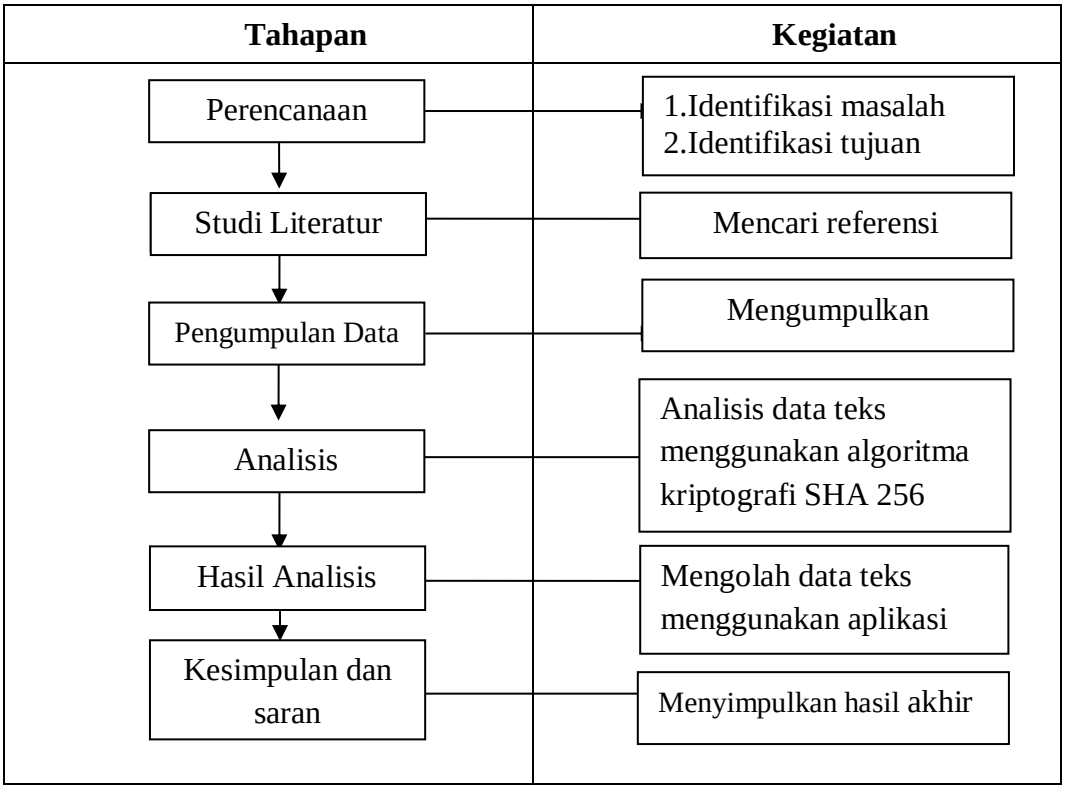
BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Ada beberapa proses yang dilakukan dalam merancang aplikasi sebelum dibuat. Mulai dari tahap desain, berbagai topik akan dibahas, termasuk data yang dikumpulkan dari lokasi penelitian dan desain aplikasi.



Gambar 3. 1 Desain Penelitian
Sumber : (Data Penelitian, 2023)

Berdasarkan tabel langkah-langkah penelitian di atas, maka penjelasannya sebagai berikut:

1. Perencanaan

   Langkah pertama dalam mencapai suatu tujuan atau pencapaian adalah perencanaan, yang pada tahap ini berupaya melakukan persiapan terhadap

tujuan dan bentuk permasalahan. Tahap ini akan melibatkan beberapa langkah, yang pertama adalah mengidentifikasi masalah yang ingin ditangani, tujuan, dan sejauh mana diskusi.

a. Menentukan masalah

   Pada Dengan menggunakan algoritma kriptografi *Secure Hash Algorithm 256*, peneliti akan mengidentifikasi kelemahan pada desain keamanan data teks pada saat ini. Peneliti akan menggunakan masalah ini sebagai topik penelitian.

b. Menentukan tujuan

   Untuk meningkatkan keamanan data teks kedepannya, peneliti akan mengidentifikasi tujuan atau solusi dari permasalahan yang ada saat ini dalam perancangan keamanan data teks dengan menggunakan algoritma kriptografi *Secure Hash Algorithm 256*.

b. Menentukan ruang lingkupPenetapan ruang lingkup akan membantu memastikan bahwa penelitian terfokus dan tidak menyimpang dari pokok bahasan. Dimana penelitian ini hanya membahas perancangan keamanan data teks dengan memanfaatkan algoritma kriptografi *Secure Hash Algorithm 256.*

2. Studi literatur

Langkah selanjutnya setelah perencanaan adalah studi literatur, dimana peneliti mencari referensi untuk menemukan pengetahuan atau penelitian sebelumnya yang diperlukan.

3. Pengumpulan data

Tahap selanjutnya disebut pengumpulan data, dan tujuannya adalah mengumpulkan informasi atau data apa pun yang dibutuhkan peneliti.

4. Analisis

Kemampuan untuk membedah dan menjelaskan sesuatu ke dalam bagian-bagian yang dapat dikelola untuk dipahami dikenal sebagai analisis. Analisis dilakukan dengan menggunakan langkah-langkah berikut:

a. Analisis Data

Analisis data dilakukan untuk megubah hasil penelitian menjadi informasi baru yang berguna untuk mengambil keputusan.

c. Analisis Permasalahan

Analisis masalah dilakukan untuk mengidentifikasi permasalahan yang ada dan menentukan solusi dari permasalahan tersebut.

5. Hasil Analisis

Tahapan ini merupakan proses pengolahan data yang di peroleh dari data perancangan keamanan data teks menggunakan algoritma *kriptografi Secure Hash Algorithm 256*.

6. Kesimpulan dan Saran

Poin-poin yang telah diselesaikan dari tahap sebelumnya membentuk tahap ini. Rekomendasi ini merupakan perbaikan yang dimaksudkan untuk mengatasi permasalahan yang penting dan bermanfaat bagi penelitian ini.

3.2 Metode Pengumpulan Data

Penumpulan data yang digunakan dalam penelitian ini:

a. Data Sekunder

Peneliti dapat memperoleh data sekunder dari sumber yang sudah ada sebelumnya, seperti buku, jurnal, makalah, dan lain-lain.

b. Data primer

Pencarian data diperlukan bagi peneliti untuk mengumpulkan informasi yang keberadaannya sesuai dengan fakta dan dapat dipertanggungjawabkan dalam suatu laporan.

Dalam penelitian ini, teknik pengumpulan data berikut yang digunakan:

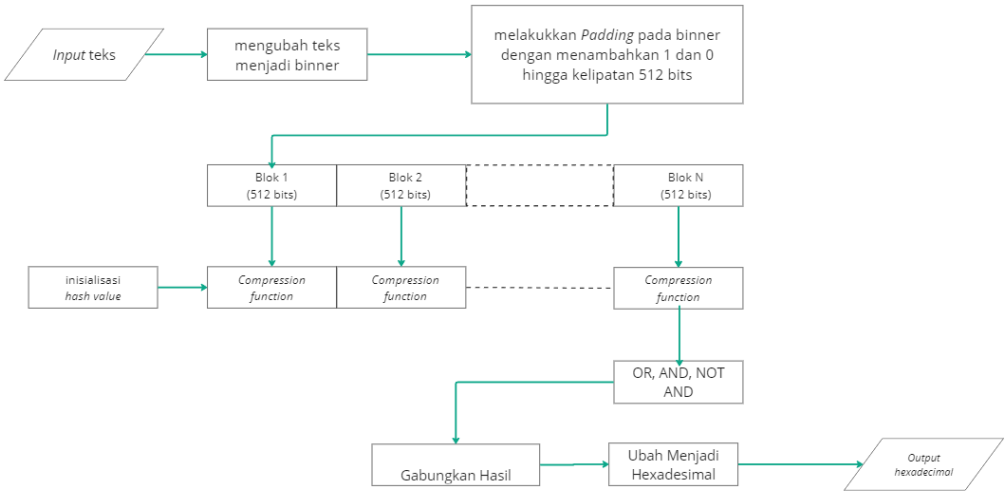
1. Observasi

Peneliti menggunakan observasi sebagai metode untuk mengumpulkan data dengan melihat subjek penelitiannya dari dekat. Dengan menggunakan algoritma kriptografi *Secure Hash Algorithm 256*, peneliti mengamati langsung proses perancangan keamanan data teks sepanjang tahap observasi ini.

2. Studi Literature

Metode pengumpulan data untuk suatu penelitian adalah studi literatur, yang melibatkan membaca dan menganalisis buku, jurnal, artikel, dan karya lain mengenai subjek terkait.

3.2.1 Algoritma SHA 256



Gambar 3. 2 Algoritma SHA

Hash 256, sebagai algoritma pengamanan yang kritis, melibatkan serangkaian proses sistematis yang kompleks. Ini mencakup tahap-tahap esensial yang digunakan oleh sistem untuk mengonversi teks menjadi hash dengan keefisienan dan keamanan yang tinggi. Berikut adalah Algoritma dari *SHA 256*:

1. Memasukkan teks yang akan dilakukan *hash*.
2. Mengubah teks menjadi binner teks yang dimasukkan diubah menjadi biner

berdasarkan nilai desimal yang terkandung dalam kode *ASCII*.

3. Melakukkan pemadatan biner (*Padding*). Setelah itu, dilakukan pemadatan teks (*padding*) dengan langkah-langkah berikut:

- Penambahan angka 1 di akhir biner yang dihasilkan
- Penambahan angka 0 untuk mengisi ruang *padding* yang diperlukan untuk memastikan setiap blok biner memiliki total 512 bit.
- Penambahan angka terakhir dalam bentuk biner, yang merepresentasikan jumlah bit dalam teks.
- Sebagai contoh, kita ingin melakukan enkripsi pada teks *"PASSWORD"* yang direpresentasikan dalam bentuk biner sebagai berikut di table 3.2.

Untuk mempermudah dalam menenyukan bilangan binert dari sebuah huruf penulis membuatkan tabel ASC II yang berisikan huruf abjad lengkap dengan bilangan biner dari huruf tersebut, Supaya memudahkan dalam melakukan enskripsi bilangan dalam bentuk biner. Seperti yang tertera di bawah ini tabel 3.1 ASC II,

Tabel 3. 1 ASC II

Char	Binary	Char	Binary
A	01000001	a	01100001
B	01000010	b	01100010
C	01000011	c	01100011
D	01000100	d	01100100
E	01000101	e	01100101
F	01000110	f	01100110
G	01000111	g	01100111
H	01001000	h	01101000
I	01001001	i	01101001
J	01001010	j	01101010
K	01001011	k	01101011
L	01001100	l	01101100
M	01001101	m	01101101
N	01001110	n	01101110
O	01001111	o	01101111
P	01010000	p	01110000
Q	01010001	q	01110001
R	01010010	r	01110010
S	01010011	s	01110011
T	01010100	t	01110100
U	01010101	u	01110101
V	01010110	v	01110110
W	01010111	w	01110111
X	01011000	x	01111000
Y	01011001	y	01111001
Z	01011010	z	01111010

Tabel 3.1 di atas merupakan tabel ASC II yang berisi huruf A – Z besar dan huruf a – z kecil yang berfungsi untuk memudahkan kita mencari biner yang akan kita gunakan dalam proses pemadatan.

Tabel 3. 2 Pemadatan Biner

Karakter	*ASCII desimal*	Biner
P	80	01010000
A	65	01000001
S	83	01010011
S	83	01010011
W	87	01010111
O	79	01001111
R	82	01010010
D	68	01000100

- Jumlah bit total dari teks *"password"* adalah 64 (3 karakter * 8 bit). Untuk memastikan setiap blok biner memiliki total 512 bit, dilakukan proses padding sebagai berikut:

Tabel 3. 3 Proses Padding

Chiperteks	*Padding 512*
01010000 01000001 01010011 01010011 01010111 01001111 01010010 01000100	**01110000 1100001 1110011 1110011 01110111 1101111 1110010 01100100 10000000 00000000 0000000 ... 01000000**

Proses ini melibatkan penambahan angka 1 di akhir blok, diikuti oleh penambahan angka 0 untuk mengisi ruang padding yang diperlukan hingga total bit mencapai 512. Akhirnya, angka biner 01000000 ditambahkan untuk merepresentasikan jumlah bit asli dari teks "*PASSWORD*" yang telah diproses, sehingga memastikan konsistensi dan integritas dalam proses hashing.

4. Membagi hasil pemadatan binner per 512 bit

Hasil dari pemadatan tersebut dibagi per 512 bit, misalnya jika terdapat lebih dari 1 blok. Namun, pada contoh "*PASSWORD*", hanya terbentuk 1 blok setelah proses pemadatan berapa karakter yang pemadatan

5. Inisialisasi konstanta *SHA-256*

Inisial *hash value* adalah nilai-nilai awal yang digunakan sebagai titik awal dalam proses pembangunan *hash* selama eksekusi algoritma tertentu, seperti algoritma *hash SHA-256*. Nilai-nilai inisial ini sering disebut sebagai "A" hingga "H" dan memberikan dasar awal untuk menghasilkan hash dari blok-blok data. Dalam konteks hash yang disebutkan, nilai inisial adalah sebagai berikut:

A: 6a09e667

B: bb67ae85

C: 3c6ef372

D: a54ff53a

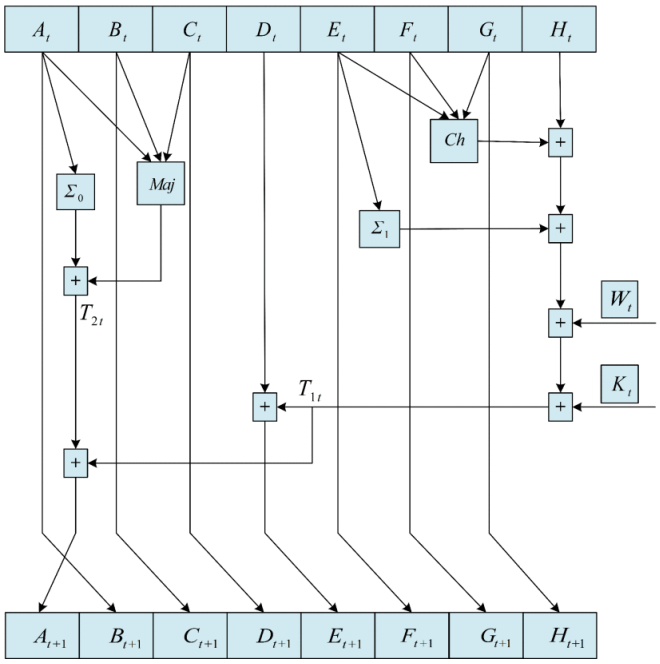
E: 510e527f

F: 9b05688c

G: 1f83d9ab

H: 5be0cd19

6. Pengelolahan hasil binner (*Compression Function*)



Gambar 3. 3 *Compression Function*

SHA-256 memiliki 64 iterasi dalam fungsi kompresi. Setiap iterasi melibatkan serangkaian operasi logika dan aritmatika pada variabel-variabel hash intermediate (A, B, C, D, E, F, G, H). dan Proses ini diulangi 64 kali, dengan nilai-nilai yang terus diperbarui pada setiap iterasi.

Pada diagram diatas, inisial "**W"** merupakan 1 blok kata yang akan diproses, contoh pada penelitian ini menggunakan blok kata "Passwarod". Inisial **K** merupakan serangkaian konstanta yang digunakan dalam setiap iterasi fungsi kompresi. Terdapat 64 konstanta K yang digunakan dan

dirancang untuk menambahkan keragaman dan kompleksitas pada proses penghasilan nilai hash. dan Inisial **"T"** adalah variable sementara (*temporary variable)* yang digunakan untuk menampung hasil dari proses logika. Proses logika yang digunakan pada hash 256 adalah:

- **Ch (*Choose*):** Fungsi ini mengambil dua argumen, F, G, dan H, dan menghasilkan output berdasarkan nilai F jika F adalah True, atau nilai H jika F adalah False.

  Ch(E, F, G) = (E AND F) XOR ((NOT E) AND G)

Tabel 3. 4 XOR

E	F	AND	E	G	Xor
0	**0**	**0**	**1**	**0**	**1**
1	**1**	**1**	**0**	**1**	**1**
0	**0**	**0**	**1**	**0**	**1**
0	**0**	**0**	**1**	**0**	**1**
0	**0**	**0**	**1**	**0**	**1**
1	**1**	**1**	**0**	**1**	**1**
0	**1**	**0**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**	**1**

- **Maj (*Majority*):** Fungsi ini mengambil tiga argumen dan menghasilkan output berdasarkan mayoritas nilai input.

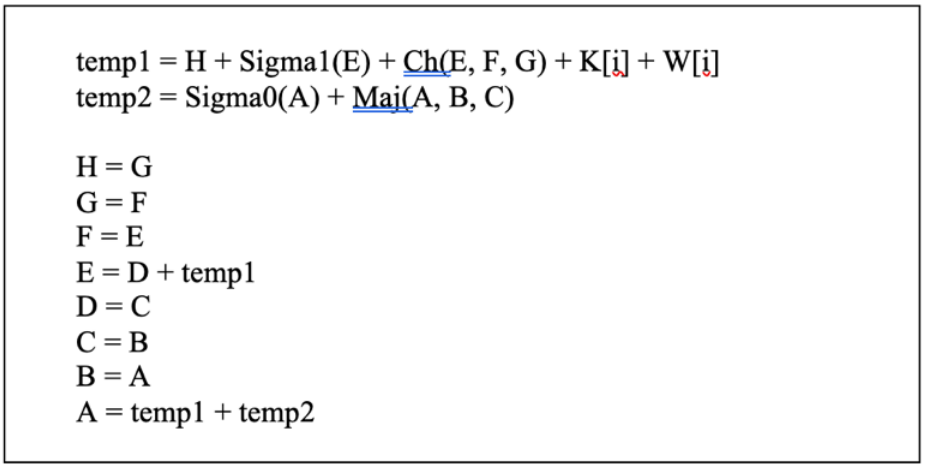
  Maj (A, B, C) = (A AND B) XOR (A AND C) XOR (B AND C)

- **Sigma0 dan Sigma1:** Kedua fungsi ini melakukan operasi pergeseran bit pada input.

  Sigma0(X) = ROTR^2(X) XOR ROTR^13(X) XOR ROTR^22(X)

  Sigma1(X) = ROTR^6(X) XOR ROTR^11(X) XOR ROTR^25(X)

  Secara sederhana proses pada diagram diatas dapat disimpulkan seperti contoh dibawah ini



Gambar 3. 4 Penyederhanaan Diagram

7. Menggabungkan hasil dari setiap blok

   Setelah dilakukan enkripsi, maka akan menggabungkan hasil enkripsi per blok (512 bit) dalam bentuk biner. Untuk contoh penelitian ini menggunakan contoh *"PASSWORD"* yang hanya menghasilkan 1 blok, sehingga hasil akan diteruskan tanpa menggabungkan blok lainnya.

8. Konversi binner menjadi *hexadecimal*.

   Hasil enkripsi dalam bentuk binner akan dikonversi terlebih dahulu ke dalam bentuk hexadecimal. Untuk mengubah bilangan biner menjadi heksadesimal, langkah pertama adalah memisahkan bilangan biner ke dalam grup 4 bit, dimulai dari ujung kanan. Jika panjang bilangan biner tidak habis dibagi 4, tambahkan digit 0 di sebelah kiri untuk membuat grup 4 bit. Setelah itu, konversi setiap grup 4 bit menjadi bilangan desimal. Terakhir, ganti setiap bilangan desimal dalam grup dengan representasi heksadesimalnya. Dengan langkah-langkah ini, Anda dapat mengonversi bilangan biner menjadi bilangan heksadesimal dengan mudah.

   Pada contoh diatas kita menggunakan kata *"PASSWORD"* yang di enkripsi sehingga menghasilkan binner seperti dibawah ini.

Tabel 3. 5 Proses dan Hasil Konversi Biner

Proses	Hasil
Hasil Enkripsi dalam bentuk binner	00001011 11100110 01001010 11101000 10011101 11011101 00100100 11100010 00100101 01000011 01001101 11101001 01011101 01010000 00010111 00010001 00110011 10011011 10101110 11101110 00011000 11110000 00001001 10111010 10011011 01000011 01101001 10101111 00100111 11010011 00001101 01100000
Memisahkan hasil per-4bit	0000 1011 1110 0110 0100 1010 1110 1000 1001 1101 1101 1101 0010 0100 1110 0010 0010 0101 0100 0011 0100 1101 1110 1001 0101 1101 0101 0000 0001 0111 0000 0001 0001 0011 1001 1011 1010 1110 1110 0001 1000 1111 0000 0000 0010 1001 1011 1010 1001 1011 0100 0011 0110 1001 1010 1011 1101 1001 0010 0111 1101 0000 1101 0110 0000
Ubah menjadi decimal	0 11 14 6 4 10 14 8 9 13 13 13 2 4 14 2 2 5 4 3 4 13 14 9 5 13 5 5 0 1 7 0 1 1 3 9 11 10 14 14 1 8 15 0 0 2 9 11 10 9 11 4 3 6 9 10 11 13 9 2 7 13 0 13 6 0
UbahMenjadi hexadecimal	0BE64AE89DDD224E225434DE95D5017039BAEE180029BA9B4369ABD927D0D60

Maka hasil encription dalam bentuk hexadesimal dari “PASSWORD” adalah

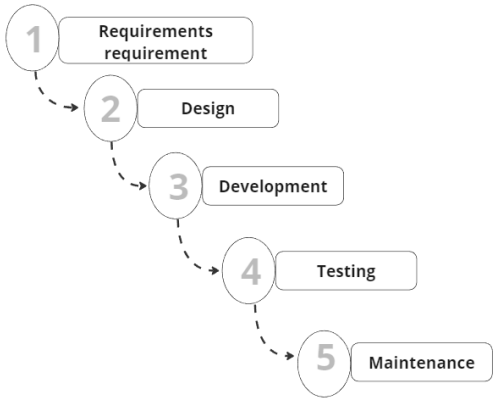
0BE64AE89DDD224E225434DE95D5017039BAEE180029BA9B4369ABD927D0D60

3.2.2 Data teks yang di enkrips

Data pengguna akan menjadi jenis data yang digunakan dalam pengujian yang mencakup informasi seperti nama, *username*, dan *password*. Proses pengujian akan

melibatkan enkripsi password saat penyimpanan data dan validasi saat proses login menggunakan formulir login yang dirancang untuk menguji dan memverifikasi keberhasilan implementasi hash 256 pada sistem keamanan *password*.

3.3 Metode Perancangan Sistem



Gambar 3. 5 *Waterfall Method*

Dalam konteks penelitian ini, peneliti memilih metode penelitian *waterfall* sebagai pendekatan utama, seiring dengan fokus eksklusif pada kegiatan penelitian. Tahapan penelitian tersebut melibatkan proses yang terstruktur seperti yang ada di gambar 3.3 di atas, dimulai dari Tahap Requirement.

Requirement mengumpulkan informasi yang relevan dan menyeluruh tentang data yang diperlukan untuk mengembangkan aplikasi keamanan teks. Data yang digunakan adalah berupa data seperti username dan password.

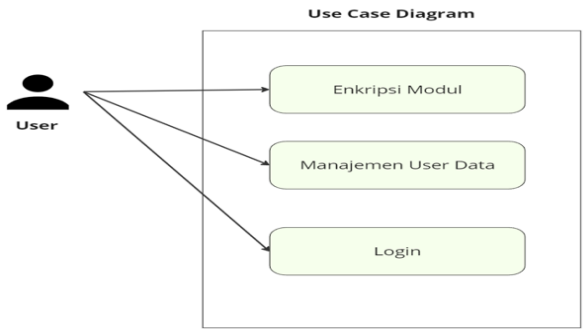
Tahap desain yaitu merancang form input, form output, form login, form user management. Serta mendesain database menggunakan SQLite yang terdiri dari table – table.

Development yaitu mengembangkan system keamanan data teks menggunakan algoritma SHA 256 terutaman algoritma sha dengan menambahkan proses pada compression fungsion xor, and, not and lalu mengambangkan output dari aplikasi hasil proses enskripsi.

Testing kemudian memverifikasi keakuratan hasil dari algoritma sha yang dapat melakukan enskripsi dengan output yang terdiri dari tiga jenis output yang berbeda dan satu hasil enskripsi dari algoritma sha.

Terakhir, maintenance memastikan program dari aplikasi ini berjalan dengan baik dan memberikan update program supaya aplikasi terus bisa mengamankan data teks dengan akurat. Dan menambahkan fitur fitur yang bisa menambah keamanan data.

3.3.1 Use Case Diagram



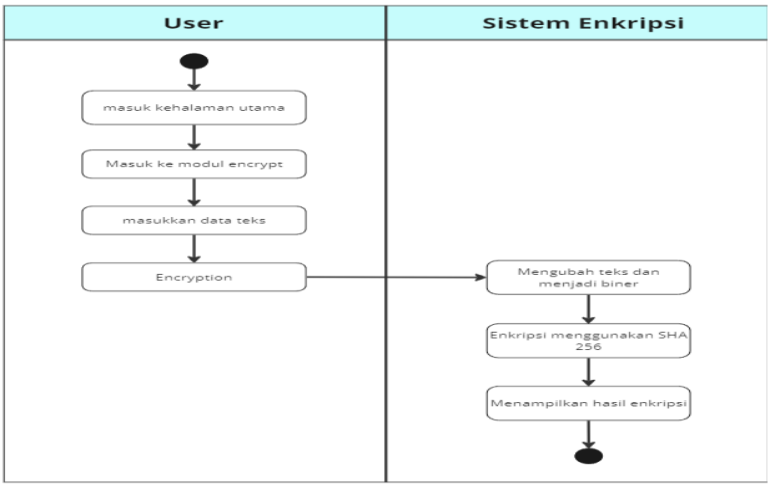
Gambar 3. 6 *Use Case Diagram*

Aplikasi ini dirancang dengan tujuan khusus, yakni menyediakan fungsionalitas enkripsi untuk keperluan analisis. Terdapat tiga modul utama yang akan dikembangkan, yaitu:

1. Enkripsi *Module* yang berfungsi untuk melakukan proses enkripsi terhadap data yang akan dianalisis.
2. *Management User Data,* dirancang untuk mengimplementasikan enkripsi ke dalam data pengguna, menyimpannya dalam database, dan mempersiapkannya untuk proses validasi selanjutnya.
3. *Log in,* bertindak sebagai modul pengujian yang akan melakukan validasi terhadap kombinasi username dan *password* yang telah dienkripsi sebelumnya. Dengan struktur tiga modul ini, diharapkan aplikasi dapat memberikan keamanan data yang optimal melalui proses enkripsi, manajemen data pengguna yang efektif, dan uji coba keamanan melalui modul *log in*.

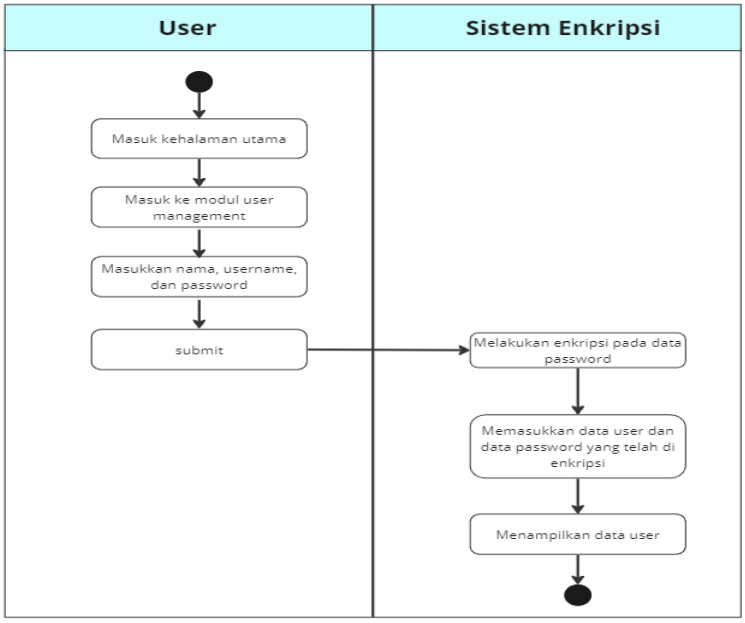
   Ketiga modul tersebut hanya akan diakses oleh 1 *user*. Sehingga akan membentuk diagram *Use Case* seperti gambar 3.4 di atas.

3.3.2 Activity Diagram



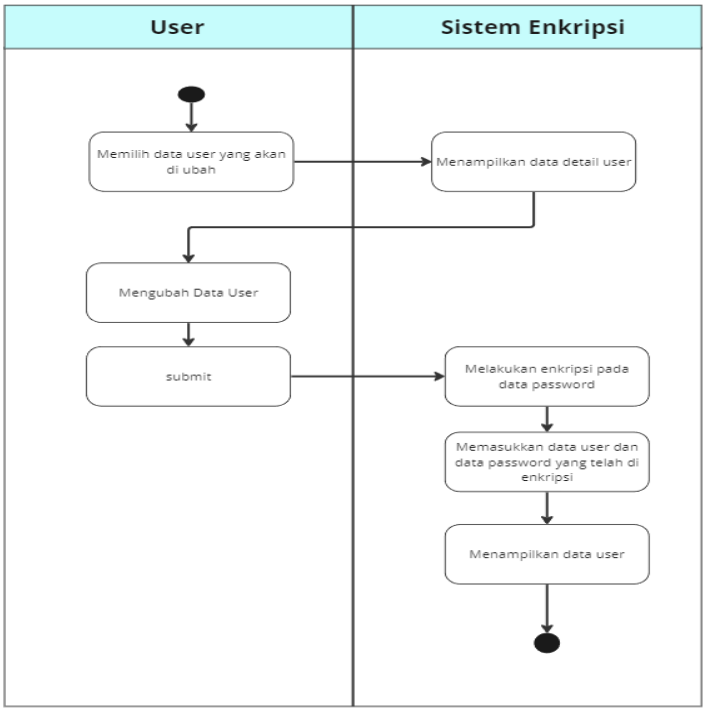
Gambar 3. 7 *Activity Diagram* Modul Enkripsi

Aplikasi ini memungkinkan pengguna memasukkan teks. Sistem akan mengubahnya menjadi biner, kemudian melakukan enkripsi sesuai dengan algoritma *SHA-256*, dan menampilkan hasil enkripsi.



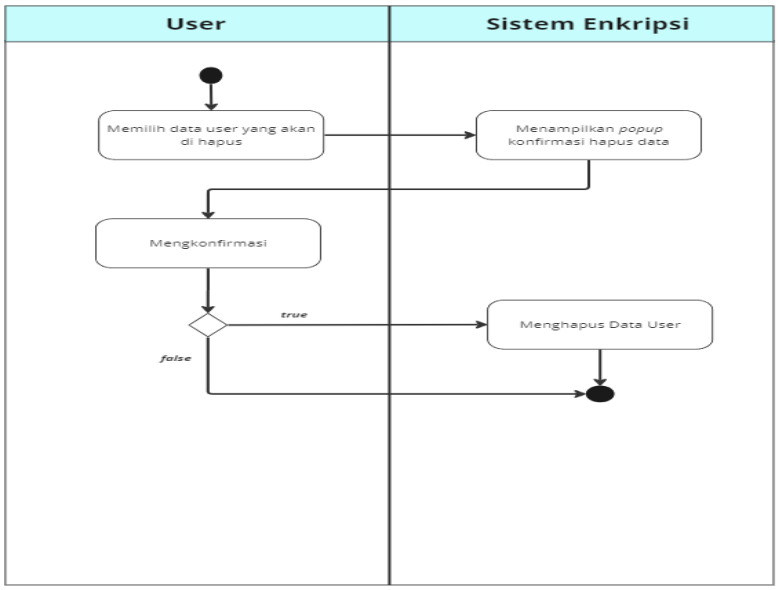
Gambar 3. 8 *Activity Diagram* Tambah *User*

Pengguna dapat memasukkan data user termasuk *password*, dan sistem akan melakukan enkripsi pada *password*, menyimpan data pengguna dan data *password* yang telah dienkripsi.



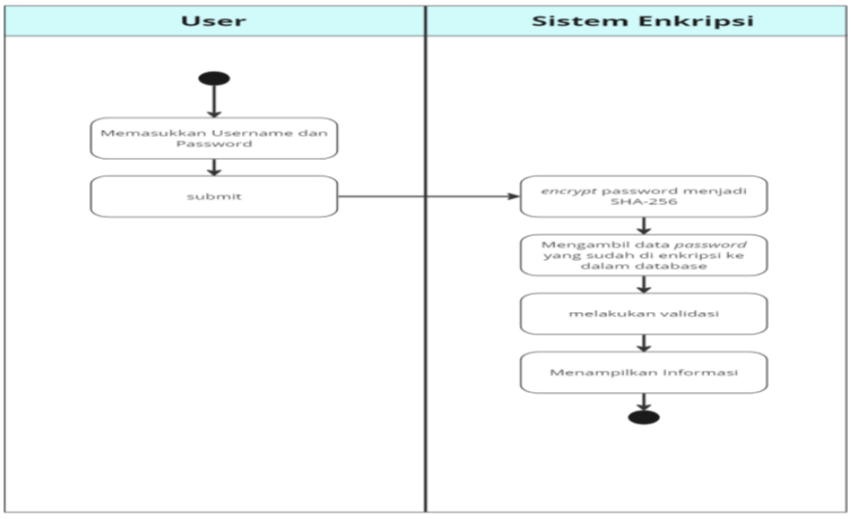
Gambar 3. 9 *Activity Diagram Edit User*

Pengguna dapat memilih data *user* yang akan di ubah. Kemudian sistem akan menampilkan data *user* yang dipilih. Pengguna dapat mengubah data *user* termasuk *password*, dan sistem akan melakukan enkripsi pada *password*, menyimpan data pengguna dengan *password* yang telah dienkripsi.



Gambar 3. 10 *Activity Diagram* Hapus *User*

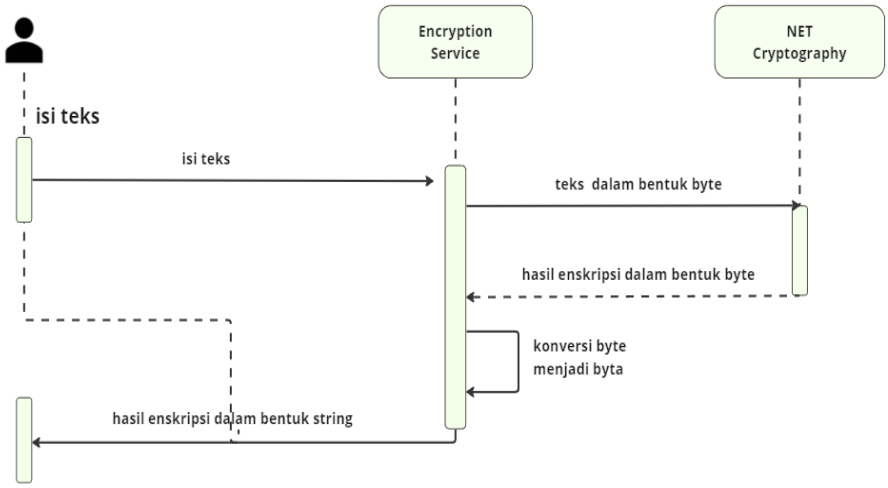
Pengguna dapat memilih data *user* yang akan di hapus. Kemudian system akan menampilkan option untuk mengkonfirmasi penghapusan data *user*. Jika di konfirmasi ‘Ya’ maka sistem akan menghapus data *user*. Jika tidak, maka sistem tidak melakukan apapun.



Gambar 3. 11 *Activity Diagram Log in*

Pengguna dapat memasukkan *username & password*. Sistem akan mengubahnya data *"password"* menjadi *byte*, kemudian mengonversi ke *SHA-256*, dan menampilkan hasil enkripsi. Setelah di enkripsi sistem akan melakukan validasi dengan data yang ada di *database*.

3.3.3 Sequence Diagram

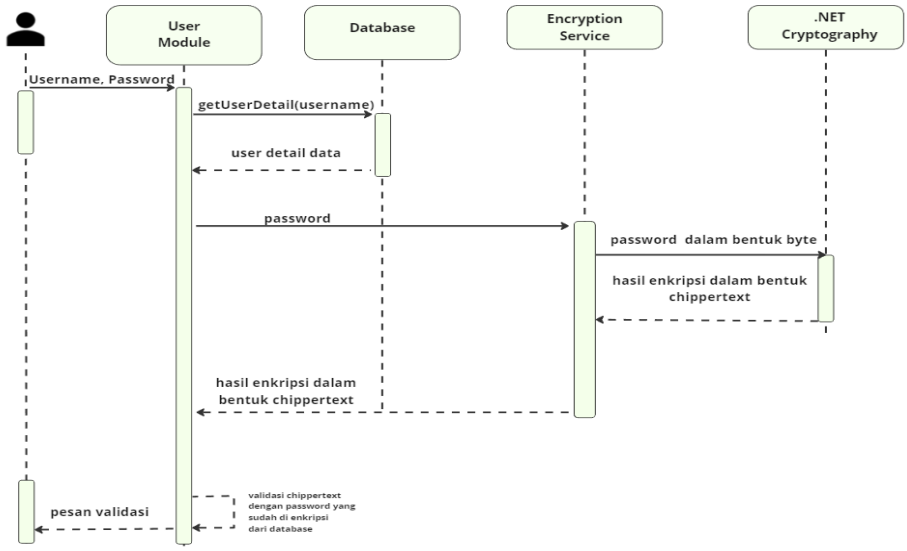


Gambar 3. 12 *Sequence Diagram* Modul Enkripsi

Dalam implementasi aplikasi menggunakan .NET (C#), algoritma *SHA-256* sudah tersedia dalam *package Cryptography* dan dapat dengan mudah diakses. Yang perlu kita lakukan adalah menyediakan data yang akan dienkripsi dalam bentuk *byte*, sehingga perlu melakukan konversi dari teks terlebih dahulu kedalam bentuk *byte*. kemudian, *package Cryptography* akan implementasi algoritma *SHA-256* yaitu mengenkripsi data tersebut sesuai dengan standar yang telah ditentukan. Dengan menggunakan fungsi-fungsi yang telah disediakan oleh .NET, proses

enkripsi dapat diintegrasikan secara efisien dalam aplikasi kita tanpa perlu mengimplementasikan algoritma *SHA-256* secara manual.

Jika diurutkan kedalam *sequence* yaitu pertama *User* dapat mengirim isi teks (yang akan di enkripsi) dan *secret key* yang digunakan untuk menambah keamanan pada saat enkripsi data. Kemudian *Encryption service* yaitu servis yang dibuat oleh penulis untuk mengubah data *text* menjadi *byte* sehingga dapat diproses oleh *package Cryptography* untuk dilakukan enkripsi.



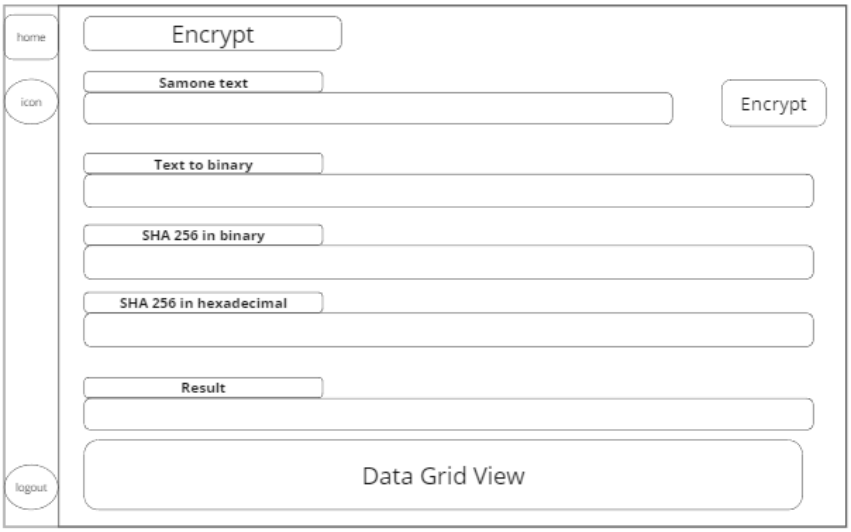
Gambar 3. 13 Sequence Diagram Modul Login

Dalam penerapan validasi sendiri, *User* dapat mengirimkan data *username & password.* Kemudian *password* akan di enkripsi terlebih dahulu sebelum dilakukan validasi oleh data yang ada pada *database*.

3.3.4 Perancangan *form input dan output*

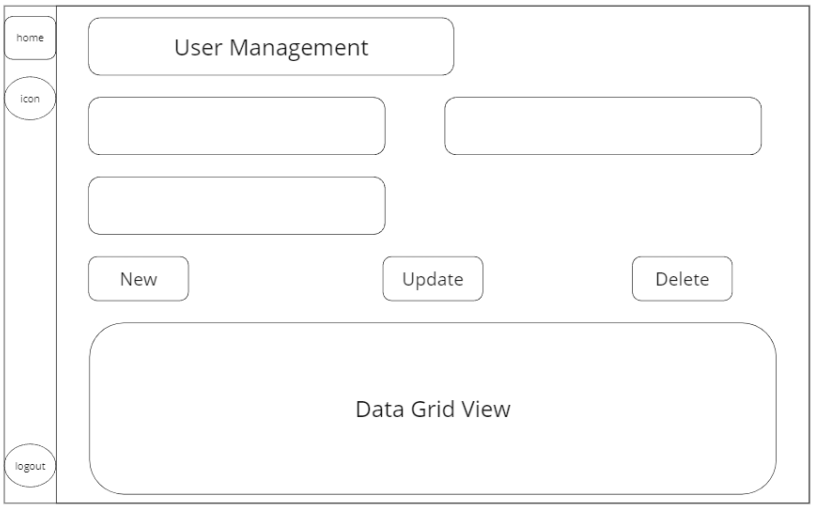
Dalam perancangan *form*, penulis membuat 3 bentuk *form* dengan tujuan yang berbeda yaitu *Form encrypt, Form user data*, dan *Form Login.*

Form encrypt bertujuan untuk melakukan uji coba pada *test* uji coba pada kalimat yang ingin di enkripsi dan hasilnya. *Form encrypt* sendiri memiliki 2 input yaitu teks yang ingin di lakukan hash. Kemudian menghasilkan *output SHA-256.*



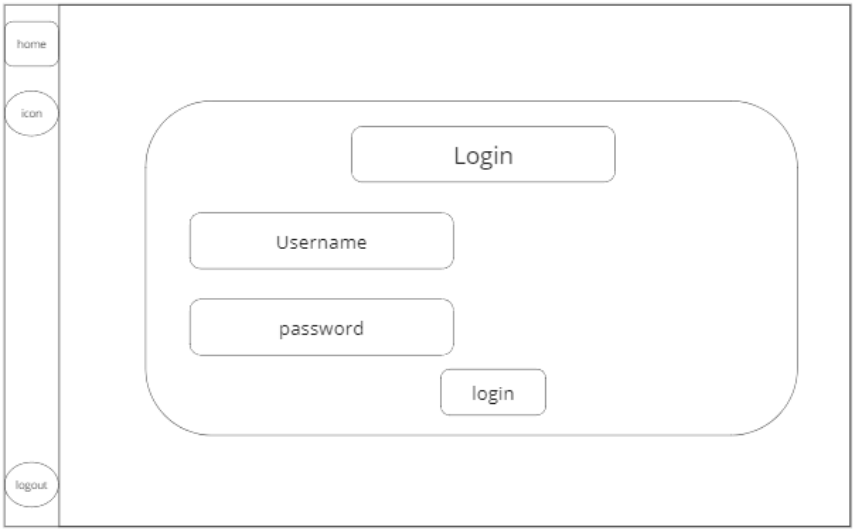
Gambar 3. 14 *Form Encrypt*

Form User bertujuan untuk melakukan manajemen *user* data dan melakukan enkripsi pada data *password.* data hasil enkripsi ini kemudian disimpan ke dalam *database* dan akan digunakan nanti pada *form log in*.



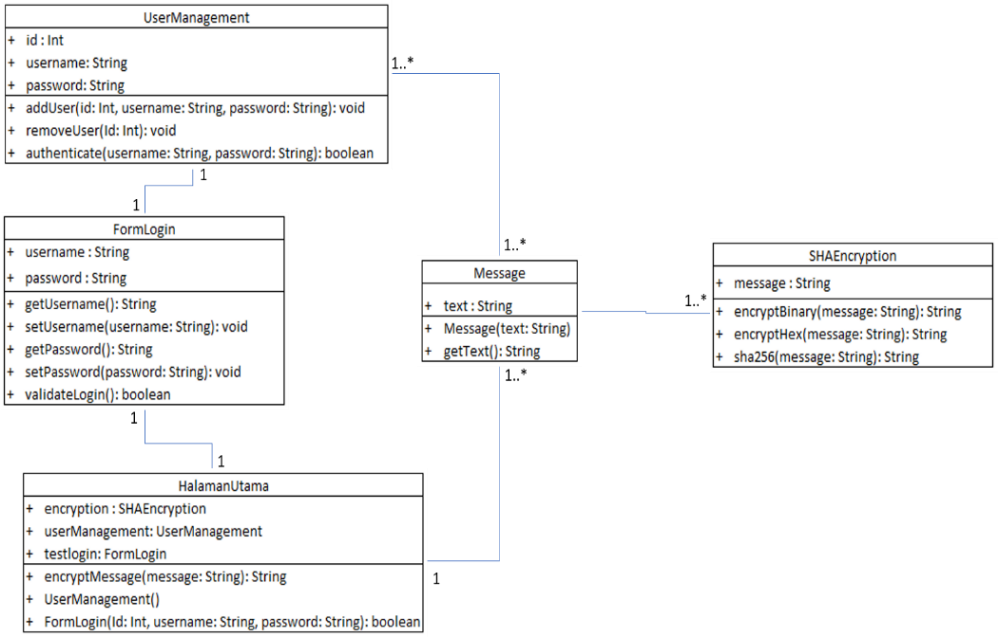
Gambar 3. 15 *Form User*

Form Log in bertujuan untuk melakukan validasi terhadap data *password* yang telah disimpan. Disini *user* harus memasukkan *username* dan *password* dan akan menampilkan pesan berhasil atau gagal di validasi.



Gambar 3. 16 *Form Log in*

3.3.5 Class Diagram



Gambar 3. 17 *Class Diagram*

Diagram kelas menggambarkan struktur objek sistem. Diagram ini menunjukan kelas objek yang menyusun sistem dan hubungan antara kelas objek tersebut. *Class diagram* suatu bentuk penggambaran penggambaran struktur dan deskripsi *class*, *package* dan objek beserta satu sama lain. Diagram kelas juga menunjukan properti dan operasi dari sebuah kelas dan batasan-batasan yang terdapat dalam hubungan tersebut. *Class diagram* merupakan gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan kelas-kelas objek yang menyusun sebuah sistem dan juga hubungan antara kelas objek tersebut.

3.4 Metode Pengujian Sistem

Dalam melakukan pengujian sistem untuk aplikasi ini, penulis menggunakametode *black-box testing. Black-box testing* memberikan perhatian utama pada kinerja perangkat lunak tanpa perlu terlalu merinci logika atau bagian dalam implementasinya. Dengan metode ini, penulis bisa membuat skenario pengujian yang dapat memenuhi semua persyaratan fungsional dan kinerja tanpa harus terlibat dalam detail teknis bagaimana sistem dijalankan. Dengan memilih *black-box testing*, penulis dapat mengevaluasi respons sistem terhadap berbagai situasi masukan dan memastikan bahwa semua aspek yang terlihat oleh pengguna sesuai dengan harapan. Pendekatan ini memungkinkan pengujian lebih berfokus pada pengalaman pengguna dan kehandalan sistem secara keseluruhan.

Tabel 3. 6 Daftar Pengujian sistem

No	Bagian Pengujian	Tahapan Pengujian
1	Pengujian halaman utama	- Buka aplikasi "Encrypt Me"
2	Pengujian Modul Encrypt	-Mengosongkan form teks yang akan di enkripsi -Mengisi teks secret key yang akan di enkripsi
3	Pengujian Modul User Management	- Membuat user baru - Mengubah user baru - Menghapus data user
4	Pengujian Modul Login	-Login dengan username dan password
5	Pengujian Halaman Informasi	-Membuka halaman informasi dari halaman utama

3.5 Tempat dan Waktu penelitian

3.5.1 Tempat penelitian

Penelitian ini akan dilakukan di piayu kecamatan sei beduk Kota Batam Kepulauan Riau.

3.5.2 Waktu Penelitian

Waktu penelitian ini dimulai dari bulan September hingga selesai.

Tabel 3. 7 Agenda Penelitian

Kegiatan	Waktu Kegiatan																			
	September (2023)				Oktober (2023)				November (2023)				Desember (2023)				Januari (2024)			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengajuan Judul																				
Penyusunan Bab I																				
Penyusunan Bab II																				
Penyusunan Bab III																				
Penyusunan Bab IV																				
Penyusunan Bab V																				
Revisi Bab I - V																				
Pengum pulan Skripsi																				

Sumber: (Data Penelitian, 2023)

Pada bulan September penulis mulai pengajuan judul setelah pengajuan judul penulis mulai mencari referensi berupa buku dan jurnal yang terkait dengan keamanan data dan SHA. Dan penguji melanjutkan membuat bab satu sampai bab tiga dan setelah selesai penulis melanjutkan ke bab empat dan bab lima.