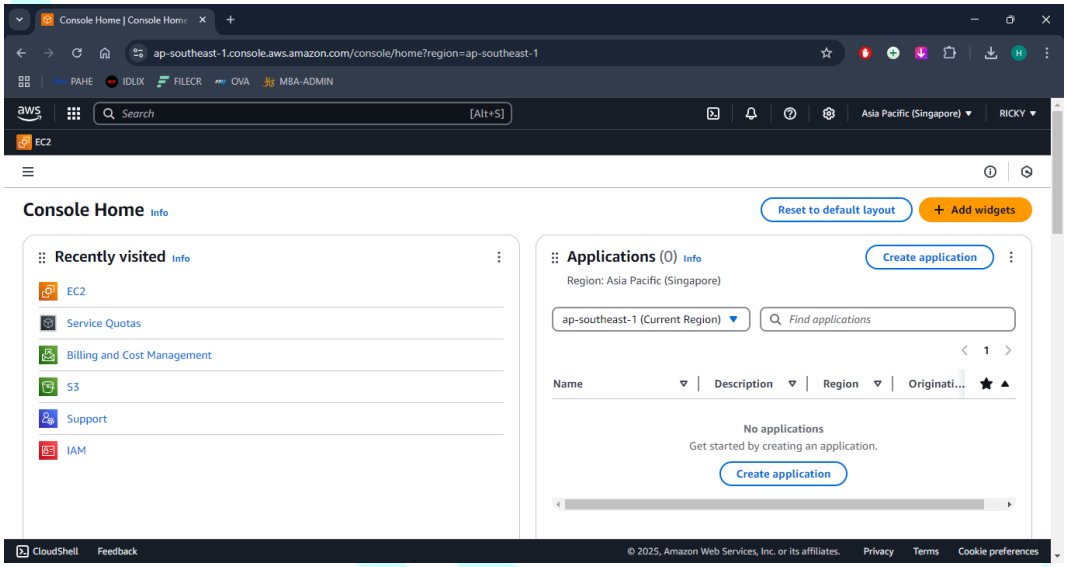
BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Penelitian ini berfokus pada penerapan enkripsi kombinasi algoritma *AES (Advanced Encryption Standard)* dan *RSA (Rivest-Shamir-Adleman)* untuk meningkatkan keamanan data dalam layanan *cloud storage berbasis Amazon S3. Cloud storage* telah menjadi solusi yang signifikan dalam berbagai sektor karena fleksibilitas dan efisiensinya, tetapi tetap menghadapi tantangan utama terkait keamanan data, termasuk ancaman serangan *eksternal* dan *internal* seperti *DoS (Denial of Service)* yang memengaruhi stabilitas layanan *cloud* (Alfazry et al., 2024).



Gambar 3.1 *Console home AWS*

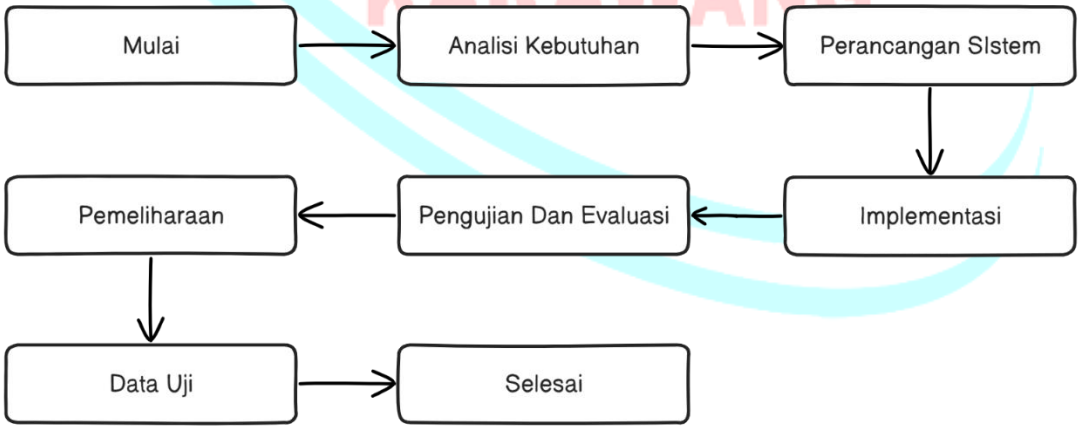
Tujuan dari penelitian ini adalah merancang dan mengimplementasikan model enkripsi yang memberikan perlindungan menyeluruh terhadap data pengguna. Kombinasi algoritma *AES* dan *RSA* dipilih karena sifatnya yang saling melengkapi, *AES* unggul dalam enkripsi data besar dengan kecepatan tinggi, sementara *RSA* memberikan keamanan tambahan dalam manajemen kunci. Penelitian sebelumnya menunjukkan bahwa kombinasi kedua algoritma ini dapat diterapkan secara efektif untuk meningkatkan keamanan data dalam berbagai skenario, termasuk pengamanan sertifikat elektronik menggunakan skema *QR-Code* dan tanda tangan digital (Nuraeni et al., 2020).

Penelitian ini juga mempertimbangkan efisiensi kinerja platform *AWS Free Tier*, sebagaimana diukur dalam pengujian sebelumnya yang menunjukkan bahwa platform ini mampu memberikan solusi komputasi dengan keterbatasan sumber daya tanpa mengurangi fungsionalitasnya (Nur Hidayat et al., 2024). Pendekatan ini sejalan dengan prinsip *Confidentiality, Integrity, and Availability (CIA),* yang menjadi standar utama dalam menjaga keamanan data *cloud* (Marliana, 2022). Dengan implementasi ini, diharapkan layanan *cloud stirage* berbasis *Amazon S3* dapat memberikan pengalaman yang aman dan efisien bagi penggunanya.

Penelitian ini dibatasi pada penerapan model enkripsi kombinasi *AES* dan *RSA* pada *platform Amazon S3*. Fokusnya adalah pada pengamanan data pengguna, tanpa mencakup pengujian pada platform *cloud* lainnya atau penggunaan algoritma selain *AES* dan *RSA*. Lingkup ini dipilih untuk mengeksplorasi solusi keamanan pada layanan *cloud* yang sering digunakan oleh individu dan organisasi kecil.

3.2 Prosedur Penelitian

Penelitian ini menggunakan model pengembangan perangkat lunak *Waterfall* yang sistematis dan berurutan, dengan pengujian menggunakan metode *Black box* untuk memastikan aplikasi enkripsi *hybrid AES-RSA* yang terintegrasi dengan layanan *cloud AWS S3* berjalan sesuai harapan. Tahapan pengembangan yang dilakukan dapat dilihat pada gambar alur penelitian.



Gambar 3.2 Alur penelitian

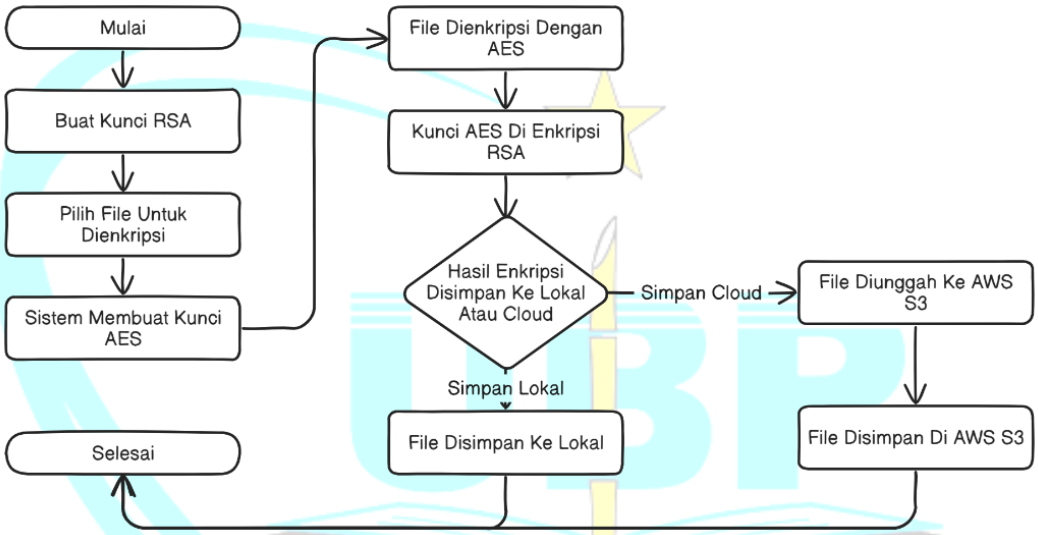
3.2.1 Analisis Kebutuhan

Pada tahap awal, dilakukan identifikasi kebutuhan sistem, seperti fitur enkripsi dan dekripsi, jenis file yang didukung *(.docx, .jpg, .mp4)*, serta integrasi

dengan *AWS S3* sebagai media penyimpanan file terenkripsi. Analisis ini menjadi dasar perancangan aplikasi.

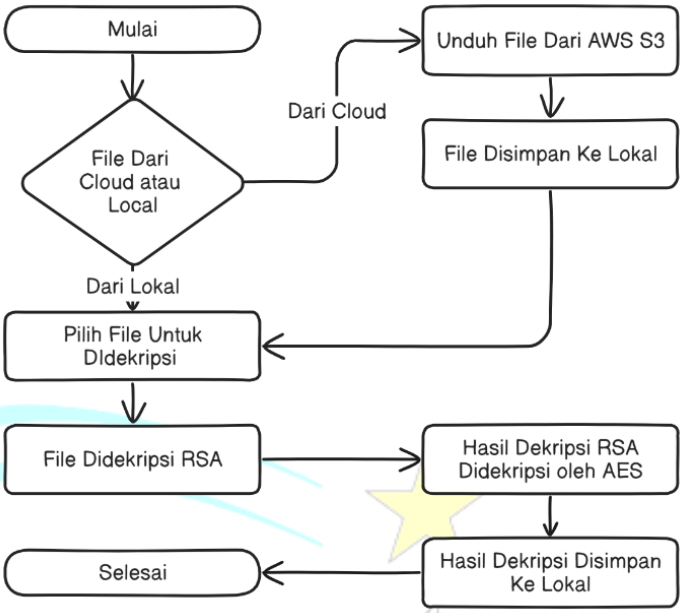
3.2.2 Perancangan Sistem

Merancang arsitektur aplikasi yang mencakup desain antarmuka pengguna berbasis *Python* dan *Tkinter*, alur kerja enkripsi *hybrid AES-RSA*, serta mekanisme pengelolaan kunci enkripsi. Desain juga mencakup integrasi dengan layanan *AWS S3* melalui *API*.



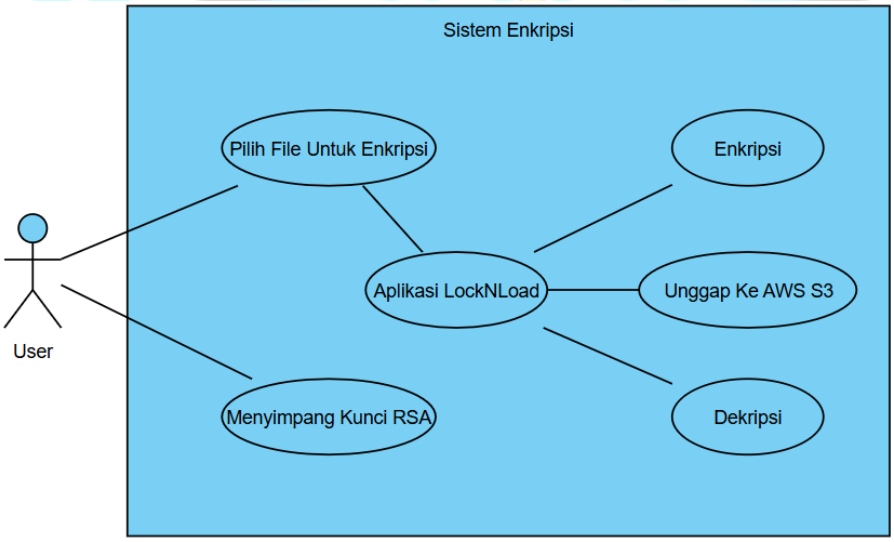
Gambar 3.3 *Flowchart* enkripsi

Pada proses enkripsi *user* diharus membuat terlebih dahulu kunci untuk *RSA* untuk melakukan enkripsi. Setelah itu memilih *file* yang akan di enkripsi dan memasukan kunci publik *RSA* terakhir memilih akan menyimpan *file* hasil enkripsi di lokal atau *cloud*. Pada tahap ini user bebas menentukan lokasi penyimpanan dan melakukan enkripsi.



Gambar 3.4 *Flowchart* dekripsi

Saat proses dekripsi *user* harus menyiapkan file yang telah dienkripsi. Jika *file* ada di *AWS S3* maka *user* harus mengunduh terlebih dahulu *file*nya, setelah itu pilih *file*nya dilokal dan pilih kunci privat *RSA* untuk melakukan dekripsi dan hasil *file* dekripsi akan disimpan dilokasi file sebelumnya.



Gambar 3.5 Diagram *use case*

Diagram ini memperlihatkan proses kerja utama dari aplikasi *LockNLoad* yang berfungsi untuk melindungi data menggunakan metode enkripsi *hybrid* serta terhubung dengan layanan *AWS S3*. Pengguna dapat dengan mudah memilih *file* yang ingin diamankan, melakukan enkripsi, mengunggah *file* terenkripsi ke *cloud*,

dan melakukan dekripsi saat diperlukan. Sistem ini juga mengelola kunci *RSA* dengan aman, sehingga hanya pengguna yang memiliki kunci yang sah yang dapat mengakses data tersebut. Dengan demikian, aplikasi ini menjamin keamanan data selama penyimpanan dan pengiriman di lingkungan *cloud*.

3.2.3 Implementasi

Tahap ini meliputi pengkodean aplikasi sesuai rancangan menggunakan bahasa Python. Algoritma *AES* dan *RSA* diimplementasikan untuk proses enkripsi dan dekripsi, serta integrasi dengan *AWS S3* dilakukan untuk penyimpanan dan pengambilan *file*.

3.2.4 Pengujian dan Evaluasi

Pengujian dilakukan dengan metode *Black box*, yang berfokus pada pengujian fungsi aplikasi dari sisi pengguna tanpa melihat kode sumber. Pengujian meliputi:

1. Pengujian fungsional untuk memastikan proses enkripsi, dekripsi, *upload*, dan *download file* berjalan dengan baik.
2. Pengujian validasi input untuk memastikan aplikasi dapat menangani *file* yang tidak valid atau rusak dengan benar.
3. Pengujian kinerja untuk mengukur waktu proses enkripsi dan dekripsi serta respon aplikasi saat berinteraksi dengan *AWS S3*.

Hasil pengujian dianalisis untuk menilai keandalan, keamanan, dan performa aplikasi dalam melindungi data pengguna.

3.2.5 Pemeliharaan

Setelah aplikasi dinyatakan memenuhi standar kelayakan dan siap digunakan, tahap pemeliharaan menjadi langkah penting berikutnya. Pada fase ini, pengembang secara rutin melakukan identifikasi dan perbaikan terhadap kesalahan atau *bug* yang mungkin muncul selama penggunaan aplikasi. Selain itu, pemeliharaan juga mencakup pembaruan sistem untuk meningkatkan performa, menambah fitur baru, atau menyesuaikan dengan perubahan kebutuhan pengguna dan teknologi terkini. Proses ini bertujuan agar aplikasi tetap berjalan dengan optimal, aman, dan relevan dalam jangka panjang, sehingga memberikan pengalaman terbaik bagi penggunanya.

3.2.6 Data Uji

Data yang dijadikan bahan dalam penelitian ini meliputi berbagai jenis *file*, yaitu dokumen berformat *.docx*, gambar dengan ekstensi *.jpg*, serta video berformat *.mp4*. *File-file* tersebut dipilih dengan ukuran yang bervariasi, mulai dari yang sangat kecil sekitar 100 *KB* hingga yang cukup besar mencapai 200 *MB*. Pemilihan rentang ukuran *file* yang luas ini bertujuan untuk menguji dan memperoleh gambaran menyeluruh mengenai kinerja aplikasi ketika menangani data dalam berbagai skala. Dengan demikian, evaluasi performa yang dilakukan dapat mencerminkan kemampuan aplikasi dalam mengelola file dengan ukuran dan tipe yang berbeda-beda secara efektif dan efisien.

Tabel 3.1 Data Uji

No.	Jenis File	Ukuran File
1.	*.docx*	100 *KB*
2.	*.docx*	1 *MB*
3.	*.docx*	5 *MB*
4.	*.jpg*	1 *MB*
5.	*.jpg*	5 *MB*
6.	*.jpg*	10 *MB*
7.	*.mp4*	50 *MB*
8.	*.mp4*	100 *MB*
9.	*.mp4*	200 *MB*