BAB III
METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode *Design and Development* (D&D). Metode penelitian *Design and Development* didefinisikan oleh Richey dan Klein (2007) sebagai, "*the systematic study of design, development, and evaluation processes with the aim of establishing an empirical basis for the creation of instructional and non-instructional product and tools and new or enhanced models that govern their development"* (Richey & Klein, 2007). Metode D&D dipilih karena menyediakan kerangka kerja yang sistematis untuk merancang, mengembangkan, dan mengevaluasi solusi praktis. D&D 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. Metodologi ini cocok untuk penelitian yang bertujuan untuk mengembangkan produk atau aplikasi, seperti aplikasi Android untuk keamanan data teks menggunakan kriptografi *Serpent* dan steganografi *Least Significant Bit* (LSB), untuk melihat bagaimana alur penelitian dengan metode *Design and Development* dapat dilihat pada Gambar 3.1.

Gambar 3. 1 Alur Penelitian Metode *Design and Development* (Diadaptasi dari J. Ellis & Levy, 2010)

3.1 Analisis

Tahap pertama dalam metode penelitian D&D adalah analisis, tahap ini dilakukan dengan melakukan studi literatur terhadap topik penelitian yang akan diambil. Dengan melakukan studi literatur yang relevan akan memberikan informasi seputar perkembangan ilmu dan hasil penelitian terdahulu. Tahapan ini penulis menentukan identifikasi masalah yang ada, rumusan masalah, serta tujuan penelitian, selain itu memahami juga berbagai pendekatan dan solusi yang telah diuji sebelumnya. Dalam konteks penelitian ini, penulis menemukan ide untuk mengembangkan aplikasi berbasis Android guna mengamankan data teks di dengan menggunakan algoritma kriptografi *Serpent* dan teknik steganografi LSB. Penulis melakukan studi literatur mengenai topik penelitian algoritma enkripsi *Serpent*

untuk mengamankan data teks, teknik steganografi LSB untuk menyembunyikan sebuah pesan pada gambar, dan pengembangan aplikasi Android.

3.2 Desain atau Perancangan Sistem

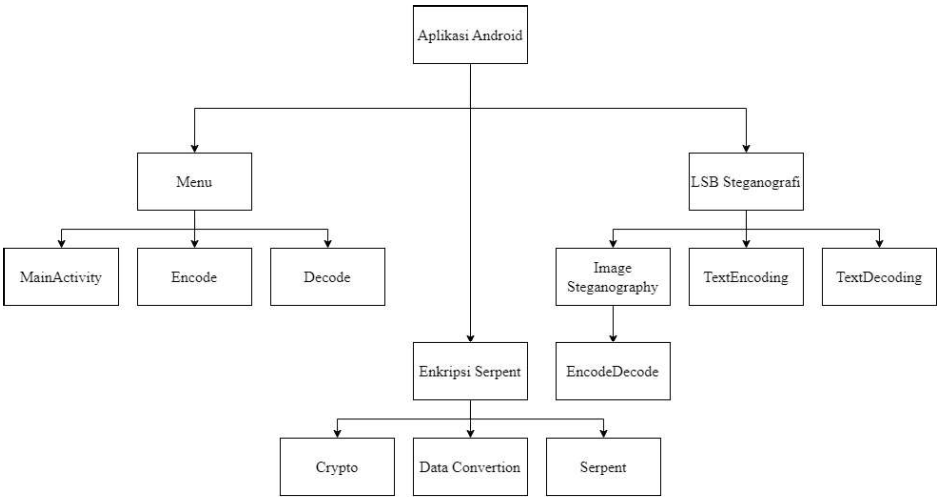
Tahap kedua adalah tahap desain atau perancangan sistem. Pada tahap ini, penulis mengembangkan kerangka konseptual dalam proses desain aplikasi. Proses perancangan sistem meliputi pembuatan beberapa diagram untuk memvisualisasikan arsitektur sistem, diagram kasus penggunaan, diagram kelas, diagram aktivitas.

3.2.1 Diagram Arsitektur

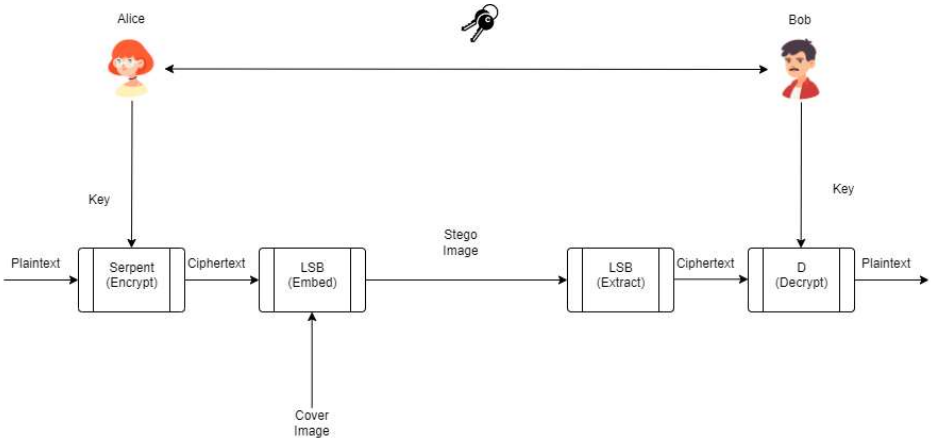
Diagram arsitektur berikut menggambarkan struktur sistem aplikasi Android yang akan dikembangkan secara keseluruhan. Gambar 3.2 menunjukkan diagram dengan komponen-komponen utama pada aplikasi dan bagaimana komponen tersebut berinteraksi satu sama lain. Terdapat tiga modul utama yaitu menu, enkripsi *Serpent*, dan LSB steganografi. Setiap modul komponen dalam aplikasi ini di desain untuk memiliki peran spesifik dalam mendukung fungsi utama aplikasi. Sebagai contoh, di dalam modul *Menu* terdapat *MainActivity,* komponen tersebut mengelola tampilan dan navigasi. Dalam modul enkripsi *Serpent* terdapat komponen *Data Convertion* yang memiliki fungsi untuk mengonversi data-data yang diproses oleh komponen *Serpent*, yang mana komponen *Serpent* berfungsi sebagai proses enkripsi pesan teks. Lalu dalam modul LSB Steganografi terdapat komponen *ImageSteganography* yang memiliki fungsi untuk melakukan teknik steganografi LSB untuk memastikan bahwa pesan dapat disembunyikan dan dapat di ekstrak dari gambar.

Selain itu pada Gambar 3.3 menunjukkan diagram arsitektur sistem yang menggambarkan alur dari seluruh proses untuk mengirimkan pesan rahasia melalui aplikasi yang akan dikembangkan dengan menggunakan kombinasi enkripsi *Serpent* dan juga teknik steganografi LSB. Proses diawali oleh pengguna pertama (Alice) yang akan melakukan enkripsi pesan teks dengan menggunakan algoritma *Serpent*, Alice memasukkan pesan dan juga kunci yang disetujui oleh Bob untuk proses enkripsi dan dekripsi pesan. Proses ini

akan menghasilkan *ciphertext*, *ciphertext* ini kemudian disisipkan ke dalam media gambar yang sudah di masukan sebelumnya oleh Alice dengan menggunakan teknik Steganografi LSB lalu menghasilkan *stego image*. *Stego image* ini lalu dikirimkan ke pengguna kedua (Bob) melalui aplikasi komunikasi melalui internet. Setelah menerima *stego image*, Bob akan melakukan proses ekstraksi data dari *stego image* untuk mengambil *ciphertext* dengan menggunakan teknik steganografi LSB. Setelah *ciphertext*-nya di dapatkan, Bob selanjutnya melakukan dekripsi pesan *ciphertext* dengan menggunakan kunci yang sama pada proses enkripsi untuk mendapatkan kembali pesan teks yang asli.



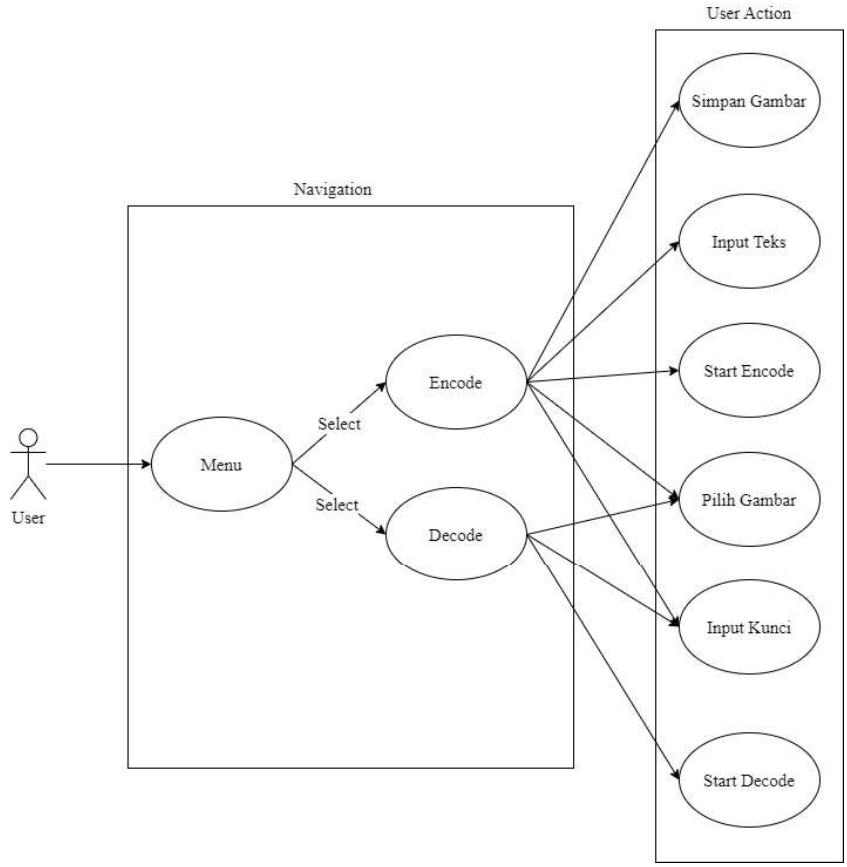
Gambar 3. 2 Diagram Arsitektur Aplikasi Android



Gambar 3. 3 Diagram Arsitektur Sistem Enkripsi dan Steganografi

3.2.2 Diagram Kasus Penggunaan

Gambar 3.4 menunjukkan gambar diagram kasus penggunaan yang menggambarkan bagaimana interaksi antara pengguna dan sistem aplikasi Android yang dikembangkan. Penulis membuat diagram kasus penggunaan untuk menunjukkan berbagai fungsi dalam aplikasi yang dapat dilakukan oleh pengguna, seperti fungsi *encoding* dan *decoding* pesan teks ke dalam gambar.



Gambar 3. 4 Diagram Kasus Penggunaan Aplikasi Android

Diagram kasus penggunaan ini dapat menggambarkan skenario pengguna saat menggunakan aplikasi. Ketika pengguna memilih opsi *Encode* pada menu utama, pengguna akan diarahkan ke dalam menu *Encode*, dimana pengguna dapat memasukkan pesan teks, kunci, dan gambar yang akan dijadikan *cover* untuk proses *encoding*. Setelah pesan teks, kunci dan gambar dimasukkan pengguna dapat memulai proses *encoding.* Ketika proses *encoding* selesai dilakukan maka pengguna dapat menyimpan gambar yang sudah disisipkan.

Sedangkan ketika pengguna memilih opsi *Decode* pada menu utama, pengguna akan diarahkan ke halaman *decode* dimana pengguna dapat memasukkan gambar yang sudah disisipkan pesan selain itu masukan juga kunci yang digunakan pada saat proses *encoding*. Setelah gambar dipilih dan kunci dimasukkan, pengguna dapat memulai proses *decoding* untuk mengekstrak data teks.

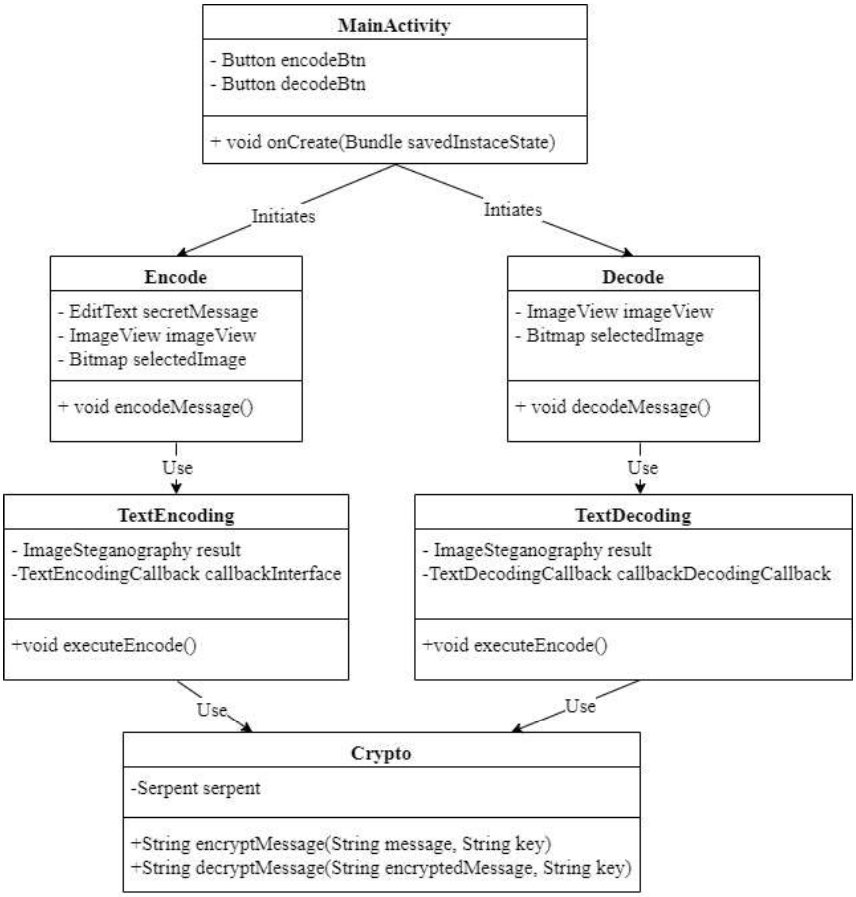
3.2.3 Diagram Kelas

Penulis membuat diagram kelas yang akan memberikan gambaran yang jelas mengenai struktur aplikasi secara menyeluruh, menunjukkan berbagai kelas yang digunakan dan juga hubungan antara mereka. Diagram kelas ini dapat dibagi menjadi tiga bagian, pertama menunjukkan komponen yang terkait dengan aktivitas utama dalam aplikasi Android yang dapat dilihat pada Gambar 3.5, pada bagian ini '*MainActivy*' berinteraksi langsung dengan pengguna melalui antarmuka dan bertindak juga sebagai kontrol utama yang melakukan inisiasi kepada dua proses '*Encode*' dan '*Decode*'. Proses *Encode* dan *Decode* bergantung dan menggunakan kelas '*TextEncoding*' dan '*TextDecoding*' untuk mengelola proses penyisipan dan ekstrak data teks dan menggunakan kelas '*Crypto*' melakukan proses enkripsi dan dekripsi data teks menggunakan algoritma kriptografi *Serpent*.

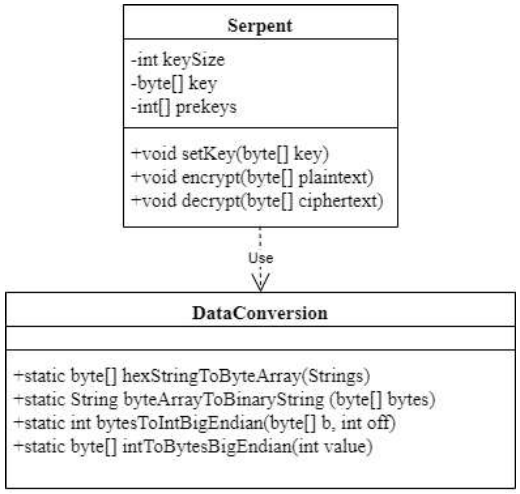
Bagian kedua menggambarkan komponen yang terlibat dalam proses enkripsi dan dekripsi data menggunakan algoritma enkripsi *Serpent* yang dapat dilihat lebih detail pada Gambar 3.6, kelas '*Serpent*' bertanggung jawab untuk mengimplementasikan algoritma enkripsi dan dekripsi, kelas ini dipanggil di dalam kelas *'Crypto'*. Sedangkan kelas '*DataConversion*' digunakan pada kelas '*Serpent*' untuk segala proses mengkonversi data ke fortmat yang sesuai untuk diproses oleh algoritma. Kedua kelas ini adalah inti dari proses pengamanan data dalam aplikasi yang memastikan bahwa data teks berhasil dienkripsi sebelum selanjutnya disembunyikan dalam gambar.

Bagian terakhir menggambarkan komponen yang terlibat dalam proses *encoding* dan *decoding* data teks ke dalam gambar dengan menggunakan teknik steganografi LSB yang dapat dilihat pada Gambar 3.7. Kelas '*ImageSteganography*' memiliki peran untuk mengelola keseluruhan proses

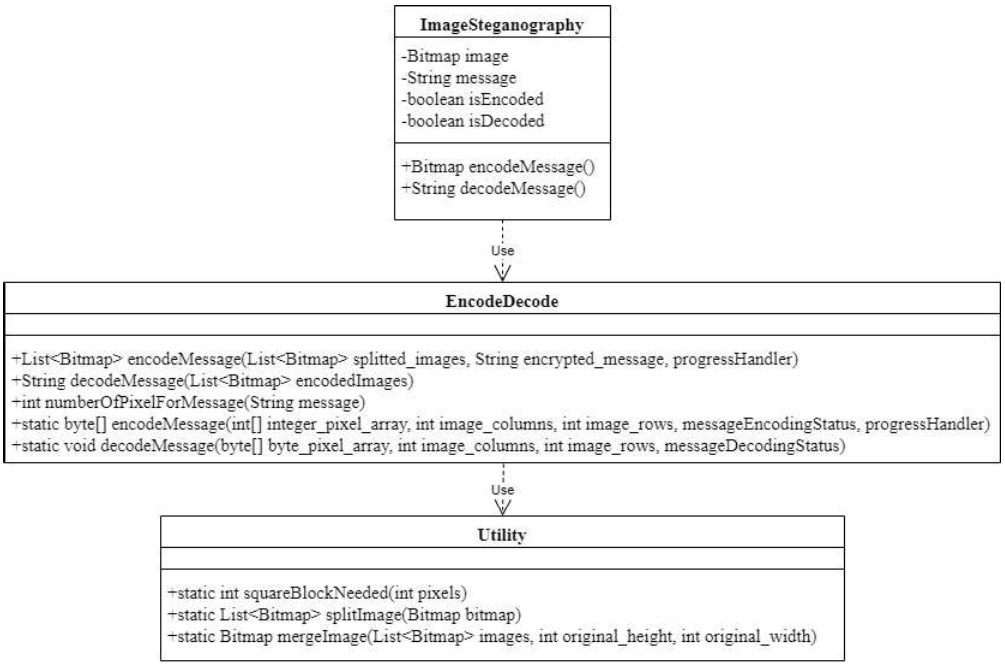
penyisipan gambar dengan steganografi LSB, dengan menggunakan kelas '*EncodeDecode*' yang melakukan penyisipan dan ekstrak data dari gambar. Kelas '*Utility*' digunakan untuk menyediakan fungsi-fungsi tambahan yang mendukung proses penyisipan dan ekstrak data.



Gambar 3. 5 Diagram Kelas Aktivitas Utama



Gambar 3. 6 Diagram Kelas Algoritma Enkripsi *Serpent*

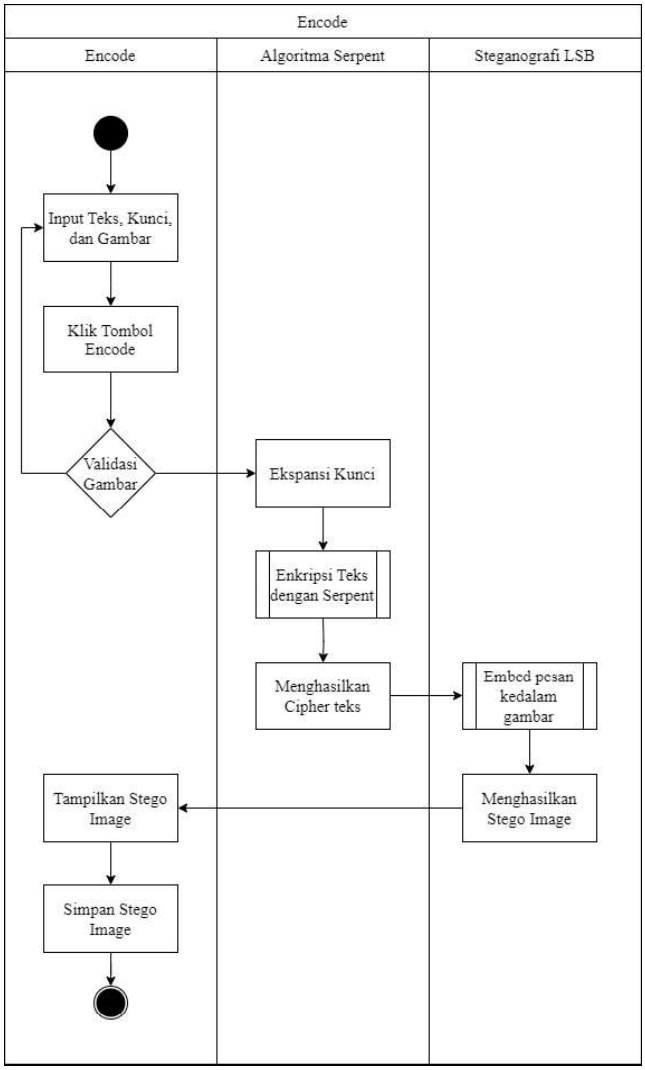


Gambar 3. 7 Diagram Kelas Steganografi LSB

3.2.4 Diagram Aktivitas

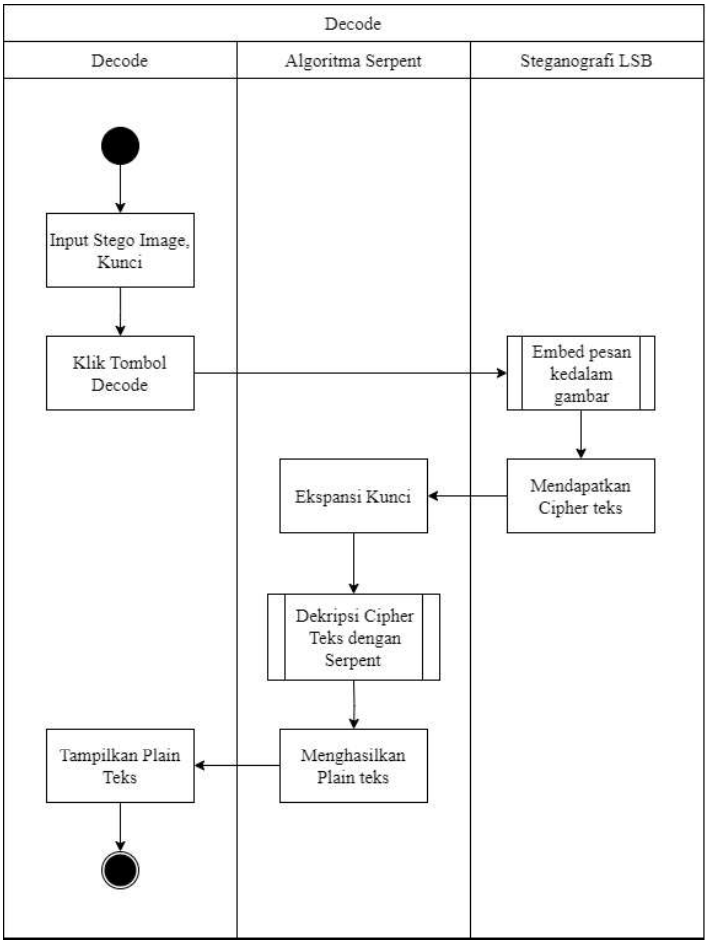
Diagram aktivitas digunakan untuk memodelkan alur kerja proses dalam aplikasi Android yang dikembangkan. Diagram aktivitas dapat membantu dalam mendefinisikan sebuah alur logika pada aplikasi, interaksi antara komponen, serta respons terhadap aksi pengguna. Gambar 3.8 menunjukkan diagram aktivitas pada proses *encoding* pada aplikasi, diagram ini dapat dibagi menjadi tiga bagian utama yaitu *encode*, algoritma *Serpent*, dan steganografi

LSB. Pada bagian ini, pengguna akan memasukkan teks, kunci, dan gambar, lalu menekan tombol *encode*. Setelah menekan tombol *encode*, gambar dan teks akan di validasi bahwa ukuran panjang teks dapat ditampung dengan ukuran gambar. Jika valid, proses akan berlanjut ke algoritma *Serpent* untuk melakukan enkripsi. Setelah dilakukan enkripsi maka akan menghasilkan *ciphertext*. Kemudian *ciphertext* akan dikirim kan ke bagian steganografi LSB untuk disisipkan ke dalam gambar yang akan menghasilkan *stego image*. *Stego image* tersebut akan ditampilkan ke halaman menu kembali dan pengguna dapat menyimpan gambar tersebut untuk dikirim kan gambar yang sudah disisipkan pesan tersembunyi.



Gambar 3. 8 Diagram Aktivitas *Encode*

Gambar 3.9 menunjukkan diagram aktivitas untuk proses *decode* pada aplikasi. Diagram ini juga terdiri dari tiga bagian utama yang serupa dengan proses *encode*, hanya saja untuk proses *decode* tahapan yang pertama dilalui adalah tahap steganografi LSB dahulu untuk mendapatkan pesan yang disisipkan. Setelah itu, melalui tahap dekripsi dengan algoritma *Serpent* yang akan menghasilkan *plaintext*. *Plaintext* yang dihasilkan akan ditampilkan oleh sistem kepada pengguna.



Gambar 3. 9 Diagram Aktivitas *Decode*

3.3 Pengembangan

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

3.3.1. Alat dan Bahan

Dalam penelitian ini, alat penelitian merupakan perangkat keras dan perangkat lunak yang digunakan. Adapun perangkat keras yang digunakan selama pengembangan aplikasi adalah sebagai berikut:

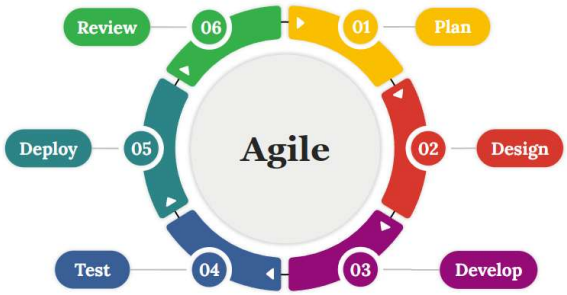
1. Prosesor Intel Core i7-10750H
2. RAM 16GB DDR 4
3. NVIDIA GeForce GTX 1650 Ti
4. 750 GB NVME SSD

Selain perangkat keras, terdapat beberapa perangkat lunak yang digunakan selama pengembangan aplikasi ini. Berikut adalah perangkat lunak yang digunakan:

1. Microsoft Windows 11
2. Android Studio
3. Visual Studio Code
4. Draw.io

3.3.2. Metode Pengembangan Aplikasi

Penulis menggunakan metode *Agile* untuk proses pengembangan aplikasi Android yang mengimplementasikan algoritma enkripsi *Serpent* dan steganografi LSB guna mengamankan data teks. Metode *Agile* digunakan karena fleksibilitasnya yang tinggi dalam menghadapi perubahan kebutuhan, sehingga pengembangan bisa kembali ke fase yang lebih awal apabila terdapat perubahan yang diperlukan (D. A. P. Putri, 2019). Dengan menggunakan metode pengembangan aplikasi *Agile*, pengembangan aplikasi dilakukan melalui iterasi pendek yang disebut *sprint*, di mana setiap *sprint* menghasilkan peningkatan atau penambahan fitur yang dapat di uji dan dievaluasi. Pada metode pengembangan sistem aplikasi *Agile*, terdapat enam proses iterasi di dalamnya seperti *plan*, *design*, *develop*, *test*, *deploy*, dan *review*. Gambar 3.10 menunjukkan ilustrasi dan tahapan iterasi secara lengkap dalam proses pengembangan aplikasi Android.



Gambar 3. 10 Metode Pengembangan Sistem *Agile* (Diadaptasi dari Ramadhan & Angelia, 2023)

1. *Plan*

Pada tahap perancangan atau *plan*, penulis memulai dengan mengidentifikasi dan mendefinisikan tujuan dan kebutuhan aplikasi yang akan dikembangkan. Pada penelitian ini, tujuan utamanya adalah untuk mengembangkan aplikasi Android yang mengimplementasikan algoritma enkripsi *Serpent* dan teknik steganografi LSB untuk mengamankan data teks. Penulis pertama mengidentifikasi fitur apa saja yang perlu diimplementasikan dalam aplikasi, seperti antarmuka aplikasi untuk pengguna memasukkan teks, gambar, dan kunci enkripsi.

2. *Design*

Selanjutnya pada tahap *design* dalam *Agile* penulis berfokus pada pembuatan rancangan awal aplikasi yang akan dikembangkan. Pada penelitian ini, penulis membuat beberapa diagram seperti, diagram arsitektur, diagram kasus penggunaan, diagram aktivitas, dan diagram kelas untuk desain proses aplikasi dan spesifikasi aplikasi pada setiap modul utama seperti enkripsi *Serpent* dan steganografi LSB.

3. *Develop*

Setelah tahap iterasi *design* selesai, maka masuk ke dalam tahapan proses iterasi *develop*. Tahap *develop* merupakan inti dari metode *Agile*, dimana penulis mulai membangun aplikasi berdasarkan desain yang sebelumnya telah dibuat. Pengembangan aplikasi dilakukan dalam iterasi atau *sprint* yang cukup lama dibandingkan tahap proses metode pengembangan aplikasi *Agile* lainnya. Setiap iterasi atau *sprint* difokuskan pada penyelesaian fitur tertentu yang telah ditentukan. Iterasi pertama berfokus pada implementasi algoritma enkripsi

Serpent. Lalu iterasi selanjutnya setelah algoritma enkripsi *Serpent* berhasil diimplementasikan, berfokus pada integrasi modul steganografi LSB.

4. *Test*

Setiap iterasi atau *sprint* yang dilakukan diakhiri dengan tahap *test* atau pengujian. Penulis melakukan pengujian secara berkelanjutan setiap iterasi untuk memastikan bahwa setiap modul dan fitur yang telah dikembangkan berfungsi dengan baik dan sesuai kebutuhan yang sudah ditetapkan sebelumnya dan memperbaiki kesalahan secepat mungkin . Pada penelitian ini, pengujian yang dilakukan adalah dengan metode pengujian *Black Box.* Pengujian ini fokus pada validasi *input* dan *output* untuk memastikan bahwa seluruh fungsionalitas fitur pada aplikasi berfungsi sesuai dengan spesifikasi dan memastikan juga bahwa tidak ada *bug* atau masalah yang terlewatkan.

5. *Deploy*

Tahap *deploy* dalam konteks penelitian ini tidak melakukan publikasi aplikasi ke publik luas melalui *marketplace* atau *deploy* komponen aplikasi melalui internet, mengingat aplikasi yang dikembangkan tidak terkoneksi ke internet. Jadi sebagai gantinya, penulis melakukan *deployment* dengan mengunggah *source code* aplikasi dan aplikasi ke platform repositori *online* seperti GitHub. Dengan ini aplikasi yang di kembangkan dapat diakses secara luas oleh pengguna atau *developer* yang ingin menggunakan atau mengembangkan aplikasi ini lebih lanjut.

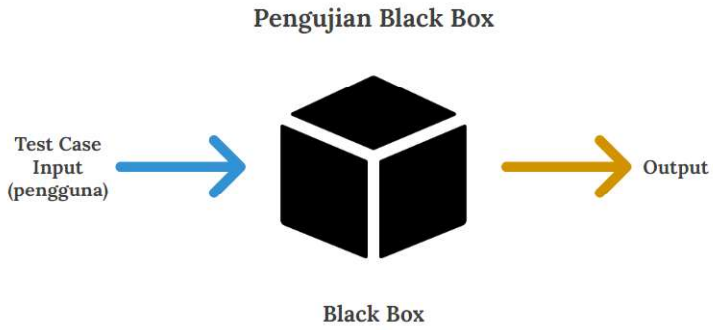
6. *Review*

Pada tahap terakhir *Agile*, tahap *review* dilakukan. Penulis melakukan evaluasi hasil dari setiap iterasi dan mendapatkan *feedback* dari pengguna atau dari pembuat aplikasi. *Feedback* digunakan untuk mendapatkan apa saja yang perlu di perbaiki dan ditingkatkan dari fitur yang sudah dikembangkan. Dengan ini, memungkinkan penulis untuk terus mendapatkan *feedback* dari proses pengembangan aplikasi yang sudah dibuat.

3.4 Pengujian dan Evaluasi

Tahap setelah melakukan pengembangan tentunya melakukan pengujian dan evaluasi terhadap artefak atau aplikasi yang disudah dibangun. Tahap ini merupakan langkah penting dalam memastikan bahwa aplikasi yang dikembangkan

sesuai dengan spesifikasi dan kebutuhan awal. Pada tahap ini, penulis menggunakan metode pengujian aplikasi *black box* atau biasa dikenal juga sebagai *behavioral testing*, pengujian ini berfokus pada persyaratan fungsional fitur pada perangkat lunak. Pengujian *black box* mencoba menemukan kesalahan dalam beberapa kategori, yaitu: (1) fungsi yang salah atau hilang, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses basis data eksternal, (4) kesalahan perilaku aplikasi atau kinerja aplikasi, (5) kesalahan inisialisasi dan terminasi (Pressman, 2010). Untuk melihat gambaran bagaimana metode pengujian *black box* bekerja dapat dilihat pada Gambar 3.10.



Gambar 3. 11 Cara Kerja Metode Pengujian Black Box (Diadaptasi dari Pressman, 2010)

Pengujian dilakukan pada satu buah perangkat Tecno Pova 4 Pro dengan spesifikasi sebagai berikut:

1. Prosesor MediaTek Helio G99, dengan arsitektur 8 *core* dan *clock speed* 2200 MHz
2. RAM 8 GB
3. Android 12

Spesifikasi perangkat gawai yang digunakan memiliki peran penting dalam performa aplikasi, terutama dalam hal waktu proses enkripsi, dekripsi, penyisipan pesan, dan ekstraksi pesan. Pengujian pada perangkat gawai lain mungkin akan bervariasi dari waktu proses enkripsi, dekripsi, dan proses lainnya.

Proses pengujian dimulai dengan melakukan pengujian terhadap algoritma enkripsi *Serpent* bekerja dengan baik. Penulis melakukan pengujian dengan melakukan enkripsi pesan teks dengan menggunakan kunci yang berbeda. Selain itu, penulis mengukur tingkat korelasi antara *plaintext* dan *ciphertext* menggunakan

perhitungan Korelasi Pearson untuk memastikan bahwa hasil *ciphertext* tidak memiliki pola yang serupa dengan *plaintext*. Lalu penulis menguji steganografi LSB dengan menyisipkan pesan terenkripsi ke dalam gambar. Penulis juga menguji untuk melihat apakah kualitas gambar tetap terjaga dan perubahan pada gambar tidak terlihat oleh mata manusia dengan menyisipkan pesan ke dalam gambar dengan ukuran kecil.

Untuk mengukur kualitas gambar setelah proses steganografi, penulis menggunakan metrik *Mean Squared Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PNSR). MSE digunakan untuk mengukur perbedaan rata-rata kuadrat antara piksel pada gambar asli dan piksel hasil steganografi. PNSR digunakan untuk mengukur kualitas rekonstruksi gambar setelah proses penyisipan pesan pada gambar, dengan nilai PNSR yang tinggi maka menunjukkan kualitas gambar semakin baik.

Evaluasi dilakukan setelah semua pengujian selesai. Penulis melakukan analisis dari hasil pengujian untuk memastikan bahwa aplikasi sudah memenuhi semua spesifikasi dan kebutuhan yang telah ditetapkan pada tahap *plan* dan *design*. Selain itu, tahap evaluasi juga dapat membantu mengidentifikasi bagian mana yang perlu perbaikan.

3.5 Pelaporan

Setelah melakukan pengujian artefak atau aplikasi dan melakukan evaluasi, tahap selanjutnya adalah pelaporan. Tahap ini mencakup penyusunan laporan hasil penelitian yang berbentuk penulisan skripsi. Dengan melakukan tahap pelaporan ini, hasil dari penelitian yang telah dilakukan dapat terdokumentasi dengan baik sehingga diharapkan memberikan kontribusi bagi perkembangan ilmu dalam bidang sekuritas data. Selain itu dapat menjadi referensi bagi penelitian lainnya yang meneliti keamanan data.