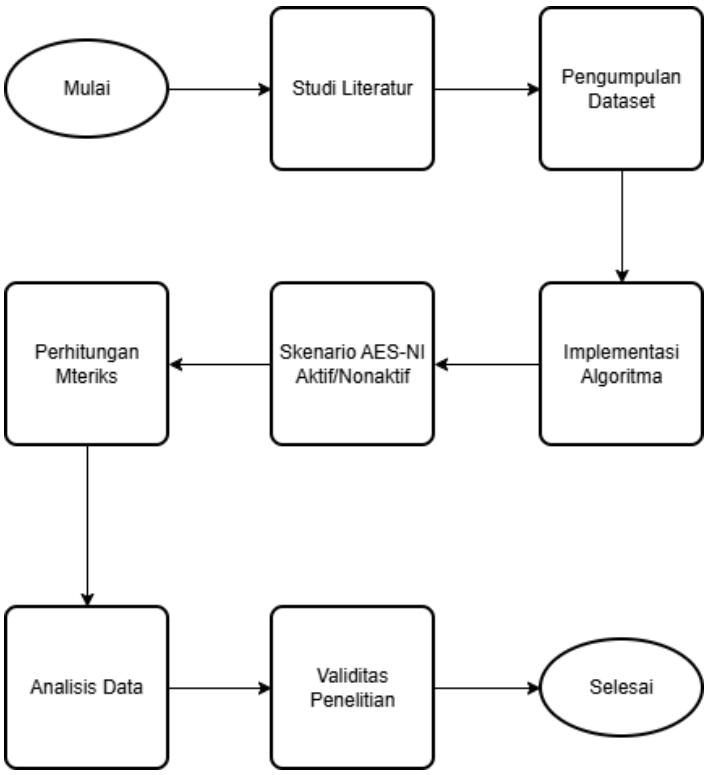
BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif yang bertujuan untuk menganalisis performa algoritma AES-GCM dan ChaCha20-Poly1305 dalam proses pengamanan *file* PDF. Pendekatan eksperimen dipilih karena penelitian dilakukan melalui pengujian langsung pada lingkungan yang terkontrol sehingga menghasilkan data numerik yang objektif dan terukur. Secara umum, tahapan penelitian disusun secara sistematis yang meliputi studi literatur, pengumpulan *dataset*, perancangan dan implementasi algoritma, pelaksanaan pengujian, analisis data, serta penarikan kesimpulan. Penelitian ini menggunakan pendekatan eksperimen yang mengikuti alur tahapan seperti ditunjukkan pada Gambar 3.1



Gambar 3. 1 Tahapan Penelitian

Gambar 3.1 menjelaskan pada tahap awal penelitian dimulai dengan studi literatur untuk memperoleh landasan teoritis mengenai kriptografi modern, khususnya algoritma *authenticated encryption* seperti AES-GCM dan ChaCha20-Poly1305. Pada tahap ini juga dilakukan identifikasi metode pengukuran performa yang relevan serta penelaahan terhadap penelitian terdahulu guna menemukan celah penelitian yang masih dapat dikembangkan. Hasil dari studi literatur digunakan sebagai dasar dalam merancang skenario eksperimen dan menentukan variabel penelitian. Selanjutnya, dilakukan tahap pengumpulan *dataset* yang berupa *file* dokumen berformat PDF dengan variasi ukuran tertentu. *Dataset* ini dipersiapkan sedemikian rupa agar representatif terhadap penggunaan

dokumen digital dalam kondisi nyata. Selanjutnya, kedua algoritma diimplementasikan menggunakan *library* kriptografi standar pada lingkungan perangkat keras dan perangkat lunak yang sama guna menjaga konsistensi hasil pengujian yaitu pada bahasa pemrograman *Python*. Setelah itu, tahap berikutnya adalah pelaksanaan eksperimen, yaitu melakukan proses enkripsi dan dekripsi terhadap *file* PDF yang telah disiapkan. Selama proses tersebut dilakukan pengukuran terhadap beberapa metrik performa yang meliputi waktu enkripsi, waktu dekripsi, *throughput*, *CPU Usage*, dan *Memory Usage*. Pengujian dilakukan secara berulang untuk memperoleh nilai rata-rata yang lebih stabil dan mengurangi kemungkinan bias pengukuran.

Kemudian data hasil pengujian kemudian dianalisis menggunakan pendekatan komparatif kuantitatif untuk mengetahui perbedaan performa kedua algoritma. Hasil analisis digunakan sebagai dasar dalam menarik kesimpulan mengenai algoritma yang lebih efisien dalam konteks pengamanan *file* PDF berukuran besar. Seluruh tahapan penelitian diakhiri dengan penyusunan laporan yang sistematis sesuai dengan kaidah penulisan ilmiah.

3.2 Studi Literatur

Tahap studi literatur dilakukan dengan mengkaji berbagai sumber ilmiah seperti jurnal, prosiding, dan buku yang relevan dengan algoritma AES-GCM dan ChaCha20-Poly1305 serta konsep *Authenticated Encryption with Associated Data (AEAD)*. Selain itu, dikaji pula metode

pengukuran performa sistem yang meliputi waktu eksekusi, *throughput*, serta penggunaan *resource* seperti *CPU Usage* dan *Memory Usage*. Hasil studi literatur digunakan sebagai dasar dalam menentukan parameter eksperimen, metode pengukuran, serta pendekatan analisis yang digunakan dalam penelitian ini.

Pada tahap ini dilakukan kajian terhadap artikel ilmiah yang membahas performa algoritma AES-GCM dan ChaCha20-Poly1305. Pada tahap ini dilakukan penelusuran terhadap jurnal internasional maupun nasional, buku referensi, standar kriptografi, serta dokumentasi teknis yang berkaitan dengan AES-GCM dan ChaCha20-Poly1305. Tujuan utama dari studi literatur adalah untuk memperoleh pemahaman mendalam mengenai karakteristik, kelebihan, serta keterbatasan masing-masing algoritma. Pada tahap studi literatur juga mencari terkait pengaruh AES-NI terhadap algoritma enkripsi autentikasi.

Selain itu, studi literatur juga mengkaji dan mengidentifikasi parameter performa yang umum digunakan dalam penelitian kriptografi, seperti waktu eksekusi, *throughput*, serta penggunaan *resource*. Melalui kajian terhadap penelitian terdahulu, ditemukan bahwa sebagian studi masih berfokus pada metrik berbasis waktu dan belum banyak yang mengevaluasi penggunaan *resource* secara komprehensif pada skenario enkripsi *file* PDF berukuran besar. Temuan ini kemudian menjadi dasar dalam merumuskan kontribusi penelitian yang dilakukan.

3.3 Pengumpulan *Dataset*

Dataset yang digunakan dalam penelitian ini berupa *file* PDF berbasis teks yang diperoleh dari sumber publik. *file* yang diperoleh kemudian diproses untuk menghasilkan variasi ukuran *file* sebesar 10 MB, 20 MB, 30 MB, 40 MB, 50 MB, 60 MB, 70 MB, 80 MB, 90 MB, dan 100 MB. Proses penyesuaian ukuran *file* dilakukan dengan menambahkan konten teks secara terstruktur tanpa mengubah format dasar *file* PDF, sehingga integritas dan validitas *file* tetap terjaga. Setiap *file* diverifikasi untuk memastikan dapat dibuka dan diproses dengan normal sebelum digunakan dalam pengujian.

File PDF yang digunakan disiapkan dalam beberapa variasi ukuran untuk mensimulasikan beban data yang berbeda. Variasi ukuran *file* yang digunakan dalam penelitian ini meliputi 10 MB, 20 MB, 30 MB, 40 MB, 50 MB, 60 MB, 70 MB, 80 MB, 90 MB, dan 100 MB. Setiap *file* diverifikasi terlebih dahulu untuk memastikan tidak mengalami kerusakan dan tidak memiliki lapisan enkripsi sebelumnya. *Dataset* yang sama digunakan pada kedua algoritma agar proses berlangsung secara adil (*fair comparison*).

3.4 Perangkat Penelitian

Perangkat penelitian yang digunakan terdiri dari perangkat keras dan perangkat lunak yang dikonfigurasi secara konsisten selama seluruh proses pengujian. Perangkat keras yang digunakan berupa komputer dengan spesifikasi tertentu yang mencakup prosesor dan kapasitas *Memory* yang

memadai untuk menjalankan proses enkripsi dan dekripsi. Perangkat lunak yang digunakan meliputi sistem operasi, bahasa pemrograman, serta *library* kriptografi yang mendukung implementasi algoritma AES-GCM dan ChaCha20-Poly1305. Seluruh pengujian dilakukan pada kondisi sistem yang sama, dengan memastikan tidak terdapat proses lain yang signifikan berjalan secara bersamaan, sehingga hasil pengukuran tidak terpengaruh oleh faktor eksternal.

Tabel 3.1 Spesifikasi perangkat penelitian

Jenis	:	Asus Tuf *Gaming* A15 FA506NF_FA506NF
CPU	:	AMD Ryzen 5 7535HS
RAM	:	8 DDR5 4800Mhz DDR5
Ukuran SSD	:	512 GB
GPU	:	Geforce RTX 2050
OS	:	Windows 11 Home *Version* 25H2 (OS *Build* 26200.8655)
Bahasa Pemorgraman	:	*Python* 3.14.3
Software	:	*Visual Studio Code* *Version:* 1.125.1 *(user setup)*
OpenSSL	:	OpenSSL 3.5.5 27 Jan 2026
Library Kriptografi	:	*cryptography* *Version* 46.0.5

Library Python CPU Usage dan *Memory Usage*	:	*Psutil*

3.5 Implementasi Algoritma

Implementasi algoritma dilakukan menggunakan *library Python* yang mendukung kedua algoritma, yaitu AES-GCM dan ChaCha20-Poly1305. Pada tahap ini, setiap *file* PDF diproses melalui dua skenario, yaitu enkripsi dan dekripsi. Proses enkripsi diawali dengan pembangkitan *key* dengan panjang tertentu serta *nonce* atau *initialization vector (IV)* yang digunakan sebagai parameter tambahan dalam algoritma. Selanjutnya, *file* PDF dienkripsi sehingga menghasilkan *ciphertext* dan *authentication tag*. Proses dekripsi dilakukan dengan menggunakan *key* dan *nonce* yang sama untuk mengembalikan *ciphertext* menjadi *plaintext* serta memverifikasi integritas data melalui *authentication tag*. Kedua algoritma dijalankan dengan parameter yang sama untuk memastikan perbandingan yang adil (*fair comparison*). Seluruh proses implementasi dijalankan pada lingkungan perangkat keras dan perangkat lunak yang sama guna meminimalkan pengaruh faktor eksternal terhadap hasil pengukuran.

Pada implementasi AES-GCM, proses dimulai dengan pembangkitan kunci simetris dan *nonce*, kemudian dilanjutkan dengan proses enkripsi *file* PDF hingga menghasilkan *ciphertext* beserta *authentication tag*. Proses dekripsi dilakukan untuk memastikan bahwa data

dapat dikembalikan ke bentuk semula secara valid. Sementara itu, implementasi ChaCha20-Poly1305 dilakukan dengan prosedur yang sepadan, yaitu pembangkitan kunci 256-bit dan *nonce*, proses enkripsi menggunakan ChaCha20, serta pembuatan *authentication tag* menggunakan Poly1305. Proses dekripsi juga dilakukan sebagai tahap verifikasi integritas. Kedua algoritma dijalankan dengan skenario yang identik agar hasil bersifat objektif.

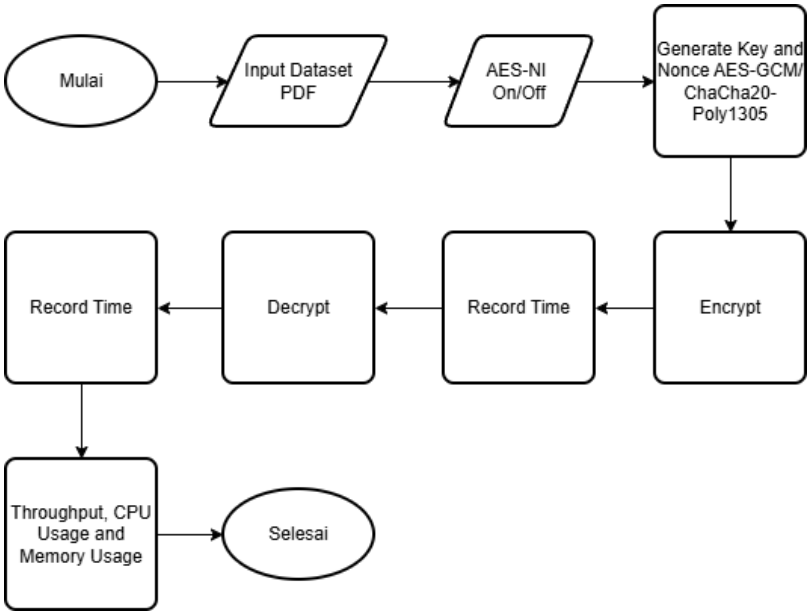
Untuk meningkatkan reliabilitas hasil pengujian, setiap kombinasi algoritma, ukuran *file*, dan kondisi AES-NI diuji sebanyak 20 kali. setiap skenario pengujian dilakukan sebanyak 20 kali iterasi pada setiap ukuran *file* PDF . Nilai yang disajikan pada seluruh tabel hasil pengujian merupakan nilai rata-rata (*mean*) dari 20 kali pengulangan pengujian. Sebelum proses pencatatan data dilakukan, satu kali *warm-up run* dijalankan untuk memastikan *library* dan *resource sistem* telah terinisialisasi dengan baik. Data hasil pengujian tidak mengalami penghapusan *outlier* dan seluruh hasil pengukuran digunakan dalam proses perhitungan rata-rata. Penelitian ini menggunakan nilai rata-rata sebagai dasar analisis performa. Standar deviasi tidak dihitung, sehingga penelitian berfokus pada kecenderungan rata-rata performa setiap algoritma.

Waktu eksekusi diukur menggunakan *modul time.perf_counter()* pada *Python*. Pengukuran dimulai ketika fungsi enkripsi atau dekripsi dipanggil dan dihentikan setelah proses kriptografi selesai dijalankan. Waktu yang dicatat hanya mencakup proses enkripsi atau dekripsi,

sedangkan proses pembacaan *file* dari media penyimpanan dan penulisan hasil ke media penyimpanan tidak termasuk dalam pengukuran. Pendekatan ini digunakan untuk memastikan bahwa nilai waktu eksekusi merepresentasikan performa algoritma kriptografi dan tidak dipengaruhi oleh kecepatan perangkat penyimpanan.

CPU Usage diukur menggunakan *library psutil*. Pengambilan data dilakukan selama proses enkripsi maupun dekripsi berlangsung dengan interval sampling tertentu. Nilai *CPU* usage yang digunakan dalam penelitian merupakan rata-rata dari seluruh sampel yang diperoleh selama proses pengujian. Karena *CPU Usage* dipengaruhi oleh mekanisme penjadwalan sistem operasi, manajemen cache, serta aktivitas proses latar belakang, maka *CPU Usage* digunakan sebagai parameter pendukung untuk menggambarkan penggunaan *resource* prosesor dan tidak digunakan sebagai satu-satunya indikator performa algoritma. *Memory usage* diukur menggunakan *library psutil* dengan memantau konsumsi *Memory Usage* proses selama enkripsi dan dekripsi berlangsung. Nilai *Memory Usage* yang digunakan merupakan nilai puncak (*peak Memory Usage*) yang tercatat selama proses pengujian. *Memory Usage* tidak hanya dipengaruhi oleh algoritma kriptografi yang digunakan, tetapi juga oleh ukuran *file*, mekanisme *buffering*, serta pengelolaan *Memory Usage* oleh sistem operasi. Oleh karena itu, hasil pengukuran *Memory usage* dianalisis secara hati-hati dan tidak digunakan sebagai satu-satunya dasar dalam menentukan algoritma yang memiliki performa lebih baik.

Pada gambar 3.2 menampilkan alur *workflow* dalam pengujian yang dilakukan. *Workflow* pengujian diawali dengan memasukkan *dataset file* PDF yang telah disiapkan dengan variasi ukuran tertentu. Selanjutnya dilakukan pengaturan kondisi AES-NI dalam keadaan aktif atau non-aktif sesuai skenario pengujian. Sistem kemudian membangkitkan *key* dan *nonce* yang digunakan oleh algoritma AES-GCM atau ChaCha20-Poly1305 untuk proses enkripsi. Setelah proses enkripsi selesai, dilakukan pencatatan waktu eksekusi serta pengukuran *throughput*, *CPU Usage*, dan *Memory Usage*. *file* hasil enkripsi kemudian didekripsi menggunakan parameter yang sama, dilanjutkan dengan pencatatan waktu dekripsi serta pengukuran *throughput*, *CPU Usage*, dan *Memory Usage*. Seluruh hasil pengujian disimpan untuk dianalisis dan dibandingkan guna mengetahui performa masing-masing algoritma berdasarkan waktu eksekusi, *throughput*, serta penggunaan *resource*.



Gambar 3.2 *Workflow*

3.6 Skenario AES-NI Aktif dan Nonaktif

Penelitian ini dilakukan pada dua skenario pengujian, yaitu kondisi AES-NI aktif dan AES-NI nonaktif. Pada kondisi AES-NI aktif, *OpenSSL* diizinkan menggunakan seluruh fitur akselerasi kriptografi yang tersedia pada prosesor. Sebaliknya, pada kondisi AES-NI nonaktif, penggunaan instruksi AES-NI dinonaktifkan melalui *environment variable OpenSSL*. Proses deaktivasi AES-NI dilakukan menggunakan perintah berikut:

Tabel 3.2 Proses deaktivasi AES-NI

```
$env:OPENSSL_ia32cap="~0x200000200000000"
```

Environment variable OPENSSL_ia32cap digunakan oleh *OpenSSL* untuk mengatur fitur prosesor yang dapat digunakan selama proses kriptografi. Nilai "*~0x200000200000000*" digunakan untuk menghapus (*masking*) bit fitur AES-NI sehingga *OpenSSL* tidak dapat memanfaatkan instruksi *AESENC, AESDEC, AESENCLAST, AESDECLAST, AESKEYGENASSIST,* dan *AESIMC* yang disediakan oleh prosesor. Dengan konfigurasi tersebut, algoritma AES-GCM dijalankan menggunakan implementasi perangkat lunak (*software implementation*) meskipun prosesor mendukung AES-NI. Sementara itu, ChaCha20-Poly1305 tetap menggunakan implementasi yang sama karena algoritma tersebut tidak bergantung pada instruksi AES-NI.

3.7 Perhitungan Metrik

Pengukuran performa dilakukan selama proses enkripsi dan dekripsi berlangsung. Parameter utama yang diukur dalam penelitian ini meliputi waktu enkripsi, waktu dekripsi, *throughput*, *CPU Usage*, dan *Memory Usage*. Waktu enkripsi dan dekripsi diukur untuk mengetahui kecepatan pemrosesan masing-masing algoritma, sedangkan *throughput* digunakan untuk menggambarkan jumlah data yang dapat diproses per satuan waktu. Selain itu, *CPU Usage* diukur untuk mengetahui tingkat beban prosesor selama proses kriptografi berlangsung, sedangkan *Memory Usage* diukur untuk mengevaluasi efisiensi pemakaian *RAM*. Seluruh pengujian dilakukan secara berulang sebanyak beberapa iterasi untuk memperoleh nilai rata-rata yang lebih representatif dan mengurangi fluktuasi hasil.

Parameter pertama yang digunakan adalah waktu enkripsi, yaitu durasi yang dibutuhkan algoritma untuk mengubah *file* PDF dari bentuk *plaintext* menjadi *ciphertext*. Pengukuran ini penting karena menunjukkan kecepatan algoritma dalam melindungi data. Semakin kecil waktu enkripsi yang dihasilkan, maka semakin baik performa algoritma dalam konteks kecepatan pemrosesan.

Parameter kedua adalah waktu dekripsi, yang merupakan waktu yang diperlukan untuk mengembalikan *ciphertext* menjadi *plaintext* yang valid. Pengukuran waktu dekripsi digunakan untuk menilai efisiensi algoritma dalam proses pemulihan data. Sama seperti waktu enkripsi, nilai waktu dekripsi yang lebih kecil menunjukkan performa yang lebih efisien.

Berikut flowchart ini adalah alur enkripsi dan dekripsi dari algoritma AES-GCm dan ChaCha20-Poly1305

Parameter ketiga adalah *throughput*, yaitu jumlah data yang dapat diproses oleh algoritma per satuan waktu. *Throughput* dihitung dengan membagi ukuran *file* terhadap waktu proses yang dibutuhkan. Metrik ini memberikan gambaran kapasitas pemrosesan algoritma dalam menangani data berukuran besar. Nilai *throughput* yang lebih tinggi menunjukkan bahwa algoritma mampu memproses data secara lebih efisien. Selain metrik berbasis waktu, penelitian ini juga mengevaluasi penggunaan *resource*.

Parameter keempat adalah *CPU Usage*, yaitu persentase pemakaian prosesor selama proses enkripsi dan dekripsi berlangsung. Pengukuran *CPU Usage* bertujuan untuk mengetahui tingkat beban komputasi yang ditimbulkan oleh masing-masing algoritma. Algoritma yang membutuhkan *CPU Usage* lebih rendah dianggap lebih efisien dari sisi komputasi.

Parameter kelima adalah *Memory Usage*, yaitu jumlah *RAM* yang digunakan selama proses kriptografi berlangsung. Pengukuran ini penting untuk menilai efisiensi algoritma pada sistem dengan keterbatasan *Memory*. Algoritma yang menggunakan *Memory* lebih sedikit dinilai lebih hemat sumber daya.

Seluruh parameter pengukuran tersebut dipilih untuk memberikan evaluasi performa yang menyeluruh, tidak hanya dari sisi kecepatan pemrosesan tetapi juga dari aspek efisiensi penggunaan *resource* sistem. Dengan menggunakan kombinasi metrik tersebut, penelitian ini diharapkan

mampu menghasilkan performa yang lebih komprehensif antara AES-GCM dan ChaCha20-Poly1305 pada pengamanan *file* PDF berukuran besar.

3.8 Analisis Data

Analisis data dalam penelitian ini dilakukan secara kuantitatif menggunakan pendekatan deskriptif terhadap hasil pengujian dua algoritma enkripsi, yaitu AES-GCM dan ChaCha20-Poly1305. Proses ini diawali dengan pengumpulan data berupa waktu proses dan ukuran *file* hasil enkripsi dari dua skenario eksperimen yang telah dijalankan sebelumnya. Data yang diperoleh dari proses pengujian dianalisis menggunakan pendekatan komparatif kuantitatif. Analisis dilakukan dengan menghitung nilai rata-rata setiap metrik performa, kemudian membandingkan hasil antara AES-GCM dan ChaCha20-Poly1305 pada setiap variasi ukuran *file*. Selain itu, dilakukan analisis tren untuk melihat pengaruh peningkatan ukuran *file* terhadap perubahan performa dan penggunaan *resource*. Hasil analisis disajikan dalam bentuk tabel dan grafik untuk mempermudah interpretasi. Berdasarkan hasil tersebut kemudian ditentukan algoritma yang memiliki performa lebih efisien dalam penggunaan *resource*.

Tahap analisis data merupakan proses pengolahan dan interpretasi terhadap data hasil pengujian yang diperoleh dari proses implementasi algoritma AES-GCM dan ChaCha20-Poly1305. Analisis ini bertujuan untuk mengetahui performa kedua algoritma dalam proses enkripsi dan dekripsi terhadap berkas dokumen PDF berdasarkan beberapa parameter pengukuran yang telah ditentukan, yaitu waktu eksekusi, *throughput*,

Memory Usage, dan *CPU Usage*. Data yang diperoleh pada tahap pengujian berupa hasil pencatatan waktu proses enkripsi dan dekripsi, jumlah data yang diproses dalam satuan waktu, serta penggunaan *resource* selama proses kriptografi berlangsung. Pengujian dilakukan menggunakan berkas PDF dengan variasi ukuran *file*, yaitu 10 MB, 20 MB, 30 MB, 40 MB, 50 MB, 60 MB, 70 MB, 80 MB, 90 MB, dan 100 MB. Variasi ukuran *file* tersebut digunakan untuk melihat pengaruh ukuran data terhadap performa algoritma yang diuji.

3.9 Validitas Pengujian

Untuk menjaga validitas hasil pengujian, seluruh eksperimen dilakukan menggunakan perangkat keras, sistem operasi, versi *Python*, *library cryptography*, dan *dataset* yang sama pada seluruh skenario pengujian. Variabel yang diubah hanya kondisi akselerasi AES-NI dan algoritma yang digunakan, sedangkan parameter lain dipertahankan konstan agar hasil yang diperoleh dapat dibandingkan secara adil. Setiap kombinasi algoritma dan ukuran *file* diuji sebanyak 20 kali. Nilai yang disajikan pada penelitian merupakan nilai rata-rata dari seluruh pengulangan untuk mengurangi pengaruh fluktuasi sistem operasi selama proses pengujian.

Dataset yang digunakan berupa *file* PDF dengan ukuran 10 MB hingga 100 MB. Seluruh pengujian dilakukan menggunakan implementasi AES-GCM dan ChaCha20-Poly1305 dari *library cryptography* dengan panjang kunci, ukuran *nonce*, dan konfigurasi yang sesuai standar *NIST SP 800-38D* dan *RFC 8439*. Pengukuran waktu dilakukan terhadap

keseluruhan proses enkripsi maupun dekripsi (*end-to-end benchmark*) sehingga hasil yang diperoleh mencerminkan performa implementasi algoritma pada lingkungan pengujian yang digunakan.

Penelitian ini menggunakan satu konfigurasi perangkat keras dan satu sistem operasi sehingga hasil yang diperoleh merepresentasikan lingkungan pengujian tersebut. Generalisasi terhadap platform lain memerlukan pengujian tambahan.

Tabel 3. 3 Variabel Penelitian

Jenis variabel	Variabel
Variabel Bebas	AES-GCM dan ChaCha20-Poly1305
Variabel Terikat	Waktu enkripsi, waktu dekripsi, *throughput*, *CPU Usage*, *Memory Usage*

3.10 Penarikan Kesimpulan

Tahap akhir penelitian adalah penarikan kesimpulan yang dilakukan berdasarkan hasil analisis data. Kesimpulan disusun untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya, serta untuk mengidentifikasi algoritma yang lebih optimal dalam pengamanan *file* PDF berukuran besar. Selain itu, pada bagian ini juga disampaikan rekomendasi untuk penelitian selanjutnya yang berkaitan dengan evaluasi performa algoritma kriptografi.

Dalam penelitian mengenai perbandingan algoritma AES-GCM dan ChaCha20-Poly1305, penarikan kesimpulan dilakukan setelah seluruh proses eksperimen dan analisis data selesai dilakukan. Data hasil pengujian yang meliputi waktu proses enkripsi dan dekripsi, *CPU Usage*, serta *Memory Usage* dianalisis secara komprehensif untuk mengetahui tingkat efisiensi dan performa dari kedua algoritma tersebut. Analisis ini dilakukan dengan cara membandingkan hasil pengukuran pada setiap parameter yang telah ditentukan sebelumnya, sehingga dapat diketahui algoritma mana yang memiliki performa lebih optimal dalam kondisi pengujian yang sama. Selanjutnya, hasil analisis tersebut disintesiskan untuk menjawab rumusan masalah dan tujuan penelitian yang telah ditetapkan pada bagian awal penelitian. Peneliti mengidentifikasi pola, kelebihan, serta keterbatasan dari masing-masing algoritma berdasarkan hasil eksperimen yang diperoleh. Dari proses tersebut, dapat ditarik kesimpulan mengenai algoritma yang lebih efisien dalam hal kecepatan proses, penggunaan *resource*, serta efektivitas dalam mengamankan data yang diuji.

Selain itu, pada tahap ini peneliti juga memberikan interpretasi ilmiah terhadap hasil penelitian dengan mengaitkannya dengan teori kriptografi serta penelitian-penelitian terdahulu yang relevan. Dengan demikian, kesimpulan yang dihasilkan tidak hanya bersifat deskriptif terhadap hasil eksperimen, tetapi juga memiliki landasan teoritis yang kuat. Hasil kesimpulan ini diharapkan dapat memberikan gambaran yang jelas mengenai performa algoritma AES-GCM dan ChaCha20-Poly1305 dalam

konteks pengamanan data, khususnya pada skenario pengujian yang menggunakan berkas digital seperti dokumen. Pada bagian akhir tahap ini, peneliti juga dapat memberikan rekomendasi terkait penggunaan algoritma yang lebih optimal berdasarkan hasil penelitian yang diperoleh. Rekomendasi tersebut dapat menjadi acuan bagi penelitian selanjutnya maupun bagi pengembang sistem yang membutuhkan algoritma kriptografi dengan performa yang efisien dan tingkat keamanan yang tinggi. Dengan demikian, tahap penarikan kesimpulan tidak hanya berfungsi sebagai rangkuman hasil penelitian, tetapi juga sebagai dasar dalam memberikan kontribusi ilmiah terhadap pengembangan kajian di bidang *Cryptography*.