BAB III
METODE PENELTIAN

3.1 Identifikasi Masalah

Saat mengakses suatu *website* tertentu, pasti di antaranya terdapat *website* yang memiliki sistem *login website*. Sistem *login website* adalah sebuah mekanisme yang memungkinkan pengguna untuk mengakses halaman *website* tertentu dengan cara mengautentikasi diri menggunakan identitas yang valid, seperti *username* dan *password*, atau metode autentikasi lainnya seperti autentikasi dua faktor. Setelah pengguna berhasil melewati proses autentikasi, sistem akan memberikan akses ke halaman *website* yang diizinkan sesuai dengan hak akses yang telah ditentukan sebelumnya. Sistem *login website* biasanya melibatkan penyimpanan dan verifikasi informasi autentikasi pengguna, serta pengelolaan sesi untuk menjaga keamanan dan privasi pengguna selama interaksi dengan *website*. Dengan demikan, agar informasi pengguna tetap aman dan tidak dapat terbaca oleh pencipta *website* itu sendiri, salah satunya perlu diterapkannya algoritma kriptografi. Algoritma tersebut di antaranya adalah menggunakan penggabungan algoritma SHA-256 dengan AES-CBC-256. Salah satu contohnya, misalnya adalah pada *website e-commerce*.

Jika pada *e-commerce* tidak memiliki sistem *login* memungkinkan pengguna yang tidak bertanggung jawab akan membeli barang-barang mahal atas nama korban dan mengirimkannya ke lokasi pengguna yang tidak bertanggung jawab tersebut, misalnya dengan menggunakan fitur *pay later* (fitur yang memungkinkan pengguna bertransaksi terlebih dahulu dan melakukan pembayaran di kemudian hari). Oleh karena itu, sistem login diperlukan untuk memverifikasi identitas pengguna, melindungi data pribadi, mencegah penyalahgunaan fitur finansial seperti pay later, serta memudahkan pelacakan riwayat transaksi guna meningkatkan keamanan dan kenyamanan dalam berbelanja. Adapun e-commerce yang tidak memerlukan login biasanya menggunakan sistem atau toko online mandiri *guest checkout*, seperti aplikasi

Sidqi Amanullah, 2025
IMPLEMENTASI KEAMANAN DATA BERBASIS WEB MENGGUNAKAN KOMBINASI ALGORITMA SECURE HASH ALGORITHM 256 DAN ADVANCED ENCRYPTION STANDARD CIPHER BLOCK CHAINING 256 DENGAN MANAJEMEN DATABASE FIRESTORE PADA AKUN WEBSITE
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

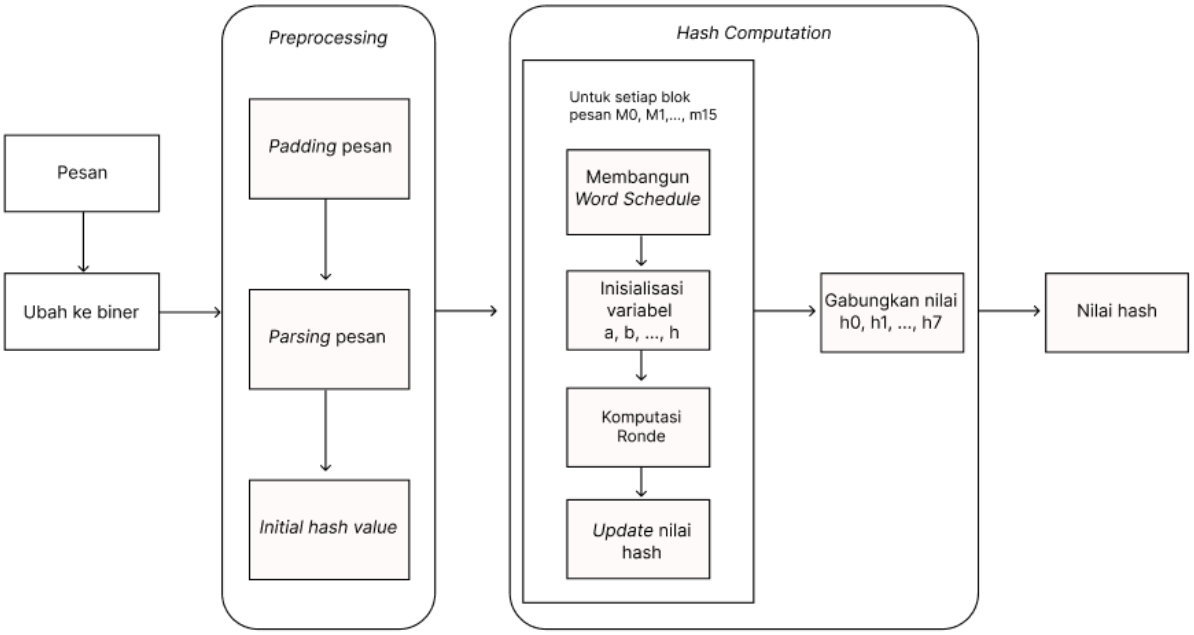
eBay. Akan tetapi, meskipun tidak mewajibkan login, tetap perlu diterapkan syarat-syarat tertentu, misalnya verifikasi alamat, email, atau nomor telepon untuk menjaga keamanan transaksi.

Pada penelitian ini, *plaintext* yang di-*hash*-ing oleh SHA-256 akan menghasilkan *message digest*. *Message digest* inilah yang akan digunakan sebagai kunci AES-256-CBC pada proses memperoleh kode rahasia. *Initialization Vector* (*IV*) pada proses kode rahasia diperoleh dari XOR antara email dan tanggal pembuatan akun. Sementara itu, untuk proses *ciphertext password* yang disimpan di *database, message digest* digunakan sebagai kunci sekaligus *plaintext* untuk proses AES-CBC-256 sedangkan *Initialization Vector* (*IV*) pada AES-CBC-256 diperoleh dari kombinasi email dan password. Hasil *ciphertext* dari AES-CBC-256 akan dimasukan kedalam *firestore database* yang mana *ciphertext* tersebut akan sulit dimengerti oleh admin atau pengelola *database*-nya sendiri. Untuk memvalidasi pengguna dilakukan dalam halaman *login* yang mana nantinya sistem akan membandingkan hasil *ciphertext* pada yang di-*input*-kan pengguna dengan hasil *ciphertext* yang ada di dalam *firestore database*. Password yang diregistrasikan harus memiliki panjang minimal 8 karakter, dengan panjang maksimum yang tidak dibatasi, dan dapat terdiri dari karakter apa pun yang ada pada keyboard (huruf besar, huruf kecil, angka, maupun simbol).

3.2 Model Dasar

3.2.1 Skema SHA-256

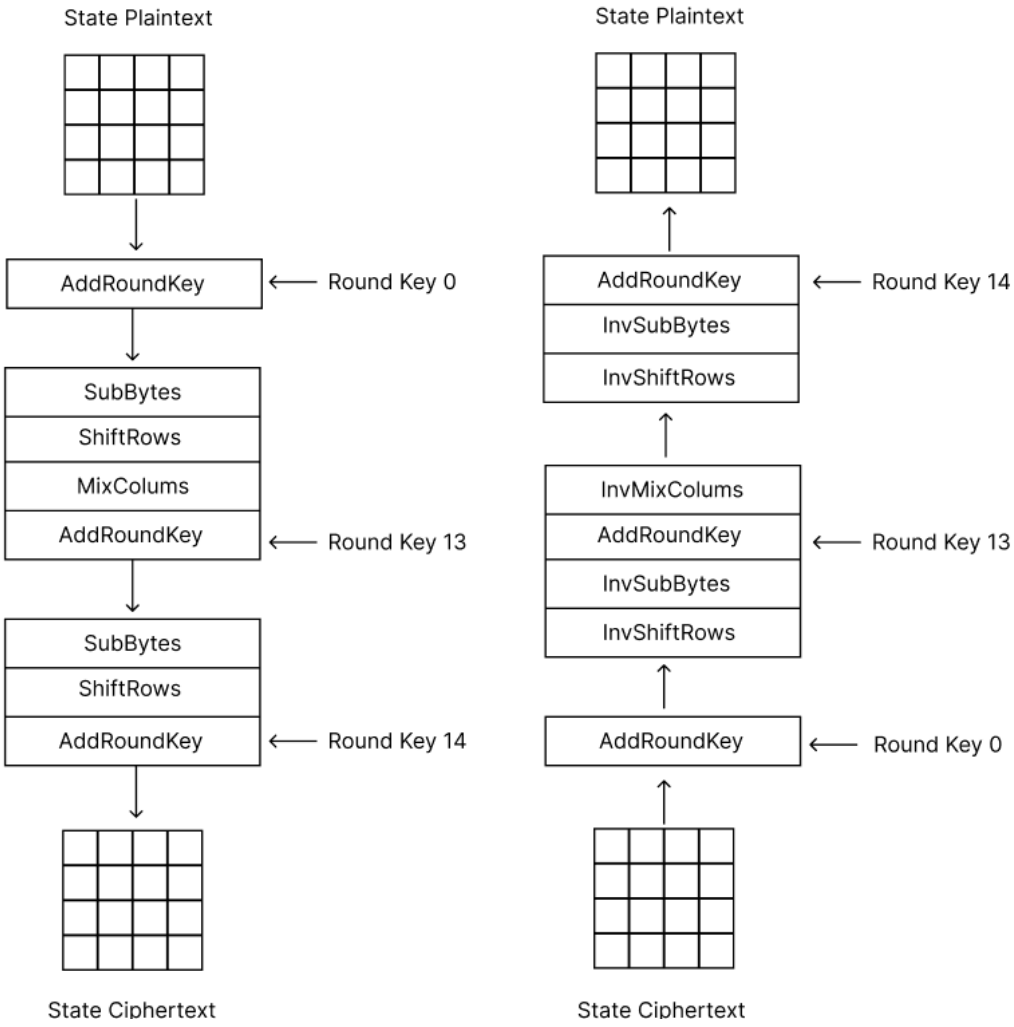
Hasil dari algoritma SHA-256 berupa nilai *hash* sepanjang 256 bit (64 karakter heksadesimal) digunakan sebagai kunci dan *plaintext* dalam enkripsi AES-256-CBC. SHA-256 menghasilkan output dengan panjang tetap, sehingga meskipun input awal pendek, tetap dapat memenuhi kebutuhan panjang yang disyaratkan oleh AES-256-CBC.



Gambar 3.1 Skema Proses SHA-256

3.2.2 Skema AES-256

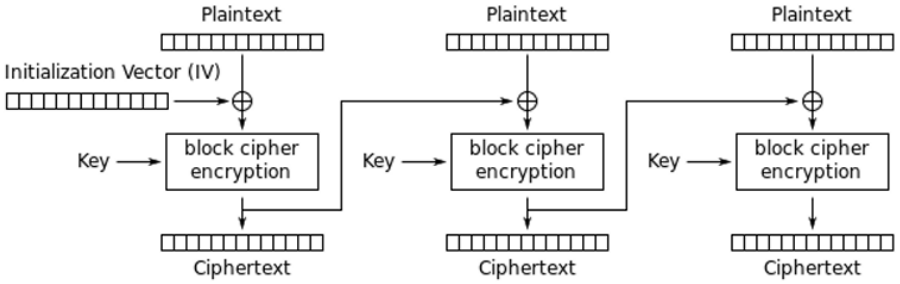
Enkripsi dan dekripsi dari AES-256 menghasilkan *plaintext* dan *ciphertext* yang akan digunakan nantinya dalam metode CBC. Proses enkripsi dan dekripsi menggunakan algoritma AES-256 dijelaskan pada skema berikut:



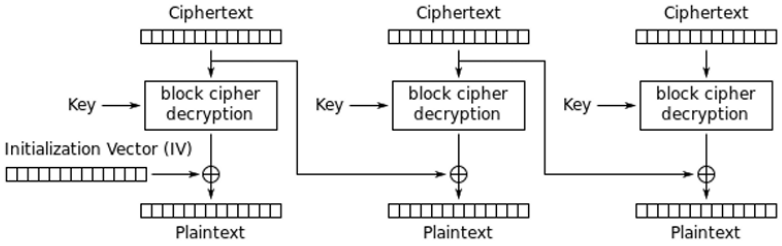
Gambar 3.2 Skema Enrkipsi dan Dekripsi AES-256

3.2.3 Skema AES-CBC-256

Pada mode CBC (*Cipher Block Chaining*), setiap blok *plaintext* di-XOR dengan blok cipherteks sebelumnya sebelum dienkripsi. Dengan demikian, setiap blok *ciphertext* sangat bergantung pada blok *plaintext* yang telah diproses. Untuk membuat setiap pesan unik, sebuah *initialization vector* harus digunakan pada blok pertama. Untuk lebih jelasnya, lihat ilustrasi pada gambar di bawah ini.



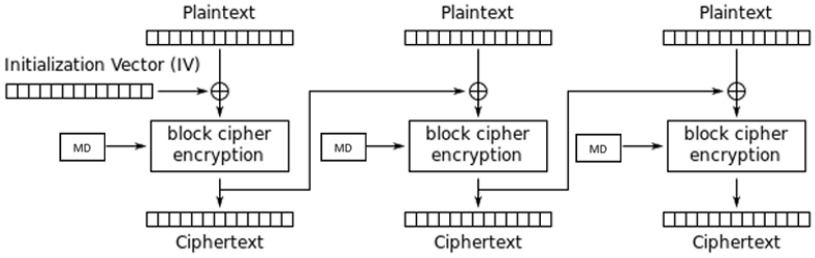
Gambar 3.3 Skema Proses Enkripsi AES-CBC-256 (Warkim dan Lewelusa, 2015)



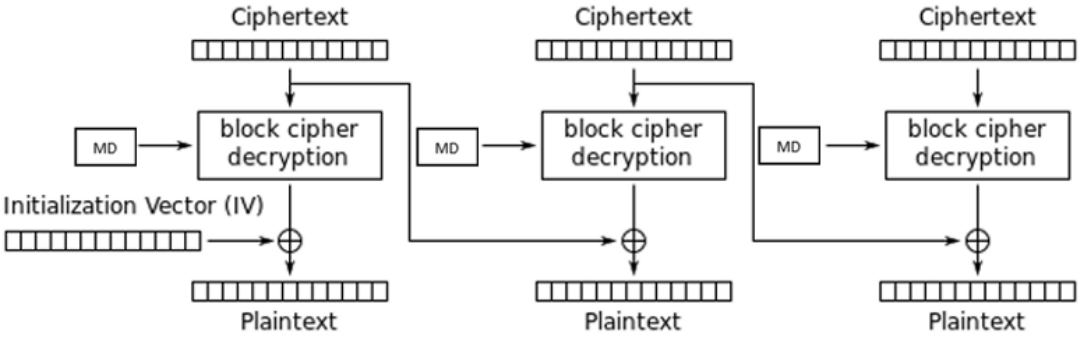
Gambar 3.4 Skema Proses Dekripsi AES-CBC-256 (Warkim dan Lewelusa, 2015)

3.3 Pengembangan Model

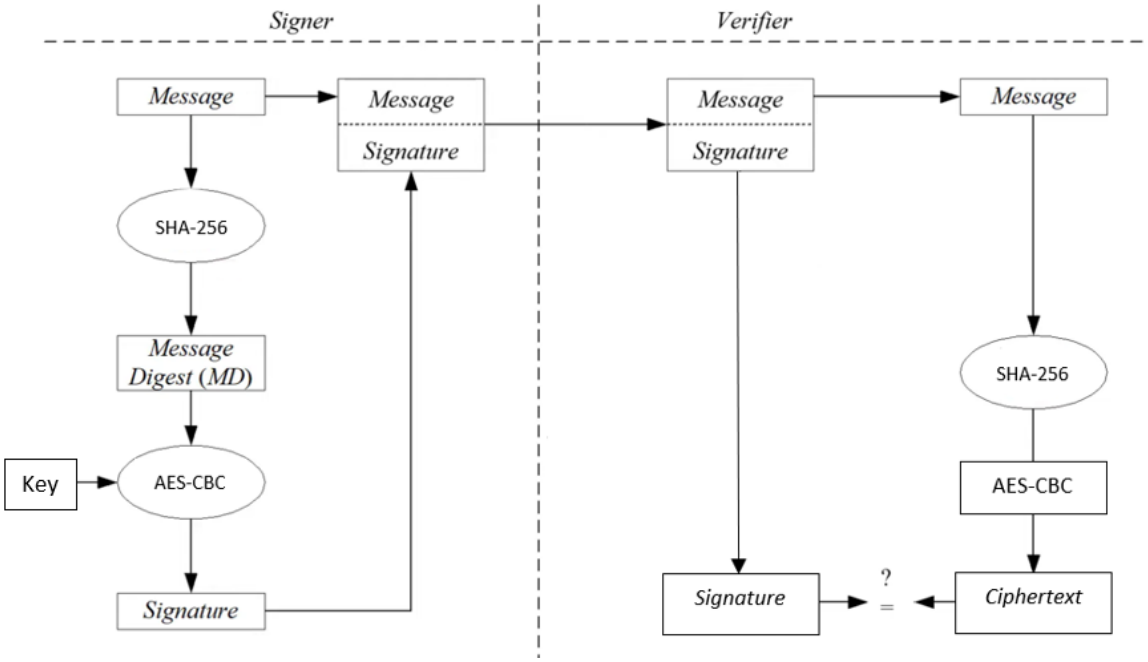
Plaintext yang di-input-kan pengguna pertama-tama akan diproses menggunakan algoritma SHA-256 menghasilkan *message digest*. *Message digest* ini yang akan menjadi *plaintext* dalam algoritma AES-CBC-256 sekaligus kunci pada enkripsi algoritma AES-256. Untuk lebih jelasnya, perhatikan ilustrasi pada gambar di bawah ini.



Gambar 3.5 Skema Proses Enkripsi Kombinasi SHA-256 dan AES-CBC-256 Untuk Fitur Lupa *Password*



Gambar 3.6 Skema Proses Dekripsi Kombinasi SHA-256 dan AES-CBC-256 Untuk Fitur Lupa *Password*



Gambar 3.7 Skema kombinasi algoritma SHA-256 dan AES-CBC-256 untuk Disimpan Dalam Database

Pada algoritma tersebut digunakan dalam enkripsi *password* pengguna yang akan disimpan dalam *database firestore*.

3.4 Konstruksi Program

Program komputer pada penelitian ini menggunakan bahasa pemrograman Java Script dengan rincian konstruksi sebagai berikut.

3.4.1 *Input* dan *Output* Aplikasi *Website*

Aplikasi *website* ini terdiri dari tiga halaman: registrasi, *login*, dan beranda. Halaman registrasi memungkinkan pengguna untuk mendaftar dengan memasukkan email dan kata sandi, serta menampilkan hasil proses pendaftaran. Halaman *login* digunakan oleh pengguna untuk masuk menggunakan email dan *password* agar dapat mengakses halaman beranda. Jika pengguna salah memasukkan data pada halaman *login*, maka proses *login* tidak akan berhasil. Di halaman beranda, pengguna yang telah berhasil *login* akan disambut dengan pesan "Halo pengguna". Jika ada pengguna yang mencoba mengakses halaman beranda tanpa *login* terlebih dahulu, mereka akan diarahkan kembali ke halaman *login*. Pada halaman beranda, pengguna dapat mengganti *password*.

3.4.2 Algoritma Deskriptif

Algoritma untuk melakukan proses enkripsi dan dekripsi pada *password* diuraikan sebagai berikut:

A. *Register* akun pengguna (Enkripsi)

1. *Input Username*, email dan *Password*.
2. Tekan Tombol register
3. User memperoleh kode rahasia

B. *Login* akun pengguna

1. Input email dan *password*.
2. Tekan tombol *login*.
3. Tunggu Verifikasi

C. Lupa *password* (Dekripsi)

1. Tekan tombol lupa *password*
2. Input email, kode rahasia dan tanggal pembuatan akun

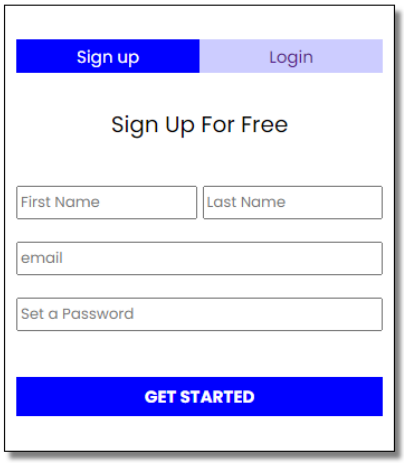
3. Tekan tombol prosess
4. Tuggu Prosess dekripsi
5. Jika valid, maka mendapatkan *password* pengguna

D. Server

1. Memverifikasi akun pengguna dari email dan hasil enkripsi *password* menggunakan AES-CBC-256 dan SHA-256.

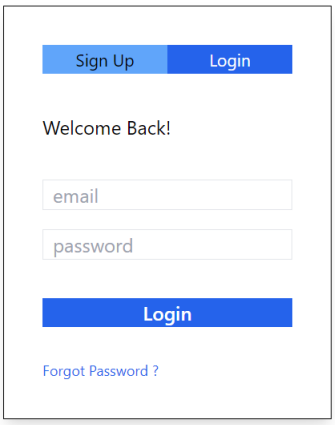
3.4.3 Rancangan Tampilan

1) Rancangan tampilan *Registrasi*



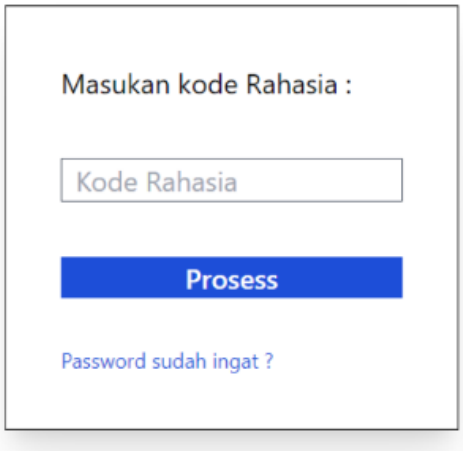
Gambar 3.8 Halaman *Registrasi*

2) Rancangan tampilan *login*



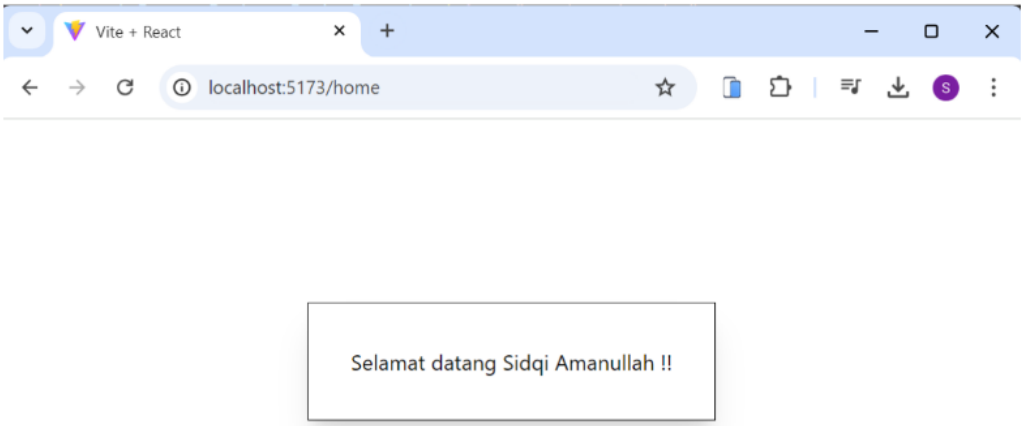
Gambar 3.9 Halaman *Login*

3) Rancangan Tampilan Lupa *password*



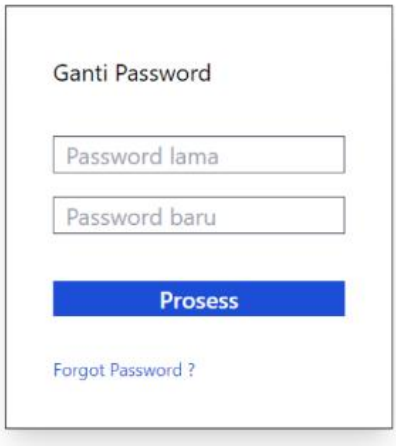
Gambar 3.10 Halaman Lupa *password*

4) Rancangan tampilan *Home*



Gambar 3.11 Halaman *Home*

5) Rancangan Tampilan Ganti *Password* :



Gambar 3.12 Halaman Ganti *Password*

6) Rancangan Tampilan Pada *Database* :

Gambar 3.10 Halaman *Database firestore*

3.5 *Library* Program

Library dalam bahasa pemrograman adalah kumpulan fungsi, kelas, atau modul yang sudah disiapkan untuk digunakan dalam pengembangan aplikasi. Library berisi kode yang telah dikemas untuk menyelesaikan tugas tertentu, sehingga pengembang tidak perlu menulis semuanya dari awal. Dengan menggunakan library, pengembangan aplikasi menjadi lebih cepat, efisien, dan lebih terorganisir.

3.5.1 *Library* Java Script

Pada pembuatan program akan didukung dengan menggunakan *library* java script, yaitu React js. React.js adalah pustaka (*library*) JavaScript yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna (UI) yang interaktif dan dinamis. React.js memungkinkan pengembang untuk membuat komponen UI yang dapat digunakan kembali, sehingga memudahkan dalam pengelolaan dan pengembangan aplikasi *website* yang kompleks. React menggunakan JSX, yaitu ekstensi sintaks JavaScript yang memungkinkan penulisan elemen UI menggunakan kode yang mirip dengan HTML. JSX membuat penulisan elemen React lebih intuitif dan mudah dibaca.

3.5.2 Framework React JS

Framework adalah kerangka kerja atau infrastruktur yang lebih besar daripada *library*. Framework mengatur cara pengembangan aplikasi dari awal hingga akhir, memberikan rangka kerja lengkap untuk memudahkan pembuatan aplikasi. Biasanya, framework menentukan alur kerja yang harus diikuti. Pada pembuatan projek ini akan menggunakan framework React JS, yaitu next JS. Next js mirip seperti react JS akan tetapi memiliki fitur-fitur tambahan dengan rendering cepat dan interaksi halaman yang lebih lancer.

3.5.3 *Firestore Database*

Dalam penyimpanan data pengguna setelah proses registrasi, data tersebut akan disimpan ke dalam *database Firestore*. *Firestore*, atau lebih dikenal sebagai *Cloud Firestore*, adalah layanan *database* NoSQL berbasis cloud yang disediakan oleh Google sebagai bagian dari *Firebase*, platform pengembangan aplikasi untuk *website* dan perangkat mobile. *Firestore* dirancang untuk menyimpan, mengelola, dan menyinkronkan data secara real-time antara klien (seperti aplikasi *website* atau mobile) dan server.

Firestore menawarkan skalabilitas tinggi dan fitur-fitur canggih, seperti penyimpanan data offline dan integrasi mudah dengan layanan lain dalam ekosistem

Firebase. Ini memudahkan pengembang untuk membangun aplikasi yang responsif dengan data yang diperbarui secara instan diberbagai perangkat pengguna.

3.5.4 Coding *Website*

Terdapat beberapa Bahasa pemrograman dan Bahasa markup yang digunakan pada proses pemrograman, yaitu:

1. HTML

   HTML, atau *HyperText Markup Languange*, adalah bahasa standar yang digunakan untuk membuat dan merancang halaman web. HTML adalah fondasi dari semua halaman web dan berfungsi sebagai kerangka kerja untuk menyusun konten dan elemen di dalam halaman tersebut

2. CSS

   CSS (*Cascading Style Sheets*) adalah bahasa yang digunakan untuk mendesain dan mengatur tampilan halaman web yang dibuat menggunakan HTML. CSS memungkinkan pengembang web untuk memisahkan konten (struktur HTML) dari tampilan (gaya) sehingga mempermudah pemeliharaan dan pengelolaan web.

3. Java Script

   JavaScript adalah bahasa pemrograman tingkat tinggi, dinamis, dan interpretatif yang digunakan terutama untuk pengembangan web. Awalnya dikembangkan oleh Netscape sebagai cara untuk menambahkan logika pemrograman ke dalam halaman web, JavaScript kini didukung oleh semua browser modern.

3.6 Proses Validasi

Pada tahap ini, program akan diuji validitasnya dengan membandingkan informasi pengguna yang telah diregistrasikan dengan data yang diinput oleh pengguna pada halaman login. Program dianggap valid apabila data dalam database sesuai dengan data yang dimasukkan oleh pengguna. Jika pengguna lupa kata sandi, tersedia fitur 'lupa password' yang memungkinkan pengguna melihat kembali kata sandi awal.

Dengan demikian, program dianggap berhasil ketika dapat memulihkan kata sandi dan memungkinkan pengguna untuk melakukan validasi ulang agar dapat mengakses halaman *website*. Selain itu, kombinasi algoritma kriptografi juga akan diuji menggunakan *website* eksternal untuk membandingkan apakah hasil ciphertext yang dihasilkan program ini sama dengan hasil yang diperoleh dari situs tersebut.

3.7 Pengambilan Kesimpulan

Tahap terakhir yang akan dilakukan adalah pengambilan kesimpulan berdasarkan hasil yang diperoleh selama penelitian, serta rekomendasi untuk penelitian selanjutnya agar memperoleh hasil yang lebih maksimal.