BAB III
METODE PENELITIAN

3.1. Desain Penelitian

Penelitian ini dilakukan menggunakan metode *Design and Development* (D&D), sebuah pendekatan metodologis yang terstruktur untuk mengatasi isu penelitian dengan desain sistematis dan evaluasi artefak. Metode D&D dipilih karena kemampuannya mengintegrasikan teori dan praktik dalam penciptaan produk atau model yang menawarkan solusi kreatif untuk masalah yang diidentifikasi. Metode ini efektif dalam menentukan area penelitian yang relevan dan mengembangkan argumen yang kuat, sambil menyediakan kerangka yang mendukung inovasi (Ellis & Levy, 2010). Selain itu, metode ini berfokus pada penciptaan artefak yang dapat langsung diterapkan dalam konteks nyata, serta memastikan bahwa solusi tersebut efektif dan efisien dalam menyelesaikan masalah yang telah diidentifikasi. Penelitian menggunakan metode ini sangat sesuai untuk pengembangkan produk atau aplikasi, seperti aplikasi berbasis web untuk keamanan gambar menggunakan algoritma kriptografi DES dan AES serta teknik steganografi metadata. Untuk melihat bagaimana desain penelitian dengan metode *Design and Development* dalam penelitian ini, dapat dilihat pada Gambar 3.1.

Gambar 3.1 Desain Penelitian

3.2. Analisis

Tahap pertama dalam metode penelitian *Design and Development* (D&D) adalah analisis. Tahap ini dimulai dengan melakukan studi literatur terhadap topik penelitian yang akan diambil. Melalui studi literatur yang relevan, dapat diperoleh informasi tentang perkembangan ilmu dan hasil penelitian terdahulu yang berkaitan dengan keamanan gambar digital. Pada tahap ini, dilakukan identifikasi masalah yang ada, perumusan masalah, serta penetapan tujuan penelitian. Selain itu, pemahaman tentang berbagai pendekatan dan solusi yang telah diuji sebelumnya menjadi dasar dalam pengembangan aplikasi. Dalam konteks penelitian ini, ide

untuk mengembangkan aplikasi berbasis web yang mampu mengamankan gambar menggunakan algoritma kriptografi AES dan DES serta teknik steganografi pada metadata gambar muncul dari hasil studi literatur tersebut.

Studi literatur mendalam dilakukan mengenai algoritma enkripsi AES dan DES untuk memahami keunggulan dan kelemahan masing-masing dalam konteks keamanan data gambar. Selain itu, teknik steganografi pada metadata gambar dipelajari untuk menyembunyikan data terenkripsi secara efisien tanpa menimbulkan kecurigaan. Penelitian juga mencakup pengembangan aplikasi berbasis web untuk memastikan bahwa solusi yang diusulkan dapat diimplementasikan dengan baik dan mudah diakses oleh pengguna. Melalui analisis ini, diperoleh pemahaman yang komprehensif tentang teknologi yang relevan dan bagaimana mengintegrasikannya untuk menciptakan aplikasi yang efektif dalam melindungi data gambar dari akses yang tidak sah.

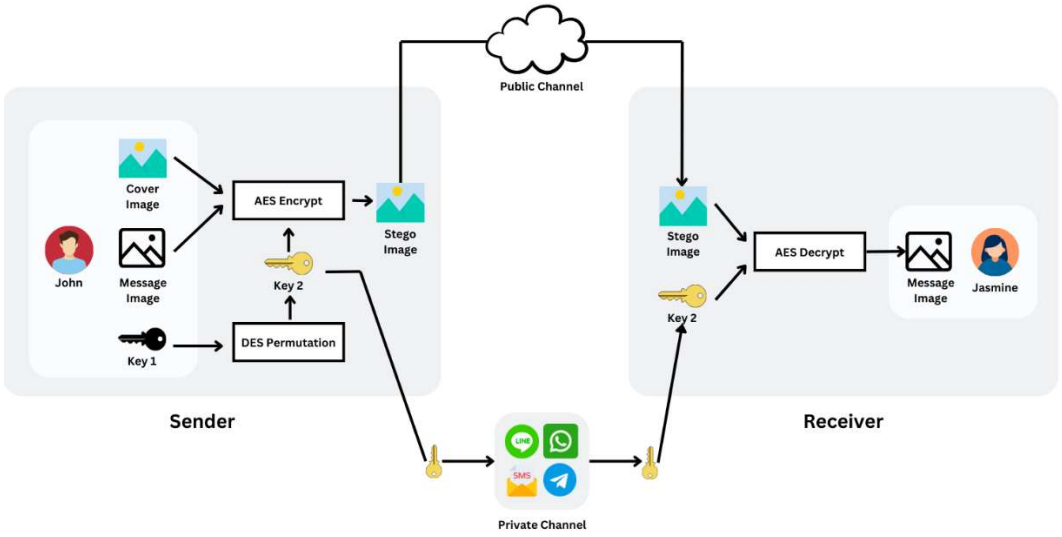
3.3. Desain atau Perancangan Sistem

Tahap kedua adalah tahap desain atau perancangan sistem. Pada tahap ini, penulis mengembangkan kerangka konseptual yang penting dalam proses desain aplikasi. Proses perancangan sistem ini mencakup pembuatan berbagai diagram untuk memvisualisasikan arsitektur sistem secara keseluruhan. Diagram yang dibuat meliputi diagram *use case* untuk menggambarkan interaksi antara pengguna dan sistem, serta diagram aktivitas yang menunjukkan alur kerja atau proses dalam sistem tersebut. Dengan adanya diagram-diagram ini, perancangan sistem menjadi lebih jelas dan terstruktur, sehingga memudahkan pengembang dalam implementasi aplikasi yang sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

3.3.1. Desain Arsitektur Sistem

Desain Dalam sistem kriptografi yang diterapkan untuk pengiriman pesan melalui jaringan publik, penting untuk menjaga kerahasiaan dan integritas data yang dikirim. Untuk mencapai tujuan ini, metode steganografi sering digunakan sebagai teknik tambahan untuk menyembunyikan pesan dalam media lain, seperti gambar, sehingga keberadaan pesan tersebut tidak terdeteksi oleh pihak ketiga yang tidak berwenang. Dalam konteks ini, kombinasi antara algoritma kriptografi Data Encryption Standard (DES) dan Advanced Encryption Standard (AES) dapat memberikan lapisan keamanan ganda yang memperkuat perlindungan data.

Proses pengiriman pesan dimulai dengan John, sebagai pengirim, yang menyiapkan sebuah gambar sampul (*cover image*) dan gambar pesan (*message image*). Gambar pesan dienkripsi menggunakan algoritma AES dengan kunci tertentu (*Key 2*) setelah melalui proses permutasi dengan algoritma DES dengan kunci lain (*Key 1*). Hasil dari proses ini adalah gambar pesan yang terenkripsi, yang kemudian disisipkan ke dalam gambar sampul untuk menghasilkan gambar stego. Gambar stego ini kemudian dikirimkan melalui saluran publik ke penerima, Jasmine. Di sisi penerima, Jasmine menggunakan kunci yang sama (*Key 2*) yang didapatkan melalui saluran pribadi sesuai dengan kesepakatan. *Key 2* digunakan untuk mendekripsi gambar stego dan mengambil kembali gambar pesan asli. Gambar 2.3 menampilkan arsitektur sistem yang digunakan dalam proses ini.

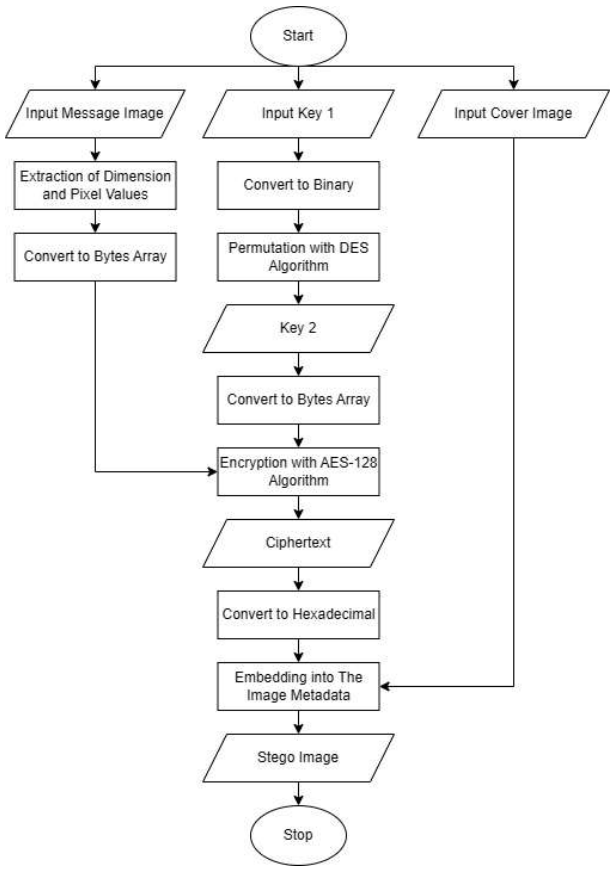


Gambar 3.2 Desain Arsitektur Sistem

3.3.2. Diagram Alir Sistem

Proses dimulai dengan pengambilan input dari beberapa komponen, yaitu gambar pesan (*message image*), kunci pertama (*key 1*), dan gambar sampul (*cover image*). Gambar pesan diolah dengan mengekstraksi nilai dimensi dan pikselnya, yang kemudian dikonversi menjadi *array byte*. Sementara itu, kunci pertama dikonversi ke dalam bentuk biner, dan selanjutnya diproses melalui permutasi menggunakan algoritma pembangkitan kunci DES untuk meningkatkan keamanan kunci. Hasil permutasi ini menghasilkan kunci kedua, yang kemudian juga dikonversi menjadi *array byte*.

Data gambar pesan dalam bentuk *array byte* tersebut kemudian dienkripsi menggunakan algoritma AES-128 dengan memanfaatkan kunci yang telah dihasilkan. Hasil dari proses enkripsi ini adalah *ciphertext*, yang selanjutnya dikonversi ke dalam format heksadesimal. Data *ciphertext* dalam bentuk heksadesimal ini kemudian disisipkan ke dalam metadata gambar sampul, sehingga menghasilkan gambar akhir yang disebut *stego image*. *Stego image* ini berfungsi sebagai media yang menyimpan pesan tersembunyi dalam metadata nya. Setelah semua proses ini selesai, sistem mencapai titik akhir, menandakan bahwa proses enkripsi dan penyisipan telah berhasil diselesaikan. Gambar 3.3 akan menampilkan detail dari alur proses ini dalam bentuk diagram alir.

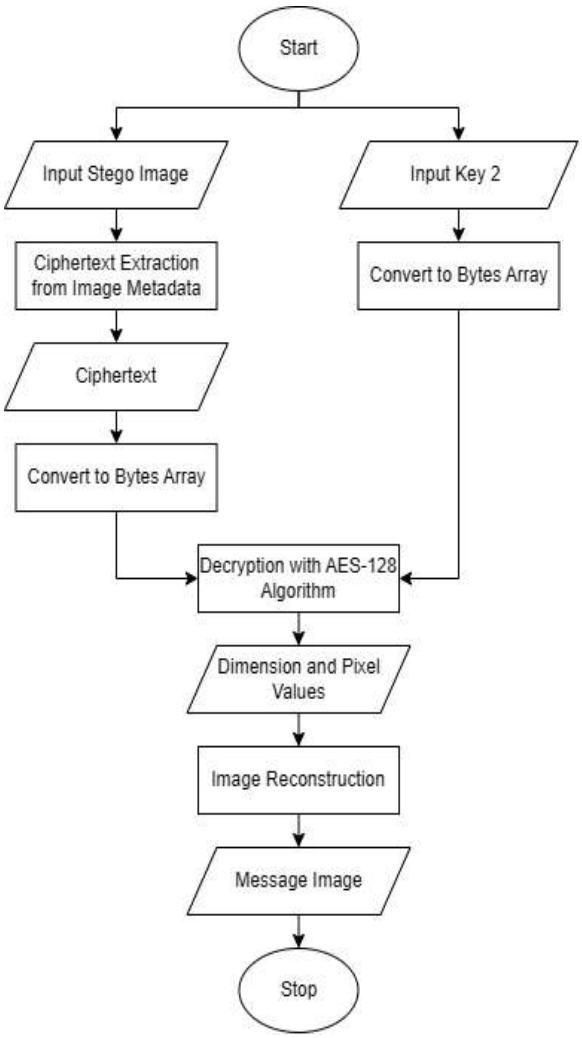


Gambar 3.3 Diagram Alir Sistem Enkripsi

Proses dekripsi dimulai dengan pengambilan input dari dua sumber utama, yaitu gambar steganografi (*stego image*) dan kunci kedua (*key 2*). Dari gambar stego, dilakukan ekstraksi *ciphertext* yang tersembunyi dalam metadata. *Ciphertext* yang telah diekstraksi ini kemudian dikonversi menjadi *array byte*, mempersiapkan data untuk proses dekripsi. Kunci kedua yang diinputkan juga dikonversi menjadi

array byte, yang diperlukan untuk proses dekripsi menggunakan algoritma AES-128. Setelah kedua input tersebut siap dalam bentuk *array byte*, proses dekripsi dijalankan menggunakan algoritma AES-128, yang mengembalikan data asli dalam bentuk dimensi dan nilai piksel gambar.

Selanjutnya, nilai dimensi dan piksel yang telah diperoleh dari proses dekripsi akan digunakan untuk merekonstruksi gambar pesan (*message image*). Proses rekonstruksi ini menghasilkan gambar pesan yang sama dengan gambar asli yang dienkripsi pada tahap awal. Setelah gambar pesan berhasil direkonstruksi, sistem mencapai titik akhir, menandakan bahwa proses dekripsi telah selesai dengan sukses. Diagram alir ini menjelaskan tahapan dekripsi dari pengambilan data tersembunyi dalam *stego image* hingga proses pemulihan gambar pesan asli, yang diilustrasikan dalam Gambar 3.4.

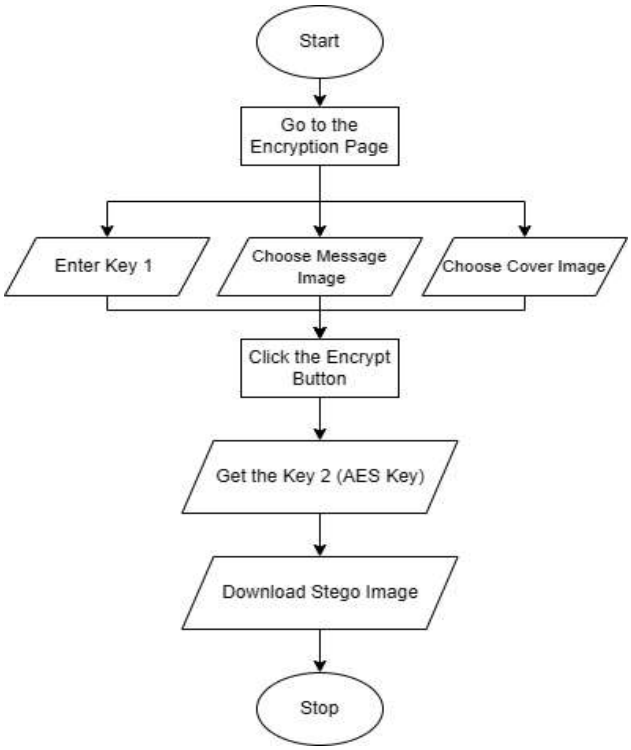


Gambar 3.4 Diagram Alir Sistem Dekripsi

3.3.3. Diagram Alir Aplikasi

Proses aplikasi enkripsi dimulai ketika pengguna mengakses halaman enkripsi (*Encryption Page*). Pada halaman ini, pengguna diminta untuk memasukkan Kunci 1 sebanyak 8 karakter ASCII (*Key 1*), memilih gambar pesan (*Message Image*), dan memilih gambar sampul (*Cover Image).* Setelah semua input yang diperlukan dimasukkan, pengguna menekan tombol enkripsi (*Encrypt Button*) untuk memulai proses enkripsi. Setelah tombol enkripsi ditekan, sistem melakukan proses enkripsi, termasuk menghasilkan Kunci 2 (*Key 2*) yang akan digunakan sebagai kunci AES untuk enkripsi gambar pesan. Kunci 2 ini diperoleh dan disimpan untuk keperluan proses dekripsi nanti. Setelah proses enkripsi selesai, sistem akan menghasilkan gambar steganografi (*Stego Image*) yang berisi data terenkripsi di dalam metadata nya. Pengguna kemudian diberikan opsi untuk mengunduh gambar stego tersebut melalui fitur "*Download Stego Image*".

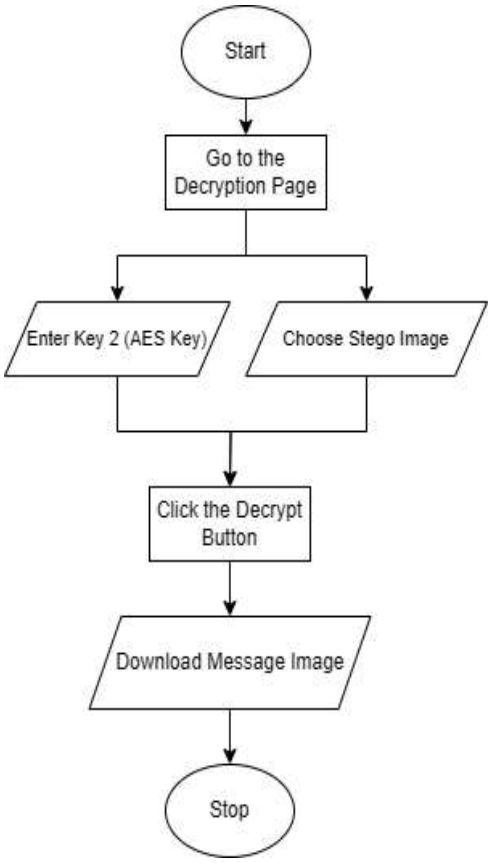
Setelah gambar stego berhasil diunduh, proses pada aplikasi enkripsi selesai, dan diagram alir ini mencapai titik akhir. Gambar 3.5 akan menampilkan secara visual langkah-langkah yang dijelaskan ini dalam bentuk diagram alir aplikasi enkripsi.



Gambar 3.5 Diagram Alir Aplikasi Enkripsi

Setelah proses enkripsi, selanjutnya adalah proses dekripsi yang dimulai ketika pengguna mengakses halaman dekripsi (*Decryption Page*) pada aplikasi. Di halaman ini, pengguna diminta untuk memasukkan Kunci 2 (*Key 2*), yang merupakan kunci AES yang digunakan selama proses enkripsi. Selain itu, pengguna juga harus memilih gambar steganografi (*Stego Image*) yang berisi data terenkripsi yang akan didekripsi. Setelah Kunci 2 dan gambar stego dipilih, pengguna menekan tombol dekripsi (*Decrypt Button*) untuk memulai proses dekripsi. Sistem kemudian akan menggunakan Kunci 2 untuk mendekripsi data yang tersembunyi di dalam metadata gambar stego. Hasil dekripsi adalah gambar pesan asli (*Message Image*), yang kemudian dapat diunduh oleh pengguna melalui fitur "Download Message Image".

Setelah pengguna berhasil mengunduh gambar pesan, proses dekripsi pada aplikasi selesai, dan diagram alir ini mencapai titik akhir. Gambar 3.6 akan menampilkan secara visual langkah-langkah yang dijelaskan ini dalam bentuk diagram alir aplikasi dekripsi.



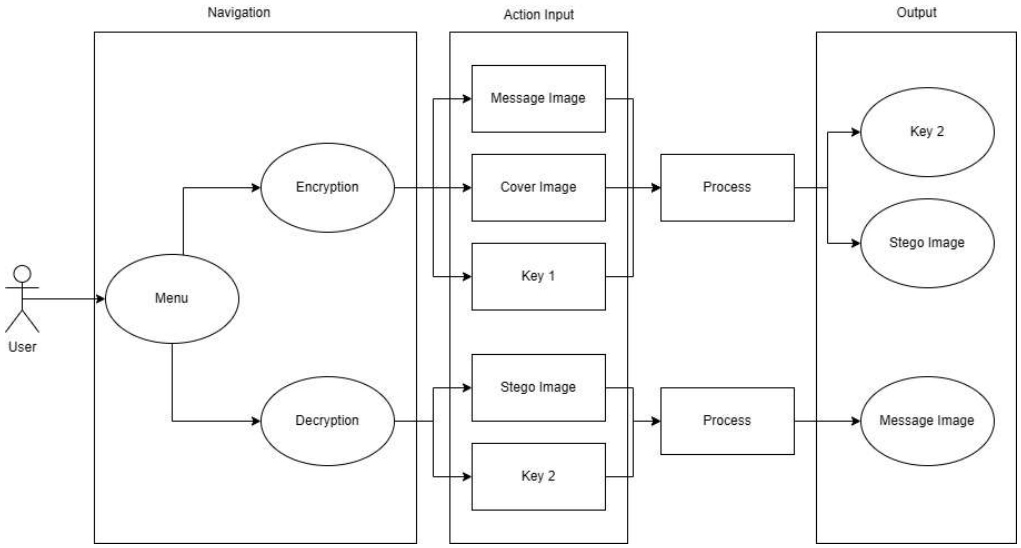
Gambar 3.6 Diagram Alir Aplikasi Dekripsi

3.3.4. Diagram *Use Case*

Diagram *Use Case* pada Gambar 3.7 digunakan untuk menggambarkan bagaimana pengguna (*User*) berinteraksi dengan sistem aplikasi, mencakup seluruh alur dari pengoperasian sistem untuk tujuan enkripsi dan dekripsi data. Diagram ini mencakup tiga bagian utama, yaitu Navigasi, Input Aksi, dan Output. Bagian Navigasi menunjukkan bagaimana pengguna mengakses aplikasi melalui menu utama. Dari menu ini, pengguna memiliki dua pilihan utama, diantaranya yaitu Enkripsi, yang memungkinkan pengguna untuk mengenkripsi gambar pesan dan menyisipkannya pada gambar lain, dan Dekripsi, yang memungkinkan pengguna untuk mengembalikan pesan yang tersembunyi di dalam gambar.

Dalam bagian Input Aksi, diagram merinci input yang diperlukan untuk setiap proses. Untuk enkripsi, input meliputi gambar pesan (*Message Image*) yang ingin disembunyikan, gambar sampul (*Cover Image*) yang akan digunakan untuk menyisipkan pesan, dan Kunci 1 (*Key 1*) yang diperlukan untuk proses enkripsi. Sementara itu, untuk dekripsi, pengguna perlu menyediakan gambar steganografi (*Stego Image*) yang berisi pesan tersembunyi dan Kunci 2 (*Key 2*) yang diperlukan untuk mendekripsi pesan tersebut. Bagian terakhir, Output, menjelaskan hasil yang dihasilkan oleh sistem setelah proses enkripsi atau dekripsi dilakukan. Output dari proses enkripsi meliputi Kunci 2 (*Key 2*) yang dihasilkan untuk digunakan dalam dekripsi di kemudian hari, serta gambar stego (*Stego Image*) yang berisi pesan terenkripsi. Untuk proses dekripsi, output yang dihasilkan adalah gambar pesan (Message Image) yang telah didekripsi dari *stego image*.

Penjelasan ini memberikan gambaran menyeluruh tentang struktur dan fungsionalitas sistem, serta alur interaksi yang dilakukan oleh pengguna untuk mencapai tujuan enkripsi dan dekripsi. Diagram Use Case pada Gambar 3.7 akan menggambarkan secara visual bagaimana semua elemen ini berinteraksi.

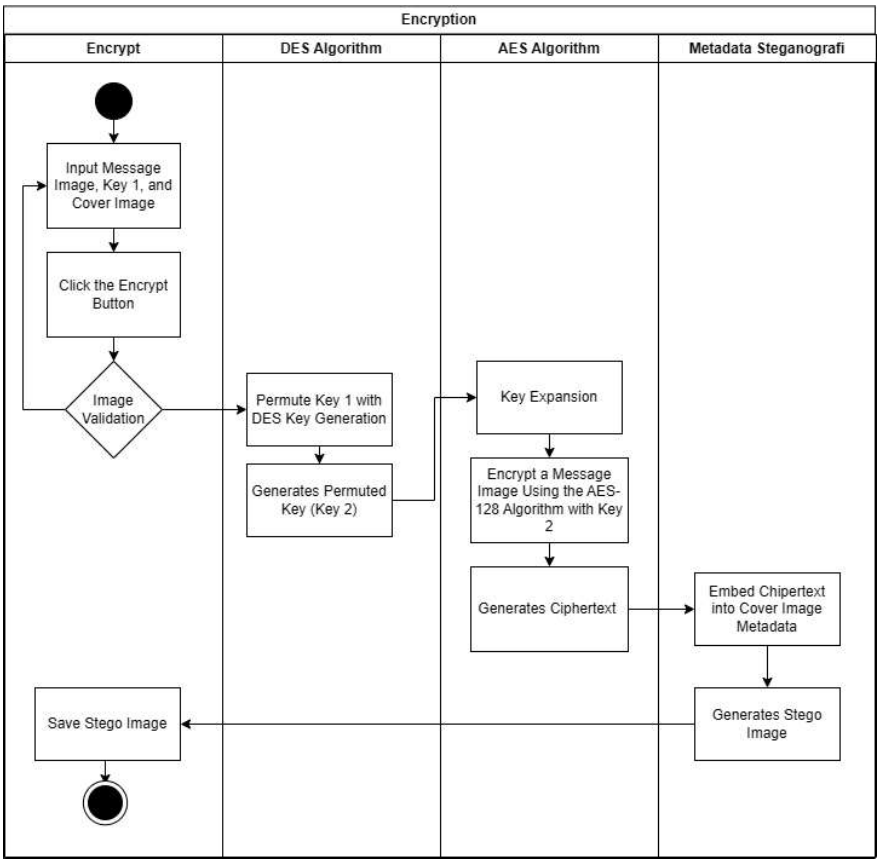


Gambar 3.7 Diagram *Use Case*

3.3.5. Diagram Aktivitas

Diagram aktivitas digunakan untuk memodelkan alur kerja atau proses yang ada dalam aplikasi yang dikembangkan. Diagram pada Gambar 3.8 mengilustrasikan alur lengkap dari proses enkripsi gambar pesan menggunakan algoritma DES dan AES, serta teknik steganografi untuk penyisipan data. Proses dimulai dengan input dari pengguna, yang mencakup gambar pesan (*Message Image*), kunci pertama (*Key 1*), dan gambar sampul (*Cover Image*). Setelah semua input diberikan, pengguna menekan tombol enkripsi (*Encrypt Button*), yang memicu validasi gambar untuk memastikan bahwa data yang diberikan memenuhi persyaratan sistem.

Setelah validasi berhasil, diagram menunjukkan proses permutasi *Key 1* menggunakan algoritma DES untuk menghasilkan kunci kedua (*Key 2*). *Key 2* ini kemudian digunakan dalam tahap enkripsi menggunakan algoritma AES-128. Proses enkripsi ini melibatkan ekspansi kunci (*Key Expansion*) dan penerapan AES-128 untuk mengenkripsi gambar pesan yang menghasilkan *ciphertext*. Selanjutnya, *ciphertext* ini disisipkan ke dalam metadata gambar sampul melalui proses steganografi, dan akan menghasilkan gambar stego (*Stego Image*). Gambar stego ini kemudian disimpan oleh sistem untuk menyelesaikan proses enkripsi. Diagram ini secara keseluruhan menggambarkan setiap langkah penting dalam proses enkripsi, dari input hingga output akhir. Diagram aktivitas enkripsi secara keseluruan dapat dilihat pada Gambar 3.8.

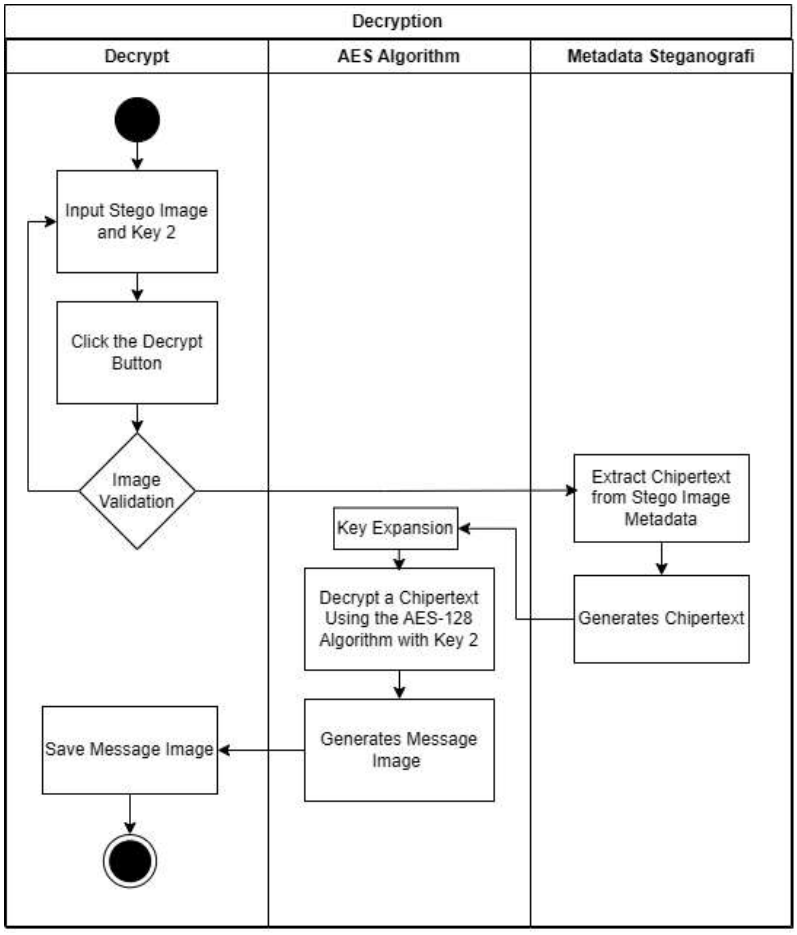


Gambar 3.8 Diagram Aktivitas Enkripsi

Gambar 3.9 Menunjukkan diagram aktivitas yang mengilustrasikan alur proses dekripsi pesan yang sebelumnya telah dienkripsi dan disisipkan ke dalam metadata gambar menggunakan teknik steganografi. Proses dimulai dengan pengguna memasukkan gambar stego (*Stego Image*) yang berisi pesan tersembunyi, serta Kunci 2 (*Key 2*) yang diperlukan untuk dekripsi. Pengguna kemudian menekan tombol dekripsi (*Decrypt Button*) untuk memulai proses. Langkah pertama adalah validasi gambar untuk memastikan input yang diberikan sesuai dengan persyaratan sistem.

Setelah validasi berhasil, proses dilanjutkan dengan ekstraksi *ciphertext* dari metadata gambar stego. Ciphertext yang telah diekstraksi kemudian diproses melalui ekspansi kunci (*Key Expansion*) menggunakan algoritma AES-128 dengan Kunci 2. Proses ini menghasilkan gambar pesan (*Message Image*) dengan mendekripsi *ciphertext* tersebut. Gambar pesan yang telah dipulihkan ini kemudian disimpan oleh sistem, menyelesaikan seluruh proses dekripsi yang digambarkan dalam diagram ini. Diagram ini menunjukkan setiap langkah penting dalam proses dekripsi, mulai dari penerimaan input hingga pemulihan dan penyimpanan pesan

gambar. Diagram aktivitas dekripsi secara keseluruan dapat dilihat pada Gambar 3.9.



Gambar 3.9 Diagram Aktivitas Dekripsi

3.4. Pengembangan

Tahap ketiga dalam penelitian ini adalah tahap pengembangan artefak atau aplikasi. Pada bagian ini, penulis akan menjelaskan metode yang digunakan dalam pengembangan aplikasi serta alat dan bahan yang digunakan selama proses pengembangan.

3.4.1. Alat dan Bahan

a) Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah Laptop dengan spesifikasi prosesor AMD Ryzen 3 7320U dengan Radeon Graphics, RAM 8 GB, dan ruang penyimpanan sebesar 512 GB.

b) Perangkat Lunak (*Software*)

Adapun perangkat lunak yang digunakan dalam penelitian ini antara lain:

1. Microsoft Windows 11
2. Intellij IDEA
3. Google Chrome
4. Draw.io
5. MATLAB

3.4.2. Metode Pengembangan Aplikasi

Penulis menggunakan metode *Agile* dalam proses pengembangan aplikasi web yang menerapkan algoritma DES dan AES serta teknik Steganografi Metadata untuk mengamankan data pada gambar. Metode *Agile* dipilih sebagai kerangka kerja karena kemampuannya untuk memberikan fleksibilitas yang tinggi. *Agile* memungkinkan tim pengembang untuk kembali ke tahap sebelumnya apabila diperlukan perubahan, yang merupakan keunggulan signifikan. Fleksibilitas ini menjadikan *Agile* sebagai pilihan yang sangat efektif dalam pengembangan perangkat lunak, terutama dalam konteks di mana kebutuhan dan kondisi proyek sering berubah selama proses pengembangan (Afriyantari & Putri, 2019). Metode *Agile* mengatur pengembangan melalui serangkaian iterasi singkat yang dikenal sebagai sprint. Setiap sprint menghasilkan peningkatan atau tambahan fitur yang kemudian diuji dan dievaluasi. Terdapat enam proses iterasi dalam metode pengembangan *Agile*, diantaranya yaitu *plan, design, develop, test, deploy*, dan *review*. Gambar 3.10 menunjukkan ilustrasi dan tahapan iterasi secara lengkap dalam proses pengembangan aplikasi dengan metode *Agile*.

Gambar 3.10 Metode Pengembangan Sistem *Agile*

1. ***Plan***

Pada tahap perencanaan, dimulai dengan mengidentifikasi dan mendefinisikan tujuan serta kebutuhan aplikasi yang akan dikembangkan. Tujuan utama dalam proyek ini adalah mengembangkan aplikasi web yang mengintegrasikan algoritma enkripsi DES dan AES bersama dengan teknik steganografi pada metadata gambar untuk mengamankan data gambar. Langkah awal termasuk mengidentifikasi fitur-fitur yang perlu diimplementasikan dalam aplikasi, termasuk antarmuka pengguna untuk memasukkan gambar pesan, gambar sampul, dan kunci enkripsi.

2. ***Design***

Selanjutnya, pada tahap desain dalam metodologi *Agile*, fokus diberikan pada pembuatan rancangan awal aplikasi yang akan dikembangkan. Dalam proyek ini, beberapa diagram disusun, termasuk, diagram alir, diagram use case, dan diagram aktivitas untuk setiap modul utama seperti enkripsi DES dan AES serta steganografi pada metadata gambar.

3. ***Develop***

Setelah menyelesaikan tahap desain, proyek bergerak ke tahap pengembangan, yang merupakan inti dari metodologi Agile. Di tahap ini, mulai dibangun aplikasi berdasarkan desain yang telah disusun sebelumnya. Pengembangan aplikasi dilaksanakan dalam iterasi atau sprint, yang relatif lebih panjang dibandingkan dengan tahapan lain dalam metodologi pengembangan *Agile*. Setiap iterasi atau *sprint* difokuskan pada pengembangan fitur tertentu yang telah ditetapkan sebelumnya. Iterasi pertama berfokus pada implementasi algoritma enkripsi DES dan AES. Setelah berhasil mengimplementasikan algoritma enkripsi tersebut, fokus iterasi berikutnya adalah integrasi modul steganografi pada metadata gambar.

4. ***Test***

Setiap iterasi atau sprint dalam proyek ini diakhiri dengan tahap pengujian. Pengujian dilakukan secara berkelanjutan di setiap iterasi untuk memastikan bahwa semua modul dan fitur yang telah dikembangkan berfungsi dengan baik dan sesuai dengan kebutuhan yang telah ditetapkan, serta untuk memperbaiki kesalahan secepat mungkin. Dalam proyek ini, metode pengujian yang digunakan adalah

pengujian *Black Box*, yang berfokus pada validasi *input* dan *output*. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa semua fungsionalitas fitur pada aplikasi berjalan sesuai spesifikasi dan memastikan juga bahwa tidak ada *bug* atau masalah yang terlewatkan.

5. Deploy

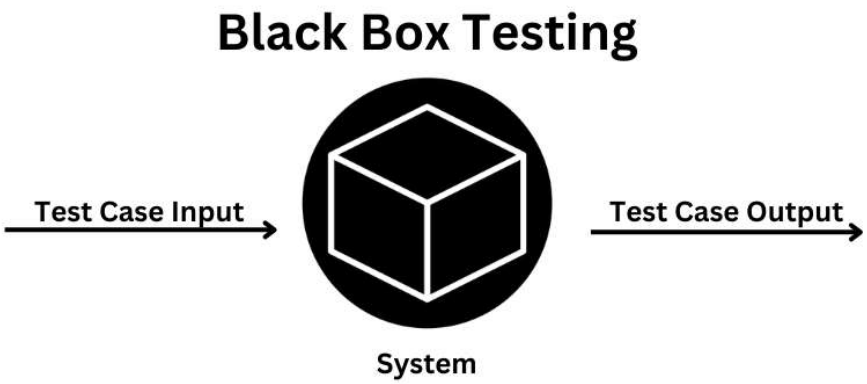
Tahap *deploy* dalam konteks skripsi ini tidak melibatkan penerbitan aplikasi ke publik umum melalui marketplace atau melalui internet, mengingat aplikasi yang dikembangkan bersifat non-terkoneksi ke internet. Sebagai pengganti, *deployment* dilakukan dengan mengunggah kode sumber aplikasi ke platform repositori online seperti GitHub. Langkah ini memungkinkan aplikasi yang telah dikembangkan dapat diakses oleh pengguna atau pengembang lain yang berkeinginan untuk menggunakan atau mengembangkan lebih lanjut aplikasi tersebut

6. Review

Pada tahap akhir metodologi *Agile*, tahap *review* dijalankan. Evaluasi dilakukan terhadap hasil dari setiap iterasi, dengan mendapatkan umpan balik baik dari pengembang aplikasi. Umpan balik ini digunakan untuk mengidentifikasi aspek-aspek apa saja yang perlu diperbaiki dan ditingkatkan dari fitur yang telah dikembangkan. Proses ini memastikan bahwa ada kesempatan berkelanjutan untuk mendapatkan masukan dan saran dari proses pengembangan aplikasi yang telah dilakukan, sehingga memungkinkan peningkatan terus-menerus.

3.5. Pengujian dan Evaluasi

Setelah fase pengembangan, langkah selanjutnya adalah melakukan pengujian dan evaluasi terhadap aplikasi yang telah dibangun. Tahap ini sangat penting untuk memastikan bahwa aplikasi yang dikembangkan sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan sebelumnya. Dalam proses ini, metode pengujian *black box*, atau dikenal juga sebagai *behavioral testing*, digunakan untuk fokus pada persyaratan fungsional fitur perangkat lunak. Pengujian *black box* dirancang untuk mengidentifikasi berbagai jenis kesalahan, termasuk fungsi yang salah atau tidak ada, kesalahan antarmuka, kesalahan dalam struktur data, kesalahan perilaku atau kinerja aplikasi, dan kesalahan pada inisialisasi serta terminasi aplikasi. Untuk melihat gambaran bagaimana metode pengujian *black box* bekerja dapat dilihat pada Gambar 3.11.



Gambar 3.11 Cara Kerja Metode Pengujian *Black Box*

Pengujian ini dilakukan menggunakan perangkat Lenovo Ideapad Slim 1 dengan spesifikasi tertentu yang mencakup prosesor AMD Ryzen 3 7320U, RAM 8 GB, dan sistem operasi Windows 11. Spesifikasi ini penting karena berpengaruh terhadap performa aplikasi, terutama dalam hal kecepatan proses enkripsi, dekripsi, penyisipan, dan ekstraksi pesan. Proses pengujian dimulai dengan menguji algoritma DES untuk memastikan keberhasilannya dalam permutasi kunci. Algoritma enkripsi AES juga dilakukan pengujian untuk memastikan fungsinya berhasil diimplementasikan. Pengujian dilakukan dengan mengenkripsi gambar menggunakan dimensi yang berbeda dan dengan kunci yang berbeda-beda.

Selanjutnya, steganografi metadata diuji dengan menyisipkan pesan terenkripsi ke dalam gambar. Pengujian juga dilakukan untuk memastikan bahwa kualitas gambar tetap terjaga dan perubahan pada gambar tidak terlihat oleh mata manusia. Untuk mengukur kualitas gambar setelah proses steganografi, metrik Mean Squared Error (MSE) dan Peak Signal-to-Noise Ratio (PNSR) digunakan. MSE mengukur perbedaan rata-rata kuadrat antara piksel gambar asli dan piksel hasil steganografi, sementara PNSR mengukur kualitas rekonstruksi gambar setelah penyisipan pesan, dengan nilai PNSR yang tinggi menunjukkan kualitas gambar yang baik.

Evaluasi dilakukan setelah semua pengujian selesai untuk menganalisis hasil pengujian dan memastikan bahwa aplikasi telah memenuhi semua spesifikasi dan kebutuhan yang ditetapkan pada tahap perencanaan dan desain. Tahap evaluasi ini juga membantu mengidentifikasi aspek-aspek aplikasi yang perlu ditingkatkan.

3.6. Pelaporan

Setelah fase pengujian dan evaluasi aplikasi selesai, langkah berikutnya adalah pelaporan. Tahap ini melibatkan penyusunan laporan hasil penelitian yang akan diformulasikan dalam bentuk penulisan skripsi. Melalui proses pelaporan ini, semua hasil dari penelitian yang telah dilakukan dapat terdokumentasi secara rinci, memastikan bahwa temuan dan metode yang diaplikasikan dalam penelitian ini tercatat dengan jelas. Hal ini diharapkan tidak hanya akan berkontribusi pada peningkatan pengetahuan dalam bidang keamanan data tetapi juga akan menyediakan dasar yang kuat bagi penelitian mendatang dalam bidang yang sama. Dengan demikian, laporan skripsi ini akan menjadi sumber referensi yang berharga untuk studi-studi selanjutnya yang mengeksplorasi berbagai aspek keamanan data, memungkinkan peneliti lain untuk membangun dan memperluas penelitian ini dengan memanfaatkan metodologi atau menanggapi temuan yang telah dijelaskan.