

BAB III

METODE PENELITIAN

3.1 Metode yang Digunakan

Metode yang digunakan oleh peneliti dalam penelitian ini adalah *Design Research Methodology* (DRM), yang memungkinkan peneliti untuk sistematis mengidentifikasi masalah, mengembangkan solusi, dan mengevaluasi efektivitas solusi tersebut. Alasan peneliti menggunakan metode DRM ini adalah kebutuhan akan penelitian yang dapat menghasilkan prototipe yang tidak hanya teoritis tetapi juga praktis dan aplikatif (Blessing & Chakrabarti, 2009).

Selanjutnya, penelitian ini mengintegrasikan pendekatan *System Development Life Cycle* (SDLC) untuk tahap pengembangan sistem, yang mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. SDLC dipilih karena kerangka kerja ini mempunyai struktur yang jelas untuk pengembangan sistem (Sommerville, 2016).

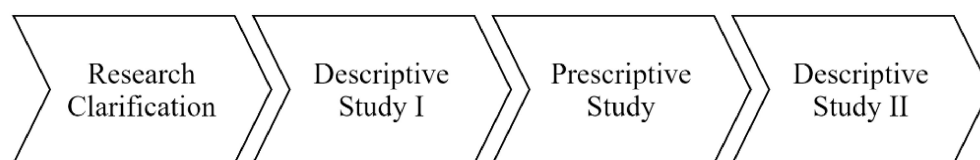
Proses pengujian sistem menggunakan metode *Black-Box Testing*, dikarenakan memastikan bahwa sistem yang dikembangkan berfungsi sesuai dengan rancangan yang telah ditetapkan tanpa perlu memahami struktur internal dari sistem. Pendekatan ini sangat efektif untuk mengevaluasi sistem dari perspektif pengguna, serta memastikan bahwa semua fungsi sistem operasional berfungsi dengan baik sebelum sistem sebar luaskan (Myers, Sandler, & Badgett, 2011).

Dengan menggabungkan metode DRM dengan SDLC, tesis ini bertujuan untuk tidak hanya mengembangkan sebuah sistem yang inovatif tetapi juga memperkaya literatur akademis. Hal ini diharapkan dapat memberikan kontribusi yang signifikan untuk keamanan data dan teknologi informasi, khususnya dalam aplikasi visualisasi data yang memerlukan tingkat keamanan yang tinggi.

3.2 Design Research Methodology (DRM)

Design Research Methodology (DRM) adalah pendekatan yang sistematis dan iteratif yang digunakan untuk memecahkan masalah desain melalui serangkaian langkah secara terstruktur. DRM dimulai dengan *Research Clarification*, di mana masalah penelitian diidentifikasi dan dianalisis untuk memastikan pemahaman yang mendalam tentang tantangan yang dihadapi. Tahap ini melibatkan pengumpulan data awal, melalui studi literatur yang ada yang relevan dengan

enkripsi AES dan steganografi (Blessing & Chakrabarti, 2009). Selanjutnya, *Descriptive Study I* mengembangkan pemahaman tentang konteks operasional sistem yaitu memodelkan interaksi antar variabel yang mempengaruhi masalah yang terjadi. Ini membantu dalam merumuskan spesifikasi yang dibutuhkan untuk sistem yang akan dikembangkan. Selanjutnya yaitu *Prescriptive Study* yang berfokus pada penciptaan rancangan solusi yang menggabungkan semua persyaratan dan parameter yang telah diidentifikasi pada tahap *Research Clarification*. Tahap ini sering membutuhkan pengembangan prototipe yang dapat diuji. Pada tahap ini peneliti menggunakan metode pengembangan SDLC untuk mengembangkan sistem yang dirancang. *Descriptive Study II* kemudian menerapkan dan menguji solusi dalam lingkungan nyata, mengumpulkan data kinerja, dan melakukan iterasi berdasarkan umpan balik (Blessing & Chakrabarti, 2009).

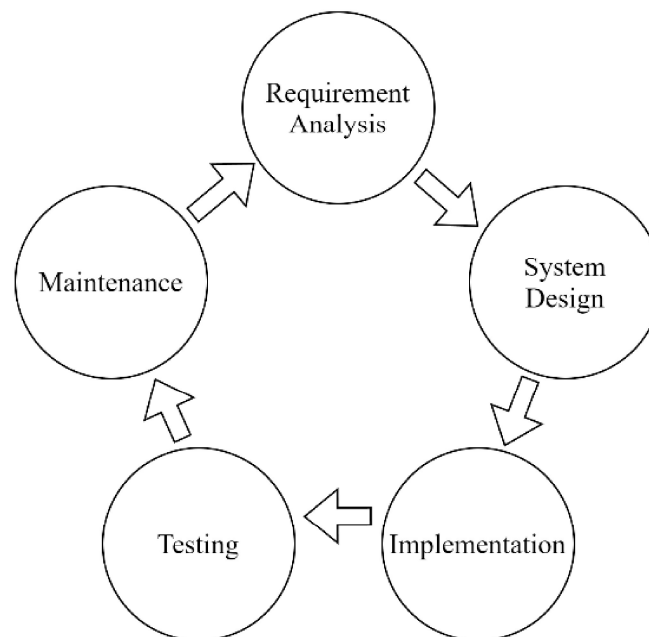


Gambar 3. 1 Tahapan Metode *Design Research Methodology*

3.3 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) adalah metodologi yang mapan dalam pengembangan sistem informasi yang menekankan pendekatan yang terstruktur dan metodis melalui berbagai fase pengembangan. SDLC dimulai dengan *Requirement Analysis*, di mana kebutuhan pengguna dan sistem dikumpulkan secara rinci. Ini memastikan bahwa semua aspek fungsional dan non-fungsional sistem dipahami sebelum pengembangan dimulai. Tahap ini penting dalam mendefinisikan ruang lingkup proyek dan menetapkan dasar untuk desain sistem yang efektif (Sommerville, 2016). Setelah kebutuhan dikumpulkan, fase *System Design* berlangsung di mana arsitektur sistem yang akan dibangun dirancang. Tahap ini melibatkan pembuatan spesifikasi desain terperinci yang mencakup database, arsitektur perangkat lunak, antarmuka pengguna, dan rencana integrasi sistem. Kemudian, Implementasi dilakukan di mana sistem yang dirancang dikembangkan.

Ini termasuk pengkodean, pengujian modul, dan integrasi sistem. Tahap implementasi ini sering kali membutuhkan iterasi dan perbaikan untuk menyesuaikan dengan kebutuhan sistem yang berkembang (Pressman & Maxim, 2014). Testing merupakan tahap kritis dalam SDLC di mana sistem yang dikembangkan diuji secara menyeluruh untuk memastikan bahwa semua fungsi bekerja seperti yang diharapkan. Pengujian dapat meliputi unit testing, *integration testing*, system testing, dan *acceptance testing*. Fase ini adalah tentang memverifikasi dan memvalidasi sistem yang dikembangkan sebelum diterapkan secara luas. Terakhir, *Deployment and Maintenance* melibatkan penerapan sistem dalam lingkungan produksi dan melakukan pemeliharaan berkelanjutan. Pemeliharaan melibatkan pembaruan sistem, pemecahan masalah, dan peningkatan fungsi sesuai kebutuhan pengguna (Sommerville, 2016).



Gambar 3. 2 Alur Kerja *System Development Life Cycle* (SDLC)

3.3.1 *Requirement Analysis*

Dalam tahap analisis kebutuhan, peneliti mengambil pendekatan yang didasarkan pada penelitian sebelumnya yang relevan dengan enkripsi *Advanced Encryption Standard* (AES). Studi ini telah memberikan wawasan kritis tentang faktor-faktor yang paling signifikan dalam penerapan AES dalam sistem keamanan data, khususnya mengenai keamanan data, kecepatan

pemrosesan, dan kemudahan penggunaan. Tetapi peneliti akan berfokus pada keamanan data dan kemudahan penggunaan dengan merubah hasil AES tersebut menjadi diagram garis menggunakan teknik steganografi.

3.3.2 System Design

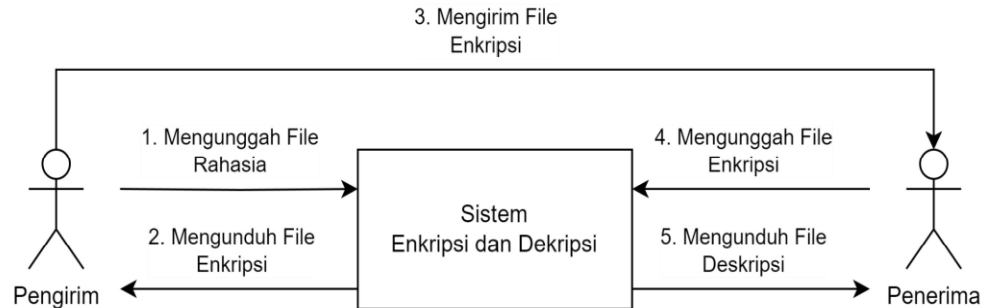
Tahap desain sistem melibatkan pembuatan arsitektur sistem, termasuk teknologi yang digunakan, desain tampilan pengguna, alur aktivitas pengguna dengan sistem, dan diagram alir sistem yang dirancang. Dalam tahap ini, desain sistem dikembangkan untuk mendukung steganografi dan enkripsi AES dengan cara yang efisien dan aman. Peneliti juga menyajikan hasil rancangan tampilan pengguna, alur aktivitas pengguna dengan sistem, dan diagram alir rancangan sistem.

Tabel 3. 1 Tabel Perencanaan Teknologi yang Digunakan

Nama	Fungsi
<i>Text Editor</i> atau <i>Visual Studio Code</i>	Sebagai Teks editor dari bahasa PHP untuk <i>mendevolop</i> aplikasi dalam sistem enkripsi AES dengan steganografinya. dan telah didukung oleh <i>library</i> seperti Enkripsi AES, <i>Smalot</i> , <i>PHPOffice</i> , dan <i>Tcpdf</i> .
PHP	Sebagai bahasa yang digunakan dalam proses pembuatan web enkripsi
XAMPP	Sebagai teknologi server untuk mengoperasikan web dalam sistem enkripsi AES dengan steganografi
Diagram Net	Sebagai software yang membantu untuk mendesain diagram pada pengembangan sistem ini
GitHub	Sebagai tempat atau wadah atau repositori kode untuk memudahkan dalam pengembangan sistem ini

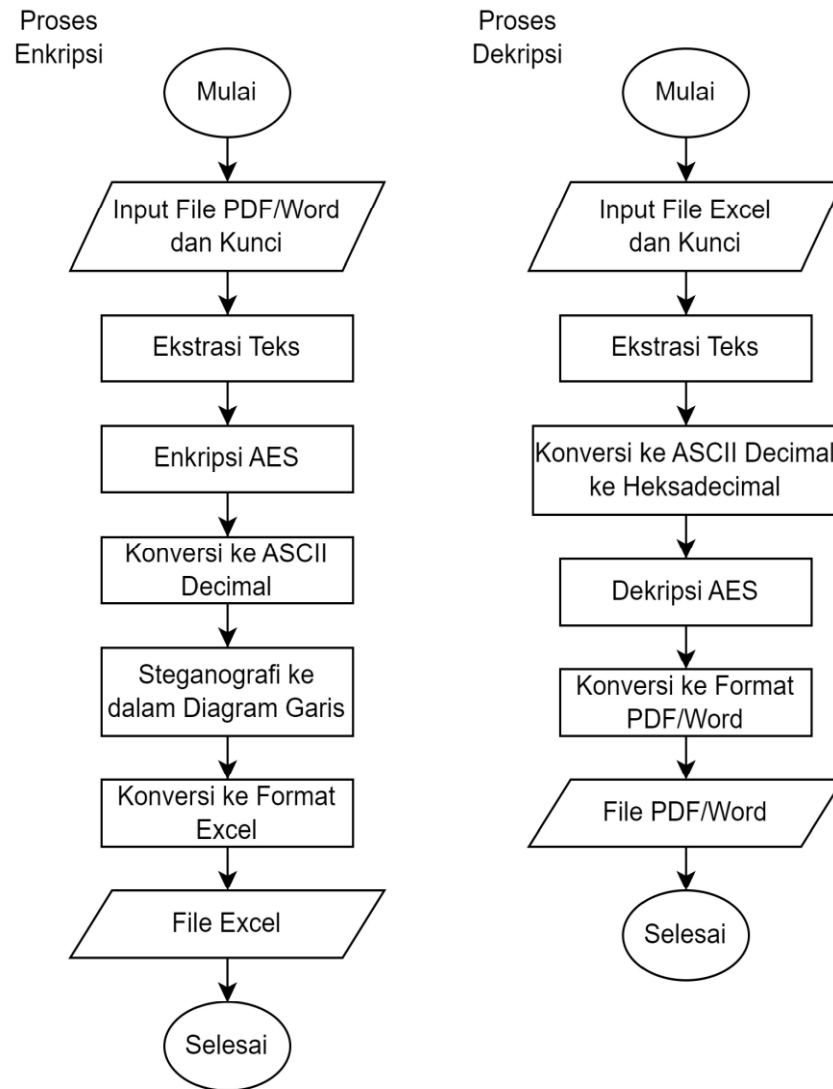
Tabel 3.1 menguraikan teknologi yang digunakan dalam pengembangan sistem enkripsi AES dengan steganografi. Visual Studio Code, yang digunakan sebagai editor teks, mendukung pengeditan kode PHP dan integrasi dengan *library* enkripsi AES, *Smalot*, *PHPOffice*, dan *Tcpdf*, yang penting untuk pengelolaan dokumen dan enkripsi data. PHP, sebagai bahasa pemrograman utama, memfasilitasi pengembangan komponen web dan operasi server. XAMPP, sebagai lingkungan server lokal, memungkinkan pengujian dan

deployment aplikasi PHP dalam lingkungan yang terkontrol. Diagram Net digunakan untuk merancang dan mendokumentasikan arsitektur sistem dan alur data secara visual, yang meningkatkan kejelasan desain selama pengembangan. Akhirnya, GitHub berfungsi sebagai repositori untuk manajemen kode sumber, memudahkan kolaborasi dan *version control* yang mendukung integritas dan efisiensi pengembangan sistem.



Gambar 3. 3 Alur Aktivitas Pengguna dengan Sistem

Pada Gambar 3.3, terlihat bahwa terdapat dua pengguna: pengirim (*User 1*) dan penerima (*User 2*), yang telah sepakat menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. *User 1*, sebagai pengirim, memiliki pesan rahasia yang ingin disampaikan hanya kepada *User 2*. *User 1* kemudian membuat *file* enkripsi dari dokumen berformat PDF yang berisi pesan rahasia melalui sistem yang dikembangkan oleh peneliti. Sistem tersebut mengenkripsi dokumen dan menghasilkan *file* Excel yang di dalamnya terdapat diagram garis hasil penerapan teknik steganografi pada *Advanced Encryption Standard* (AES). Setelah mendapatkan *file* Excel, *User 1* mengirimkannya kepada *User 2* sebagai penerima pesan rahasia. Selanjutnya, *User 2* melakukan dekripsi menggunakan sistem yang sama, di mana *User 2* harus mengunggah *file* Excel ke sistem dan sistem akan memproses *file* tersebut. Hasil dari proses ini memberikan *User 2* pilihan untuk mengunduh *file* dalam format Word atau PDF. *User 2* akhirnya menerima pesan tersembunyi yang dikirim oleh *User 1*.



Gambar 3. 4 Flowchart Rancangan Sistem

Dari *Flowchart* yang diilustrasikan pada Gambar 3.4, dapat disimpulkan bahwa sistem yang dirancang akan menghasilkan dua jenis *output*: *output* dari proses enkripsi yang menghasilkan *file* Excel dan *output* dari proses dekripsi yang menghasilkan *file* PDF atau Word. Alur kerja sistem dijelaskan sebagai berikut: Pertama, *file* PDF diunggah bersama kunci sebagai *input* untuk proses enkripsi. Proses ini meliputi ekstraksi pesan dari *file* yang diunggah menjadi teks biasa yang kemudian diproses oleh algoritma AES. Hasil enkripsi AES, yang berbentuk heksadesimal, diubah menjadi model ASCII Desimal. Selanjutnya, teks enkripsi yang telah diubah menjadi ASCII Desimal ditransformasi menjadi diagram garis menggunakan teknik steganografi, lalu

dikonversi menjadi format yang didukung oleh Excel. Setelah itu, *file* Excel dapat diunggah oleh pengguna untuk dikirimkan kepada pengguna berikutnya. Pengguna kedua kemudian mengunggah *file* yang diterima ke sistem dengan kunci sebagai *input* untuk proses dekripsi. Proses ini meliputi ekstraksi poin-poin penting dari diagram garis dalam *file* Excel yang diunggah, yang menghasilkan teks ASCII Desimal. Teks ini diubah kembali menjadi format heksadesimal, lalu didekripsi menggunakan AES bersama kunci yang diberikan, menghasilkan pesan rahasia yang dikirim oleh pengguna pertama. Setelah didekripsi, teks tersebut diubah sesuai dengan pilihan pengguna antara format PDF atau Word. Pesan akhirnya dapat diunduh sesuai dengan format yang dipilih oleh pengguna.

3.3.3 Implementation

Pada implementasi, kode sumber sistem dikembangkan berdasarkan spesifikasi desain. Pengembangan menggunakan bahasa pemrograman PHP, dengan mengintegrasikan modul-modul seperti modul enkripsi dan modul steganografi. Detail lebih lanjut dari implementasi sistem dapat dilihat pada Bab IV pada Hasil Pengembangan Sistem.

3.3.4 Testing

Pengujian sistem meliputi serangkaian pengujian yang dirancang untuk memverifikasi bahwa semua bagian sistem berfungsi seperti yang diharapkan dan memenuhi persyaratan yang telah dirancang. Pada pengujian peneliti menggunakan *Black-Box* Testing. Detail lebih lanjut dari pengujian ini dapat dilihat pada bagian 3.4 *Black-Box* Testing.

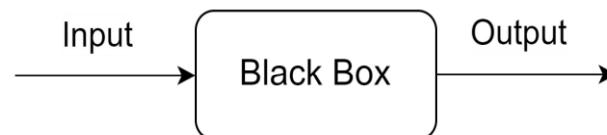
3.3.5 Maintenance

Setelah prototipe sistem dikembangkan dan melewati fase pengujian awal untuk memverifikasi fungsionalitasnya, langkah selanjutnya adalah penerapan dan evaluasi prototipe pada lingkungan pengujian lokal. Tahap ini tidak melibatkan *deployment* ke lingkungan produksi atau internet, melainkan fokus pada pengujian internal untuk menilai efektivitas sistem.

3.4 Black-Box Testing

Metode pengujian sistem yang digunakan oleh Penulis adalah *black-box testing* untuk memeriksa dan memvalidasi fungsionalitas program yang sudah

diimplementasikan. Karena teknik ini, penguji sistem tidak perlu mengetahui kode sistem yang dibuat, hanya perlu menguji fungsionalitasnya dari luar (A. Verma dkk., 2017). Maka penulis bisa konsentrasi pada *input* dan *output* yang diinginkan untuk dievaluasi langsung dari hasil sistem dalam kondisi dan skenario yang telah dibuat. Gambar 3.5 menunjukkan contoh dari proses kerja metode pengujian sistem *black-box* testing.



Gambar 3. 5 Alur Pengujian Metode *Black-box Testing*

Dengan menggunakan metode pengujian *black box* testing, peneliti dapat merancang skenario yang akan diuji dari sistem yang telah dirancang. Pengujian ini mencakup beberapa aspek, seperti Pengujian Fungsionalitas sistem dan Hasil dari Proses Enkripsi. Adapun ruang lingkup dari pengujian tersebut dapat dilihat pada Tabel 3.2 dan Tabel 3.3.

Tabel 3. 2 Tabel Pengujian Fungsionalitas Sistem

Nama Proses	Deskripsi	Keberhasilan
<i>Upload dan Download File</i>	Memastikan <i>file</i> dapat diunggah dan diunduh dengan benar dalam format yang direncanakan (PDF, Excel, Word).	Berhasil / Tidak
Konversi <i>File</i>	Verifikasi bahwa konversi antara format <i>file</i> (PDF/Word ke Excel, dan sebaliknya) berjalan sesuai dengan ekspektasi.	Berhasil / Tidak
Enkripsi dan Dekripsi	Menguji keberhasilan proses enkripsi dan dekripsi menggunakan kunci yang diberikan, memastikan bahwa <i>output</i> adalah teks yang diharapkan.	Berhasil / Tidak

Nama Proses	Deskripsi	Keberhasilan
Pilihan Format <i>File</i> pada <i>Output</i>	Memastikan bahwa sistem menawarkan pilihan format <i>file</i> yang tepat pada saat pengunduhan (PDF atau Word).	Berhasil / Tidak

Tabel 3. 3 Tabel Hasil dari Proses Enkripsi

Nama Proses	Dekripsi	Keberhasilan
Keakuratan Enkripsi	Memeriksa apakah enkripsi menghasilkan <i>output</i> yang sesuai,	Berhasil / Tidak
Keamanan Data	Verifikasi bahwa data yang dienkripsi tidak dapat dibaca tanpa kunci yang sesuai.	Berhasil / Tidak