nomor kartu BPJS: 0000342001986. Pada data riwayat rekam medis yaitu lama sakit, prognosa, TACC *(Time, Age, Comorbidity, Complication)*, kesadaran, terapi obat, terapi non obat, tinggi badan, berat badan, *sistole*, *diastole*, MAP (*Mean Arterial Pressure*), *heart rate*, suhu tubuh, *respiratory rate*, saturasi, lingkar perut, IMT (Indeks Masa Tubuh), hasil IMT, BMHP (Bahan Medis Habis Pakai), luas permukaan tubuh anak, edukasi, *triage*, status hamil, detak jantung, catatan, cara ukur tinggi badan, *inform consent*, diagnosa dan tindak lanjut. Sedangkan *cipherkey*-nya adalah: PUSKESMASCILEBAK.
5. Aplikasi dibuat menggunakan Bahasa Pemograman PHP dengan menggunakan framework Laravel, dengan database MySQL.
6. *Tools* yang digunakan dalam membangun aplikasi ini adalah XAMPP, Visual Studio Code dan Google Chrome.
7. Penelitian hanya berfokus pada pengembangan sistem pengelolaan data rekam medis yang aman menggunakan AES-128, tanpa mencakup pembuatan atau penyediaan modul pelatihan keamanan data bagi staf.

1.5 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang sistem pengelolaan data rekam medis yang aman dan terintegrasi untuk seluruh pasien di Puskesmas Cilebak, baik pasien BPJS maupun umum.
2. Menerapkan algoritma enkripsi AES-128 untuk memastikan kerahasiaan dan keamanan data rekam medis pasien.
3. Mengurangi risiko penyalahgunaan data rekam medis melalui penerapan metode enkripsi yang kuat dan sesuai dengan standar keamanan data.

1.6 Manfaat Penelitian

a. Manfaat Teoritis

1. Mengimplementasikan ilmu yang telah diperoleh selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Kuningan dengan membuat penelitian secara ilmiah dan sistematis.
2. Memperluas wawasan dan kemampuan dalam menerapkan teori mata kuliah ke dalam penelitian yang nyata.
3. Memperoleh pemahaman mendalam tentang perancangan dan implementasi sistem pengelolaan data rekam medis yang aman menggunakan enkripsi AES-128.

b. Manfaat Praktis

1) Bagi Pasien

1. Mendapatkan perlindungan privasi dan keamanan yang lebih baik terhadap informasi pribadi mereka dalam data rekam medis.
2. Merasakan keamanan bahwa data pribadi mereka dikelola secara baik dan terlindungi dari potensi kebocoran.
3. Meningkatkan kepercayaan terhadap layanan kesehatan di Puskesmas Cilebak, dengan keyakinan bahwa data mereka dirahasiakan dengan aman.

2) Bagi Puskesmas

1. Memiliki sistem pengelolaan data rekam medis yang aman, terintegrasi, dan mencakup seluruh pasien, baik BPJS maupun umum.
2. Mengurangi risiko kebocoran informasi pasien sekaligus menjaga kerahasiaan data rekam medis.
3. Meningkatkan kepercayaan masyarakat terhadap layanan kesehatan dengan adanya perlindungan data yang lebih baik.

1.7 Pertanyaan Penelitian

Adapun pertanyaan yang ditanyakan dalam penelitian ini yaitu sebagai berikut:

1. Apakah dapat dirancang sistem pengelolaan data rekam medis di Puskesmas Cilebak yang saat ini masih bersifat manual agar menjadi lebih terstruktur dan mampu meminimalkan risiko kehilangan atau kesalahan data?
2. Apakah algoritma *Advanced Encryption Standard (AES)* 128-bit dapat diimplementasikan untuk meningkatkan keamanan data rekam medis di Puskesmas Cilebak guna melindungi informasi sensitif pasien dari risiko penyalahgunaan atau akses yang tidak sah?

1.8 Hipotesis Penelitian

Berdasarkan pemaparan di atas, hipotesis yang diajukan adalah, **"Penerapan algoritma enkripsi AES-128 pada sistem pengelolaan data rekam medis di Puskesmas Cilebak diyakini dapat meningkatkan keamanan dan kerahasiaan data pasien, serta mengurangi risiko kebocoran informasi sensitif akibat akses tidak sah".**

1.9 Metodologi Penelitian

1.9.1 Metode Pengumpulan Data

Metode penelitian dan pengumpulan data yang peneliti terapkan adalah sebagai berikut.

1. Observasi

   Peneliti melakukan pengamatan langsung di Puskesmas Cilebak yang beralamat di Desa Cilebak, Kec. Cilebak, Kab. Kuningan. Observasi dilakukan untuk memahami kondisi aktual pengelolaan data rekam medis, alur kerja tenaga kesehatan, serta sistem yang telah digunakan. Data hasil observasi digunakan sebagai dasar untuk merancang sistem pengelolaan data rekam medis yang sesuai dengan kebutuhan Puskesmas.

2. Wawancara

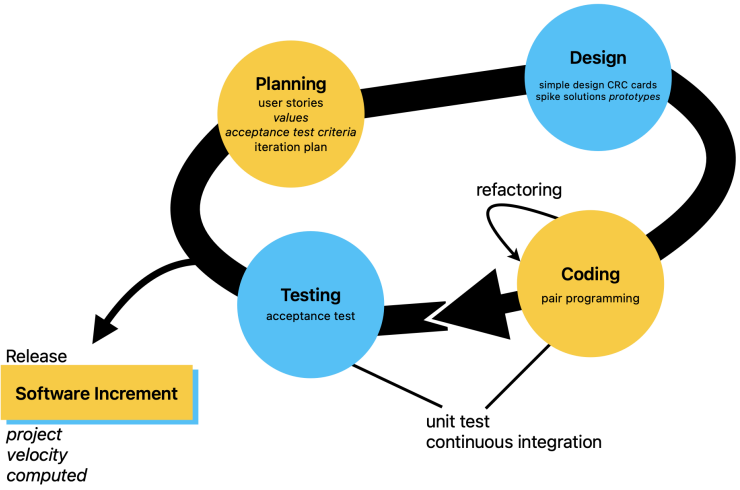
   Peneliti mengumpulkan data melalui metode wawancara dengan bidan Puskesmas, yaitu Ibu Laela Qodariah, yang memiliki pengetahuan tentang pengelolaan data rekam medis di Puskesmas Cilebak. Wawancara bertujuan untuk menggali informasi terkait permasalahan dalam pengelolaan data, kebutuhan keamanan data, serta harapan terhadap sistem yang akan dirancang.

3. Studi Pustaka

   Peneliti mengumpulkan data melalui studi pustaka dengan menelaah berbagai literatur yang relevan, seperti skripsi mahasiswa sebelumnya, dan jurnal ilmiah. Studi pustaka ini mencakup pembahasan mengenai algoritma enkripsi AES-128, aspek keamanan data dalam sektor kesehatan, teori sistem informasi, serta regulasi yang berkaitan dengan privasi dan perlindungan data kesehatan. Informasi yang diperoleh dari sumber-sumber tersebut digunakan untuk memperkuat landasan teori dan mendukung proses pengambilan keputusan dalam perancangan dan pengembangan sistem.

1.9.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah **Extreme Programming (XP)**. Extreme Programming (XP) merupakan salah satu metodologi dalam *Agile Software Development* yang berfokus pada pengkodean (*coding*) sebagai aktivitas utama dalam seluruh tahapan siklus pengembangan perangkat lunak (Ridwan Nawawi et al., 2022). XP menawarkan pendekatan pengembangan yang lebih adaptif dan fleksibel dengan menyederhanakan proses melalui iterasi singkat dan kolaborasi intensif antara pengembang dan pengguna (Icha Riani & Rayyan Firdaus, 2024). Metode ini juga memungkinkan pengembangan sistem secara bertahap untuk memenuhi kebutuhan yang berubah-ubah dengan tetap menjaga kualitas perangkat lunak.



Gambar 1. 1 Tahapan dalam proses XP

Berdasakan gambar diatas, Terdapat empat tahapan yang harus dikerjakan pada metode Extreme Programming (XP) menurut (Ridwan Nawawi et al., 2022) yaitu :

1. *Planning* (Perencanaan).

   Tahap ini mencakup identifikasi kebutuhan, analisis prioritas, dan penjadwalan pengembangan. Peneliti mengumpulkan kebutuhan sistem melalui observasi dan wawancara dengan staf Puskesmas Cilebak, Ibu Laela Qodariah. *User stories* digunakan untuk menggambarkan fitur dan fungsi sistem serta menjadi acuan dalam pengembangan. *Iteration plan* disusun untuk menentukan urutan pengerjaan *user stories*, sedangkan *velocity* digunakan untuk mengukur jumlah *user stories* yang dapat diselesaikan dalam satu iterasi dan menentukan durasi iterasi.

2. *Design* (Perancangan).

   Tahap ini mencakup perancangan sistem secara rinci, termasuk pemodelan sistem, arsitektur, database, dan antarmuka pengguna (UI) menggunakan UML. *CRC Card* digunakan untuk memetakan tanggung jawab antar kelas berdasarkan *class diagram*. Selain itu, *Spike Solution* dilakukan dengan membuat *prototype* aplikasi sebagai gambaran awal desain dan fungsi

sistem, guna mengurangi risiko teknis dan memastikan kesesuaiannya dengan kebutuhan pengguna.

3. *Coding* (Pengkodean).

   Tahap pengkodean merupakan proses mengubah rancangan sistem menjadi kode program menggunakan bahasa pemrograman yang sesuai. Peneliti mulai mengimplementasikan fitur sesuai kebutuhan, didukung metode *Pair Programming* untuk menjaga kualitas kode melalui kolaborasi dan deteksi dini kesalahan. *Refactoring* dilakukan dengan menyederhanakan struktur kode tanpa mengubah fungsinya guna meningkatkan keterbacaan dan kemudahan pemeliharaan.

4. *Testing* (Pengujian).

   Setelah pengkodean selesai, dilakukan pengujian sistem untuk mendeteksi kesalahan (bugs) dan memastikan sistem berjalan sesuai dengan kebutuhan pengguna. Peneliti menggunakan metode *blackbox testing* untuk menguji fungsi sistem secara keseluruhan dan *whitebox testing* untuk memeriksa logika pemrograman.

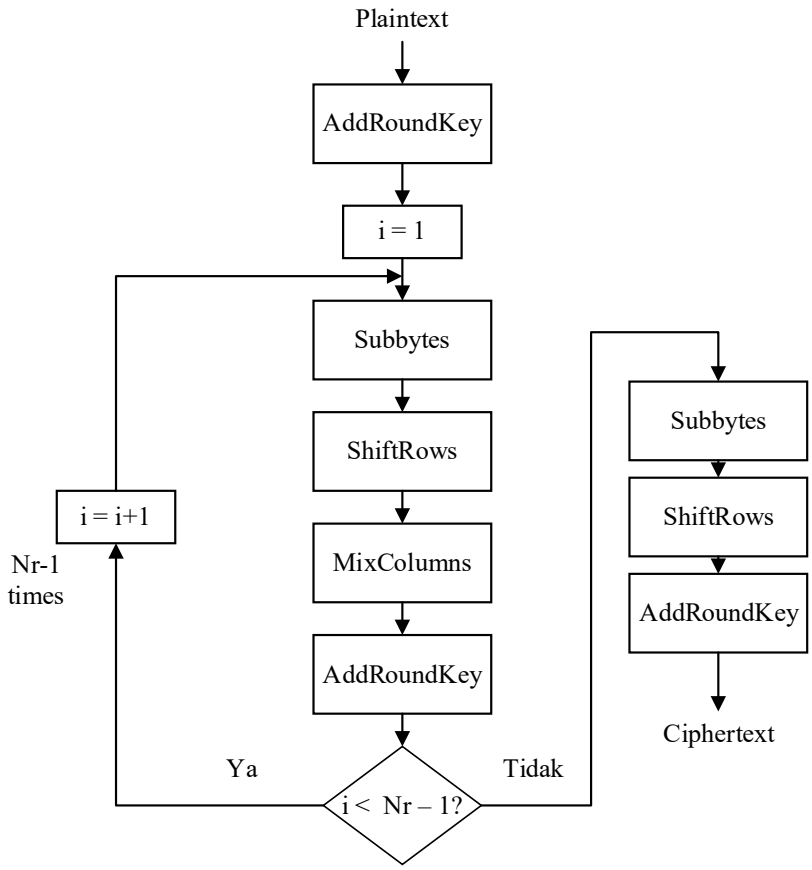
1.9.3 Metode Penyelesaian Masalah

Berdasarkan tujuan utama dari penelitian ini, peneliti merancang dan mengimplementasikan sistem pengelolaan data rekam medis di Puskesmas Cilebak menggunakan algoritma *Advanced Encryption Standard* (AES) dengan panjang kunci 128-bit (AES-128) untuk menjaga keamanan data sensitif pasien. Algoritma AES adalah metode enkripsi blok simetris yang diusulkan oleh *National Institute of Standards and Technology* (NIST) pada tahun 2000 untuk menggantikan algoritma *Data Encryption Standard* (DES) dan *Triple Data Encryption Standard* (DES). AES menggunakan panjang kunci yang bervariasi, yaitu 128-bit, 192-bit, dan 256-bit, dengan jumlah putaran enkripsi masing-masing 10, 12, dan 14 (Chai et al., 2023).

Algoritma AES-128 dipilih dalam penelitian ini karena memberikan keseimbangan antara keamanan dan efisiensi dalam hal proses enkripsi dan dekripsi, tidak terlalu membebani sistem dan mudah diimplementasikan.

AES-128 memiliki tingkat keamanan yang sangat tinggi terhadap serangan *brute-force*, yaitu metode peretasan dengan mencoba semua kemungkinan kunci secara bergantian. Hal ini disebabkan oleh ukuran kunci sepanjang 128-bit, yang menghasilkan total 2^{128} (340.282.366.920.938.463.463.374.607.431.768.211.456) kemungkinan kombinasi kunci yang unik. Dengan demikian, AES-128 secara praktis tidak mungkin diretas melalui serangan *brute-force* dengan teknologi komputasi saat ini, menjadikannya pilihan yang sangat aman untuk perlindungan data.

Selain itu, algoritma ini menjalankan 10 putaran enkripsi, yang merupakan jumlah optimal untuk memastikan keamanan tanpa membuat proses enkripsi menjadi terlalu lambat. Setiap putaran terdiri dari operasi substitusi, permutasi, dan pencampuran data, sehingga semakin memperkuat hasil enkripsi. Dibandingkan varian AES dengan kunci lebih panjang (seperti AES-192 atau AES-256), AES-128 lebih ringan dalam penggunaan sumber daya, sehingga cocok untuk aplikasi yang membutuhkan kecepatan dan keamanan secara bersamaan. Dengan demikian, AES-128 dengan 10 putaran menjadi pilihan yang tepat untuk penelitian ini. Proses enkripsi dan dekripsi pada algoritma ini adalah sebagai berikut :



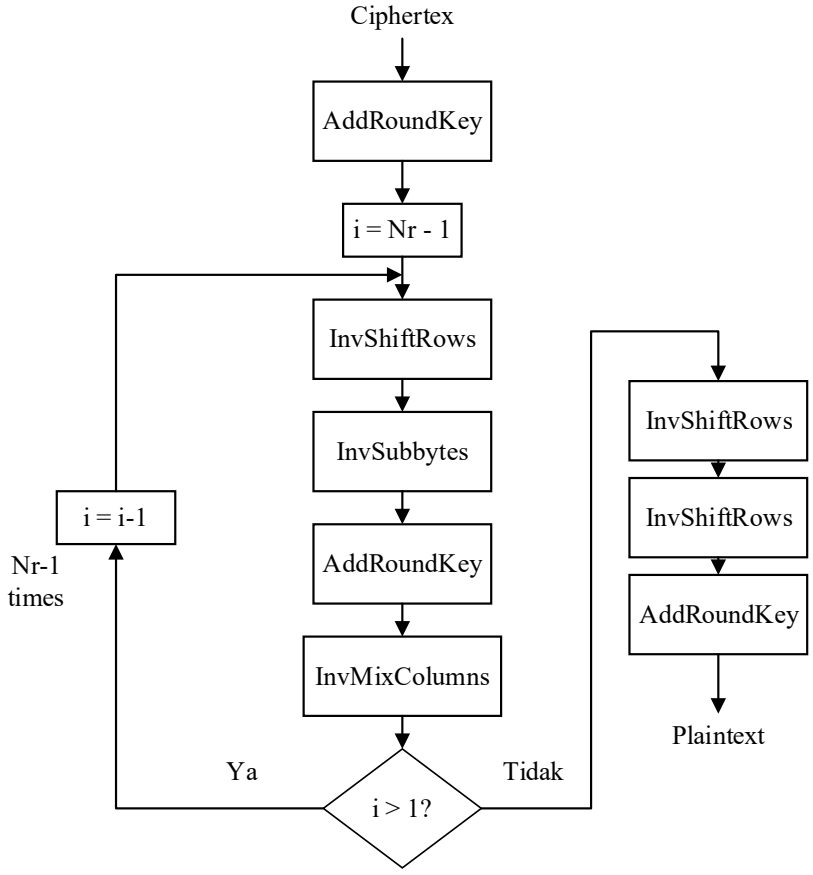
Gambar 1. 2 Enkripsi AES (Chai et al., 2023)

Seperti yang ditunjukkan pada gambar di atas, kunci 128-bit dipilih untuk mengenkripsi pesan, oleh karena itu dilakukan 10 kali iterasi. Secara umum proses enkripsi pada AES terbagi menjadi dua yaitu proses enkripsi itu sendiri (Proses Enkripsi) dan pembangkitan kunci (Ekspansi kunci) atau *round key*. Berikut adalah langkah-langkah utama dalam enkripsi AES-128:

1. *AddRoundKey* : XOR antara *plaintext* awal dengan *cipherkey*.
2. Bulatkan sebanyak Nr-1 kali (sembilan putaran pada AES128) Proses setiap putaran adalah :
   a. *SubBytes* : Melakukan substitusi *byte* menggunakan tabel substitusi (*S-box*).
   b. *ShiftRows* : Menggeser *byte* di setiap baris kecuali baris pertama (baris kedua digeser satu posisi, baris ketiga dua posisi, dan seterusnya).

c. *MixColumns* : Mengalikan setiap kolom data dengan matriks tertentu untuk menghasilkan kolom baru.
d. *AddRoundKey* : XOR status terkini dengan kunci pembulatan.

3. Jika proses sudah mencapai Nr, selanjutnya Babak Final. Proses untuk babak final :
   a. Subbytes
   b. ShiftRows
   c. AddRoundKey
4. Akhirnya, *ciphertext* (data terenkripsi) dengan karakter acak dihasilkan setelah menyelesaikan semua putaran.



Gambar 1. 3 Dekripsi AES (Chai et al., 2023)

Dekripsi AES adalah kebalikan dari proses enkripsi, di mana setiap langkah pada proses enkripsi digantikan oleh operasi *invers*nya.

Sebagaimana ditunjukkan pada gambar diatas, proses dekripsi terdiri dari:

1. *AddRoundKey* : XOR antara ciphertext dengan kunci.
2. Bulatkan sebanyak Nr-1 kali (sembilan putaran pada AES128) Proses setiap putaran adalah :
   a. InvShiftRows : Menggeser byte ke posisi semula (kebalikan ShiftRows).
   b. InvSubBytes : Menggunakan tabel substitusi invers (invers S-box).
   c. AddRoundKey : XOR status dengan kunci.
   d. InvMixColumns : Mengalikan kolom data dengan matriks invers tertentu.
3. Jika proses sudah mencapai Nr, selanjutnya Babak Final. Proses untuk babak final :
   a. InvShiftRows
   b. InvSubBytes
   c. AddRoundKey
4. Akhirnya, plaintext (data asli) berhasil dipulihkan setelah menyelesaikan semua putaran dekripsi menggunakan operasi invers dari proses enkripsi.

Pemilihan Algortima AES-128 didasarkan pada efisiensinya untuk sistem yang memerlukan performa cepat tanpa mengorbankan keamanan. Dalam aplikasi ini, algoritma AES-128 digunakan untuk mengenkripsi data sensitif pasien, seperti identitas pasien, dan riwayat rekam medis, memastikan kerahasiaan data tetap terjaga meskipun terjadi akses yang tidak sah.

Berikut ini contoh perhitungan algoritma AES-128 :

a. Proses Enkripsi

   Plaintext : 3208277105020002

   Cipherkey : PUSKESMASCILEBAK

1. Konversikan *plaintext* dan *cipherkey* kedalam bentuk heksadesimal.

3	2	0	0
2	7	5	0
0	7	0	0
8	1	2	2

→

33	32	30	30
32	37	35	30
30	37	30	30
38	31	32	32

P	E	S	E
U	S	C	B
S	M	I	A
K	A	L	K

→

50	45	53	45
55	53	43	42
53	4D	49	41
4B	41	4C	4B

2. Lakukan *addroundkey*

Plaintext

33	32	30	30
32	37	35	30
30	37	30	30
38	31	32	32

⊕

Cipherkey

50	45	53	45
55	53	43	42
53	4D	49	41
4B	41	4C	4B

Untuk melakukan *addroundkey*, bilangan yang berbentuk heksadesimal diubah terlebih dahulu ke bentuk heksadesimal. Setelah diubah ke bentuk binner dilakukan proses XOR. Hasil dari proses XOR kemudian diubah kembali ke dalam bilangan heksadesimal.

Contoh perhitungan XOR:

(33)	0011	0011		(32)	0011	0010		
(50)	0101	0000	⊕	(55)	0101	0101	⊕	
	0110	0011	(63)		0110	0111	(67)	

(30)	0011	0000		(38)	0011	1000		
(53)	0101	0011	⊕	(4B)	0100	1011	⊕	
	0110	0011	(63)		0111	0011	(73)	

3. Dari hasil XOR diatas, maka diperoleh hasil sebagai berikut:

Plaintext

33	32	30	30
32	37	35	30
30	37	30	30
38	31	32	32

⊕

Cipherkey

50	45	53	45
55	53	43	42
53	4D	49	41
4B	41	4C	4B

Hasil Xor

63	77	63	75
67	64	76	72
63	7A	79	71
73	70	7E	79

4. Kemudian dilakukan *subbyte.* Yaitu mentrasformasi hasil dari XOR dengan tabel *S-Box*.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Sehingga didapatkan hasil sebagai berikut :

Hasil Xor

63	77	63	75
67	64	76	72
63	7A	79	71
73	70	7E	79

→

Hasil *Subbytes*

FB	F5	FB	9D
85	43	38	40
FB	DA	B6	A3
8F	51	F3	B6

5. Lakukan proses transformasi *shiftrows,* yaitu dengan menggeser tempat *byte* dari kiri ke kanan kecuali baris pertama.

Hasil *Subbytes*

FB	F5	FB	9D
85	43	38	40
FB	DA	B6	A3
8F	51	F3	B6

→

Hasil *Shiftrows*

FB	F5	FB	9D
43	38	40	85
B6	A3	FB	DA
B6	8F	51	F3

6. Proses *Mixcolumnns*

Hasil *Shiftrows*

FB	F5	FB	9D
43	38	40	85
B6	A3	FB	DA
B6	8F	51	F3

X

Matriks AES

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Proses *MixColumns* pada kolom pertama dimulai dengan mengambil elemen hasil dari proses *ShiftRows*, yaitu FB, 43, B6, dan B6, yang masing-masing berada pada baris ke-0 hingga ke-3. Untuk memperoleh nilai pada baris pertama kolom pertama dari hasil *MixColumns*, digunakan baris pertama matriks AES, yaitu [02 03 01 01]. Maka dilakukan perhitungan sebagai berikut:

$$(02 \times FB) \oplus (03 \times 43) \oplus (01 \times B6) \oplus (01 \times B6)$$

Langkah pertama adalah menghitung (02 × FB). Nilai FB dalam biner adalah 11111011. Operasi 02 × FB dilakukan dengan cara shift kiri satu bit: 1111 1011 << 1 = 1 1110 110 (1F6). Karena hasil ini melebihi 8 bit, maka dilakukan reduksi dengan polinomial irreducible AES (0x1B), maka angka 1 yang didepan tidak dibawa, yang hasilnya 1F6 maka hitungnya langsung ke F6, seperti dibawah ini:

(F6) 1111 0110
(1B) 0001 1011 ⊕
1110 1101 (ED)

Langkah kedua, (03 × 43) dihitung sebagai (02 × 43) ⊕ 43. Nilai 43 dalam biner adalah 01000011. Shift kiri satu bit: 0100 0011 << 1 = 1000 0110 (86) . Kemudian:

(86) 1000 0110
(43) 0100 0100 ⊕
1100 0101 (C5)

Langkah ketiga dan keempat, (01 × B6) dan (01 × B6), karena dikalikan dengan 01 maka nilainya tetap, dan hasil XOR-nya:

$$B6 \oplus B6 = 00$$

Akhirnya, hasil akhir didapatkan dengan:

$$ED \oplus C5 \oplus 00 = 28$$

Sehingga, nilai pada baris pertama kolom pertama hasil *MixColumns* adalah 28 dalam bentuk heksadesimal. Proses ini terus menerus hingga semua kolom terisi.

Hasil *mixcolumns*

28	95	87	9C
0A	F4	3C	0A
0E	1A	A5	B9
94	9A	0F	1E

7. Proses *Addroundkey*

Hasil *mixcolumns* ⊕ Sub kunci ke-1

28	95	87	9C
0A	F4	3C	0A
0E	1A	A5	B9
94	9A	0F	1E

⊕

7D	38	6B	2E
D6	85	C6	84
E0	AD	E4	A5
25	64	28	63

Hasil *Addroundkey*

55	AD	EC	B2
DC	71	FA	8E
EE	B7	41	1C
B1	FE	27	7D

Hasil dari *mixcolumns* kemudian dilakukan proses XOR dengan sub kunci-1. Proses enkripsi dilakukan dari *round* 0 sampai *round* 10. Namun pada *round* 10 tidak dilakukan proses *mixcolumns*, langsung dilakukan proses XOR antara hasil *shiftrows* dengan sub kunci ke-10, maka hasilnya itu menjadi *ciphertext.* Adapun hasil dari proses perhitungan enkripsi algoritma AES-128 dapat dilihat pada Tabel 1.1 :

Tabel 1. 1. Proses Perhitungan Enkripsi Algoritma AES-128

	Subbytes				**Shiftrows**				**Mixcolumns**				**Addroundkey**				**Subbkunci**			
Round 1	FB	F5	FB	9D	FB	F5	FB	9D	28	95	87	9C	63	77	63	75	7D	38	6B	2E
	85	43	38	40	43	38	40	85	0A	F4	3C	0A	67	64	76	72	D6	85	C6	84
	FB	DA	B6	A3	B6	A3	FB	DA	0E	1A	A5	B9	63	7A	79	71	E0	AD	E4	A5
	8F	51	F3	B6	B6	8F	51	F3	94	9A	0F	1E	73	70	7E	79	25	64	28	63
Round 2	FC	95	CE	37	FC	95	CE	37	61	12	3F	9A	55	AD	EC	B2	20	18	73	5D
	86	A3	2D	19	A3	2D	19	86	C0	B8	3F	0C	DC	71	FA	8E	D0	55	93	17
	28	A9	83	9C	83	9C	28	A9	58	D8	51	B7	EE	B7	41	1C	1B	B6	52	F7
	C8	BB	CC	FF	FF	C8	BB	CC	DA	9E	15	F5	B1	FE	27	7D	14	70	58	3B
Round 3	83	67	29	C6	83	67	29	C6	12	E4	8A	AE	41	0A	4C	C7	D4	CC	BF	E2
	CA	55	91	AF	55	91	AF	CA	2F	CE	6A	10	10	ED	AC	1B	B8	ED	7E	69
	1A	9F	7B	09	7B	09	1A	9F	A6	62	CA	17	43	6E	03	40	F9	4F	ID	EA
	8B	28	E3	8B	8B	8B	28	E3	BD	3C	9E	D9	CE	EE	4D	CE	58	28	70	4B
Round 4	B4	34	96	29	B4	34	96	29	58	F0	C3	21	C6	28	35	4C	25	E9	56	B4
	88	26	FA	B6	26	FA	B6	88	A5	FE	51	79	97	23	14	79	3F	D2	AC	C5
	CF	D8	0E	54	0E	54	CF	D8	5F	16	B0	72	5F	2D	D7	FD	4A	05	18	F2
	D9	FA	28	4F	4F	D9	FA	28	71	5B	37	7B	E5	14	EE	92	C0	E8	98	D3
Round 5	FF	D4	2A	2A	FF	D4	2A	2A	76	4A	CF	83	7D	19	95	95	93	7A	2C	98
	B8	71	54	65	71	54	65	B8	82	F8	66	BF	9A	2C	FD	BC	B6	64	C8	0D
	59	7D	C2	CD	C2	CD	59	7D	4C	42	4A	E3	15	13	A8	80	2C	29	31	C3
	C8	6D	79	C2	C2	C8	6D	79	36	75	98	49	B1	B3	AF	A8	4D	A5	3D	EE
Round 6	D9	04	11	AF	D9	04	11	AF	AD	A9	DB	14	E5	30	E3	1B	64	1E	32	AA
	18	DE	E4	37	DE	E4	37	18	41	34	64	18	34	9C	AE	B2	98	FC	34	39

	D0	7F	21	B7	21	B7	D0	7F	A1	F6	0D	43	60	6B	7B	20	04	2D	1C	DF
	21	70	06	5C	5C	21	70	06	37	1D	34	81	7B	D0	A5	A7	0B	AE	93	7D
Round 7	DD	A9	1E	AE	DD	A9	1E	AE	B0	89	4B	FD	C9	B7	E9	BE	36	28	1A	B0
	E5	E8	53	FD	E8	53	FD	35	3B	9D	98	48	D9	C8	50	21	06	FA	CE	F7
	06	B9	82	DE	82	DE	06	B9	E1	7B	58	16	A5	DB	11	9C	FB	D6	CA	15
	EB	6D	5C	B0	B0	EB	6D	5C	6D	A0	03	DD	3C	B3	A7	FC	A7	09	9A	E7
Round 8	44	32	D1	E3	44	32	D1	E3	D3	A3	D0	CF	86	A1	51	4D	DE	F6	EC	5C
	27	85	B1	08	85	B1	08	27	04	B2	EF	E7	3D	67	56	BF	5F	A5	6B	9C
	A2	95	4F	7B	4F	7B	A2	95	C4	E9	E8	DC	1A	AD	92	03	6F	B9	73	66
	74	D3	EE	80	80	74	D3	EE	1D	74	7F	4B	CA	A9	99	3A	40	49	D3	34
Round 9	D7	FC	EB	DC	D7	FC	EB	DC	78	BA	EB	2A	0D	55	3C	93	1B	ED	01	5D
	39	F0	5F	21	F0	5F	21	39	C2	09	28	CA	5B	17	84	7B	6C	C9	A2	3E
	62	53	14	F4	14	F4	62	53	62	84	67	EB	AB	50	9B	BA	77	CE	BD	DB
	4C	27	91	D2	D2	4C	27	91	39	2C	2B	2C	5D	3D	AC	7F	02	43	90	A4
Round 10	FB	5B	87	F5	FB	5B	87	F5					63	57	EA	77	9F	72	73	2E
	E4	BA	7E	BF	BA	7E	BF	E4					AE	C0	8A	F4	D5	1C	BE	80
	59	D6	57	04	57	04	59	D6					15	4A	DA	30	3E	F0	4D	96
	C3	A8	EA	C4	C4	C3	A8	EA					33	6F	BB	88	46	05	95	31
Encoded													**64**	**29**	**F4**	**DB**				
													6F	**62**	**01**	**64**				
													69	**F4**	**14**	**40**				
													82	**C6**	**3D**	**DB**				

Dari tabel diatas dapat dilihat hasil perhitungan enkripsi algoritma AES-128 menghasilkan *ciphertext* yaitu :

64	**29**	**F4**	**DB**
6F	**62**	**01**	**64**
69	**F4**	**14**	**40**
82	**C6**	**3D**	**DB**

b. Proses Dekripsi

1. Lakukan proses *addroundkey*

Untuk melakukan proses *addroundkey*, lakukan proses XOR antara *ciphertext* dengan sub kunci ke-10. Dari hasil enkripsi diatas itu mendapatkan hasil *ciphertext* dan sub kunci ke-10 sebagai berikut :

Ciphertext			
64	29	F4	DB
6F	62	01	64
69	F4	14	40
82	C6	3D	DB

$\oplus$

Sub Kunci ke-10			
9F	72	73	2E
D5	1C	BE	80
3E	F0	4D	96
46	05	95	31

(64) 0110 0100 (6F) 0110 1111

(9F) 1001 1111 $\oplus$ (D5) 1101 0101 $\oplus$

1111 1011 (FB) 1011 1010 (BA)

(69) 0110 1001 (82) 1000 0010

(3E) 0011 1110 $\oplus$ (46) 0100 0110 $\oplus$

0101 0111 (57) 1100 0100 (C4)

Hasil *addroundkey*

FB	5B	87	F5
BA	7E	BF	E4
57	04	59	D6
C4	C3	A8	EA

2. Transformasi *Invshiftrows*

   Pada proses ini dilakukan pergeseran *byte* dari kanan ke kiri. Sehingga didapatkan hasil sebagai berikut :

Hasil *addroundkey*

FB	5B	87	F5
BA	7E	BF	E4
57	04	59	D6
C4	C3	A8	EA

→

Hasil *Invshiftrows*

FB	5B	87	F5
E4	BA	7E	BF
59	D6	57	04
C3	A8	EA	C4

3. Proses *Invsubbytes*

   Pada proses ini dilakukan transformasi *invsubbytes* dengan tabel *inverse* S-Box.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Hasil *Invshiftrows*

FB	5B	87	F5
E4	BA	7E	BF
59	D6	57	04
C3	A8	EA	C4

→

Hasil *InvSubbytes*

63	57	EA	77
AE	C0	8A	F4
15	4A	DA	30
33	6F	BB	88

4. Proses *Addroundkey*

   Pada proses ini dilakukan proses XOR pada hasil dari *invsubbytes* dengan sub kunci ke-9.

Hasil *Invsubbytes* Sub kunci ke-9

63	57	EA	77
AE	C0	8A	F4
15	4A	DA	30
33	6F	BB	88

$\oplus$

1B	ED	01	5D
6C	C9	A2	3E
77	CE	BD	DB
0A	43	90	A4

Hasil *addroundkey*

78	BA	EB	2A
C2	09	28	CA
62	84	67	EB
39	2C	2B	2C

5. Proses *Invmixsolumns*

   Pada proses ini dilakukan perkalian hasil *addroundkey* dengan matriks AES.

Hasil *addroundkey* Matriks AES

78	BA	EB	2A
C2	09	28	CA
62	84	67	EB
39	2C	2B	2C

X

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

Hasil *Invmixcolumns*

D7	FC	EB	DC
F0	5F	21	39
14	F4	62	53
D2	4C	27	91

6. Proses dekripsi dilakukan dari round 10 hingga *round* 0, namun pada *round* 0 tidak dilakukan *invmixcolumns*, maka hasilnya akan menjadi *plaintext* atau teks asli.

1.10 Jadwal Penelitian

Adapun jadwal dari penelitian yang dilakukan adalah sebagai berikut:

Tabel 1. 2. Jadwal Kegiatan Penelitian

Kegiatan	Bulan																			
	November				Desember				Januari				Februari				Maret			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Planning (Perencanaan)																				
Design (Perancangan)																				
Coding (Pengkodean)																				
Testing (Pengujian)																				

Keterangan :

1. Pada tahap perencanaan yang berlangsung pada bulan November-Desember, peneliti memulai kegiatan dengan melakukan observasi dan wawancara langsung dengan staf Puskesmas, yaitu Ibu Laela Qodariah. Kegiatan ini bertujuan untuk mengumpulkan data awal yang dibutuhkan dalam penelitian. Melalui wawancara tersebut, peneliti memperoleh informasi pengelolaan data rekam medis, alur kerja tenaga kesehatan, serta sistem yang telah digunakan. Data hasil observasi digunakan sebagai dasar untuk merancang sistem pengelolaan data rekam medis yang sesuai dengan kebutuhan Puskesmas. Peneliti juga melakukan studi pustaka untuk meninjau literatur yang relevan, termasuk studi terkait algoritma

Advanced Encryption Standard (AES), yang akan digunakan dalam aplikasi ini untuk pengamanan data yang mendukung implementasi aplikasi.

2. Tahap desain dimulai pada bulan Desember hingga januari, di mana peneliti membuat perancangan sistem secara detail. Hal ini mencakup pembuatan diagram *UML* untuk menjelaskan alur kerja aplikasi, desain antarmuka pengguna (*UI*) untuk memastikan aplikasi mudah digunakan, dan perancangan database untuk pengelolaan data pasien dan rekam medis dengan aman.
3. Tahap pengkodean dilaksanakan pada bulan Januari hingga Februari. Pada tahap ini, peneliti mengimplementasikan desain yang telah dibuat sebelumnya menjadi aplikasi yang dapat digunakan. Proses *coding* menggunakan bahasa pemrograman PHP dengan *framework* Laravel, serta penerapan algoritma AES-128 untuk enkripsi dan dekripsi data. Semua fitur yang dirancang sebelumnya mulai diwujudkan secara bertahap dengan fokus pada pengujian di setiap iterasi untuk memastikan aplikasi berjalan sesuai rencana.
4. Tahap terakhir adalah pengujian, yang dilakukan pada bulan Januari hingga Maret. Peneliti menerapkan metode pengujian *Black-box* untuk memeriksa fungsionalitas aplikasi dan *White-box* untuk memverifikasi logika kode. Pengujian ini dilakukan agar aplikasi berfungsi dengan benar dan menghasilkan output yang diharapkan.

1.11 Sistematika Penelitian

Adapun gambaran secara umum mengenai sistematika penulisan skripsi ini, ialah sebagai berikut

BAB I : PENDAHULUAN

Pada bab ini menyajikan pembahasan mengenai latar belakang masalah, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, hipotesis, metodologi penelitian, dan sistematika penelitian

BAB II : LANDASAN TEORITIS

Pada bab ini memaparkan teori-teori yang berkaitan dengan permasalahan yang diteliti, hasil penelitian terdahulu, serta penjelasan mengenai kerangka teoritis.

BAB III : ANALISA DAN PERANCANGAN

Bab ini membahas analisis permasalahan yang sedang berjalan, analisis sistem, perancangan sistem, serta perancangan antarmuka.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas proses implementasi dan pengujian sistem yang telah dibangun. Pengujian dilakukan menggunakan metode *Black Box Testing*, *White Box Testing*.

BAB V : KESIMPULAN DAN SARAN

Bab ini menjadi bagian penutup dari penulisan skripsi yang mencakup kesimpulan serta saran untuk perbaikan di masa mendatang.