

## BAB III

### ANALISIS MASALAH DAN DESAIN SISTEM

#### III.1. Analisis Masalah

Berdasarkan masalah yang sudah penulis buat, disini penulis akan menjelaskan tentang permasalahan yang ada.

1. Database Tidak aman.

Mengapa disebut tidak aman? Yang penulis maksud tidak aman disini bukan berarti database MySQL, justru dengan MySQL administrator dapat menentukan siapa-siapa pihak yang berhak untuk mendapatkan akses penuh pada MySQL, sehingga akses terhadap data-data yang bersifat privasi dapat diminimalisir dengan kendali sistem tersebut. Namun itu semua sifatnya hanya untuk meminimalisir saja, tetap saja jika data tersebut tidak dienkripsi, akan tetap ada saja orang-orang yang mampu menembus sistem keamanan tersebut, dan ketika itu orang tersebut dapat dengan mudah mengetahui isi dari database tersebut.

2. Metode *Vigenere Cipher* tidak bisa digunakan untuk mengenkripsi dan mendekripsikan karakter selain karakter huruf alfabet, baik itu angka ataupun semua karakter yang terdapat dalam *Code ASCII*.

Untuk kasus yang kedua ini, sudah jelas bahwa metode *Vigenere Cipher* hanya diperuntukkan untuk mengenkripsi dan mendekripsikan karakter huruf alfabet saja. Bahkan tidak memberikan perbedaan terkait huruf besar atau huruf kecil.

### III.1.1. Strategi Penyelesaian Masalah

Disini penulis akan menjelaskan stratei penyelsaian masalah berdasarkan dari masalah yang ada

#### 1. Database tidak aman

Untuk strategi penyelesaian masalah pada Database yang tidak aman adalah penulis akan menggunakan metode enkripsi dan dekripsi menggunakan vigenere cipher untuk memberikan pengamanan pada database.

#### 2. Metode *Vigenere Cipher* tidak bisa digunakan untuk mengenkripsi dan mendekripsikan karakter selain karakter huruf alfabet, baik itu angka ataupun semua karakter yang terdapat dalam *Code ASCII*.

Untuk kasus yang kedua solusinya adalah penulis akan membuat aplikasi menggunakan referensi yang penulis dapatkan untuk membuat agar nantinya vigenere cipher mampu melakukan enkripsi dan dekripsi untuk karakter angka, huruf dan simbol yang terdapat dalam Code ASCII.

### III.2. Penerapan Metode

Dalam penerapan metode vigenere cipher dengan code ASCII ada beberapa perubahan yang dilakukan mulai dari perubahan bagian mod, mengubah nilai dari karakter yang dimasukkan menjadi nilai desimal sesuai dengan code ASCII serta dilakukan pergeseran pada kunci (key text). Namun dikarenakan dari rumus yang ada pada referensi yang penulis dapatkan kurang efektif dan juga dikarenakan adanya pembatasan yang penulis buat pada penulisan skripsi ini, untuk karakter yang akan dienkrpsi hanya sebatas karakter angka, huruf dan simbol, yaitu mulai dari nomor 32-126, dan untuk karakter yang didekripsikan

adalah mulai dari nomor 0-126. Maka dari itu penulis pun melakukan sedikit perubahan pada rumus yang terdapat pada metodenya.

Berdasarkan jurnal acuan yang penulis gunakan yaitu oleh Khoirul Amri, Anugrah Rizki E., Herman Sah Putra S., Sura Purna A. S., dan Suryadi Sudirja, pada tahun 2015 dalam jurnalnya yang berjudul “Aplikasi Enkripsi Dan Dekripsi Menggunakan Algoritma *Vigenere Cipher* ASCII Berbasis Java”. Berdasarkan contoh yang mereka buat, dari segi rumusnya mereka hanya melakukan perubahan pada modnya saja tetapi untuk programnya mereka melakukan shift (pergeseran) dengan cara pengurangan sebanyak 97 digit, tidak ada alasan pasti disebutkan oleh para penyusunnya.

Berdasarkan jurnal tersebut penulis membuat sedikit perubahan agar *vigenere cipher* benar-benar bisa digunakan untuk melakukan enkripsi karakter Dpada nomor 32-127 yang terdapat dalam kode ASCII. Dan melakukan dekripsi pada karakter 0-127. Dan untuk lebih jelasnya dapat dilihat perubahan dan penjelasannya sebagai berikut :

Enkripsi

$$C_i = (P_i + K_i) \bmod 128$$

Dekripsi

$$P_i = (C_i - K_i) \bmod 128$$

Contoh :

Plaintext : Makan

Key : Nasi

- $M = 77$     $N = 78$

$$C1 = (77 + 78) \bmod 128$$

$$= 155 \bmod 128$$

$$= 27$$

Karakter dari Code ASCII 27 adalah “ ESC ”

$$P1 = (27 - 78) \bmod 128$$

$$= -51 + 128$$

$$= 77$$

Karakter dari Code ASCII 77 adalah “ M “

- $a = 97$      $a = 97$

$$C2 = (97 + 97) \bmod 128$$

$$= 194 \bmod 128$$

$$= 66$$

Karakter dari Code ASCII 67 adalah “ B “

$$P2 = (66 - 97) + 128$$

$$= (-31) + 128$$

$$= 97$$

Karakter dari Code ASCII 97 adalah “ a “

- $k = 107$      $s = 115$

$$C3 = (107 + 115) \bmod 128$$

$$= 222 \bmod 128$$

$$= 94$$

Karakter dari Code ASCII 94 adalah “ ^ “

$$P3 = (94 - 115) + 128$$

$$= (-21) + 128$$

$$= 107$$

Karakter dari Code ASCII 107 adalah “ k “

- $a = 97 \quad i = 105$

$$C4 = 97 + 105 \text{ mod } 128$$

$$= 202 \text{ mod } 128$$

$$= 74$$

Karakter dari Code ASCII 75 adalah “ J “

$$P4 = (74 - 105) + 128$$

$$= (-31) + 128$$

$$= 97$$

Karakter dari Code ASCII 97 adalah “ a “

- $n = 110 \quad N = 78$

$$C5 = (110 + 78) \text{ mod } 128$$

$$= 188 \text{ mod } 128$$

$$= 60$$

Karakter dari Code ASCII 68 adalah “ C “

$$P5 = (60 - 78) + 128$$

$$= (-18) + 128$$

$$= 110$$

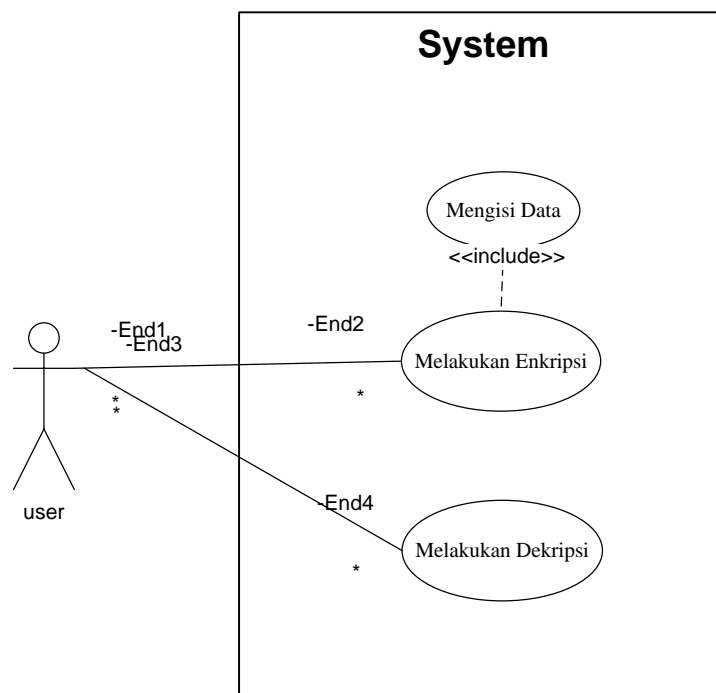
Karakter dari Code ASCII 110 adalah “ n “

### III.3. Desain Sistem

Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram*.

#### III.3.1. Usecase Diagram

Sistem yang akan dirancang digambarkan suatu kegiatan aktivitas yang dilakukan oleh user dan admin dengan *Usecase Diagram* yang terdapat pada Gambar III.2 berikut :



**Gambar III.1. Usecase Diagram**

Adapun penjelasan dari Usecase Diagram diatas adalah sebagai berikut :

1. User mengisi data

Sebelum melakukan enkripsi, terlebih dahulu user mengisi data yang akan di enkripsi

2. User melakukan enkripsi

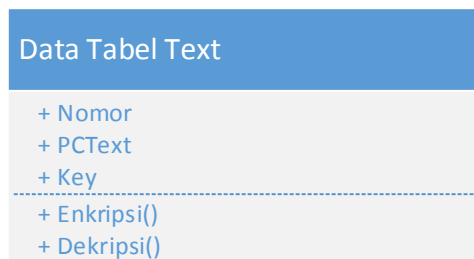
Pada tahap ini user akan melakukan enkripsi atau perubahan pada data (Plaintext) yang sudah diisi sebelumnya pada kolom Text(Plaintext/Cipher Text) ke dalam bentuk cipher text.

3. User melakukan dekripsi

Pada tahap ini user akan melakukan perubahan kembali text yang sebelumnya telah terenkripsi menjadi text dalam bentuk asal atau yang disebut dengan Plaintext.

### III.3.2. Class Diagram

Untuk rancangan Class Diagram atau databasenya dapat dilihat pada gambar dibawah ini :



**Gambar III.2. Class Diagram**

Keterangan :

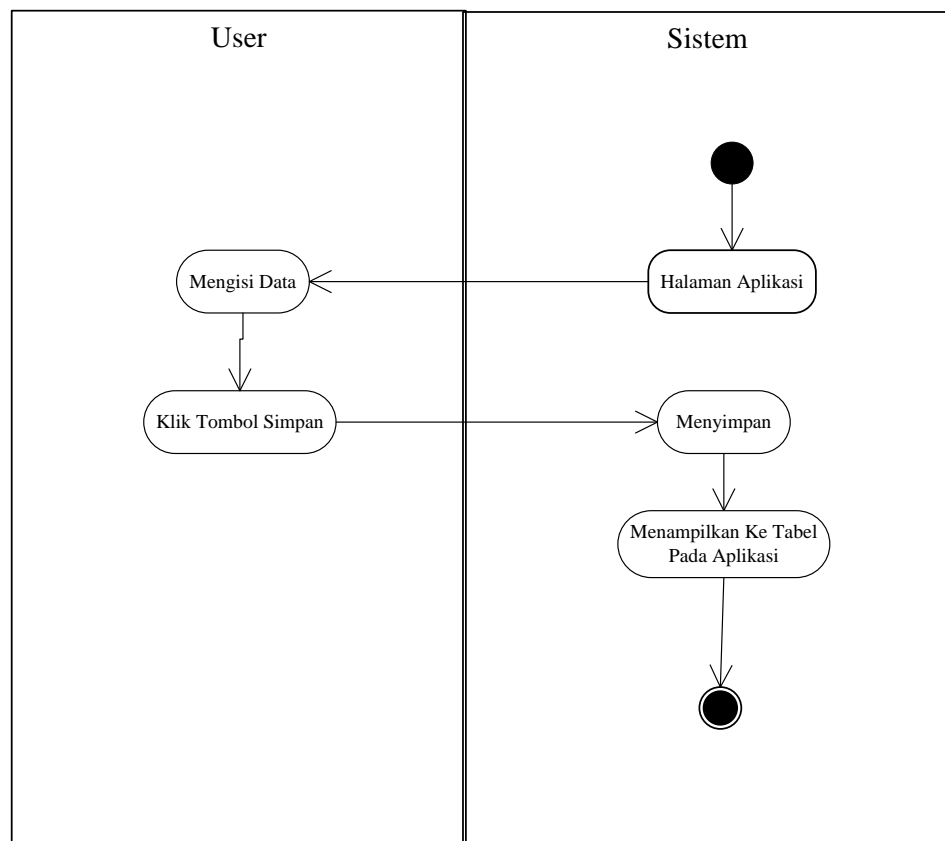
1. Pada aplikasi ini hanya terdiri dari satu tabel class diagram saja dengan nama Tabel Data Text sebagai nama Class.
2. Pada tabel tersebut terdiri dari 3 atribut yaitu Nomor, PCText, dan Keytext.
3. Yang menjadi primary key adalah atribut Nomor.

4. Pada atribut PCText, adalah bagian yang akan menjadi Plaintext atau Cipher Text
5. Atribut Key digunakan sebagai kunci untuk proses enkripsi dan dekripsi.
6. Pada bagian operasi terdapat 2 jenis operasi yaitu Enkripsi dan Dekripsi.

### III.3.3. Activity Diagram

Untuk lebih memperinci penjelasan dari Usecase yang sebelumnya dibuat penulis menggunakan Activity Diagram, dan berikut penjelasannya.

1. Activity Diagram Isi Data

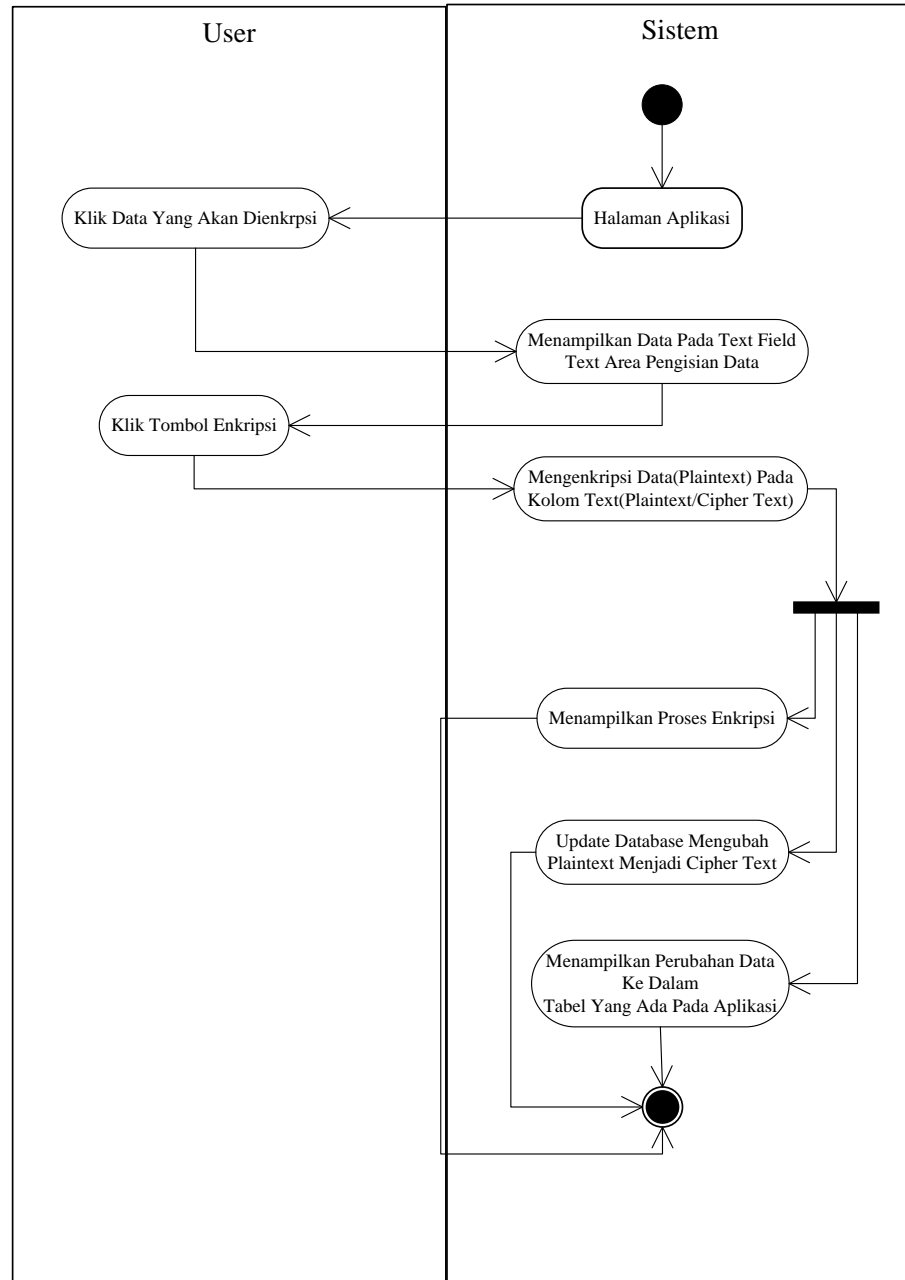


**Gambar III.3. Activity Diagram Isi Data**

Keterangan :

- a. Pada halaman aplikasi terdapat 1 *jtextfield*, dimana di *jtextfield* tersebut *User* mengisikan Nomor, Text (P/C) yang akan menjadi *Plaintext*, beserta kuncinya atau *Key Text*.
- b. Setelah itu *User* melakukan klik pada tombol simpan.
- c. Setelah diklik tombol enkripsi, Sistem akan langsung menyimpannya kedalam database dan menampilkanya

## 2. Activity Diagram Enkripsi



**Gambar III.4. Activity Diagram Enkripsi**

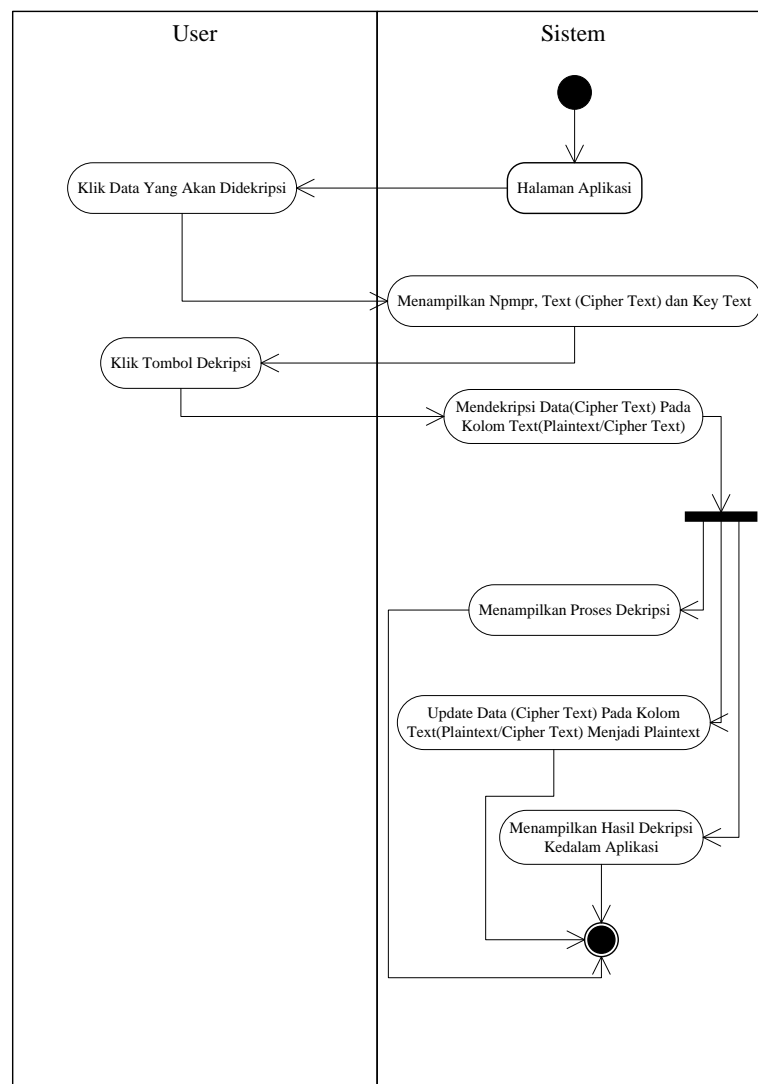
Keterangan :

- a. User melakukan klik pada data yang ingin dienkrpsi yang ada pada tabel tersebut, lalu secara otomatis Nomor, Plaintext dan Key Text akan

muncul pada *Text Field* dan *Text Area* pengisian data.

- b. Lalu user melakukan klik pada tombol enkripsi.
- c. Kemudian sistem akan melakukan enkripsi pada *Plaintext* yang ada pada bagian kolom *Text (Plaintext/Cipher Text)*.
- d. Sistem pun akan menampilkan proses enkripsi sekaligus melakukan update pada data tersebut, dan menampilkan kembali isi database setelah di update pada tabel yang ada pada aplikasi.

### 3. Activity Diagram Dekripsi



**Gambar III.5. Activity Diagram Dekripsi**

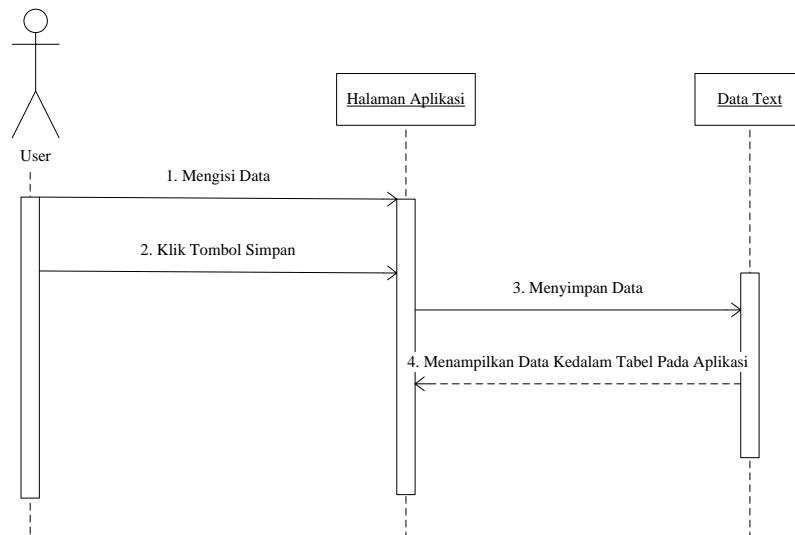
Keterangan :

- a. Pada aplikasi terdapat tabel dimana pada tabel tersebut adalah tampilan dari apa yang ada didalam database.
- b. User melakukan klik pada data yang ingin didekripsikan yang ada pada tabel tersebut, lalu secara otomatis Nomor, cipher text dan key text akan muncul pada *Text Field*, dan *Text Area*.
- c. Lalu user melakukan klik pada tombol dekripsi.
- d. Kemudian *Cipher Text* pada kolom *Text (Plaintext/Cipher Text)* akan langsung terdekripsi.
- e. Sistem pun akan menampilkan proses dekripsi sekaligus melakukan update pada data tersebut, dan menampilkan kembali isi database setelah di update.

#### **III.3.4. Sequence Diagram**

Adapun bentuk rancangan *sequence diagram* yang penulis rancang adalah sebagai berikut:

## 1. Sequence Diagram Isi Data

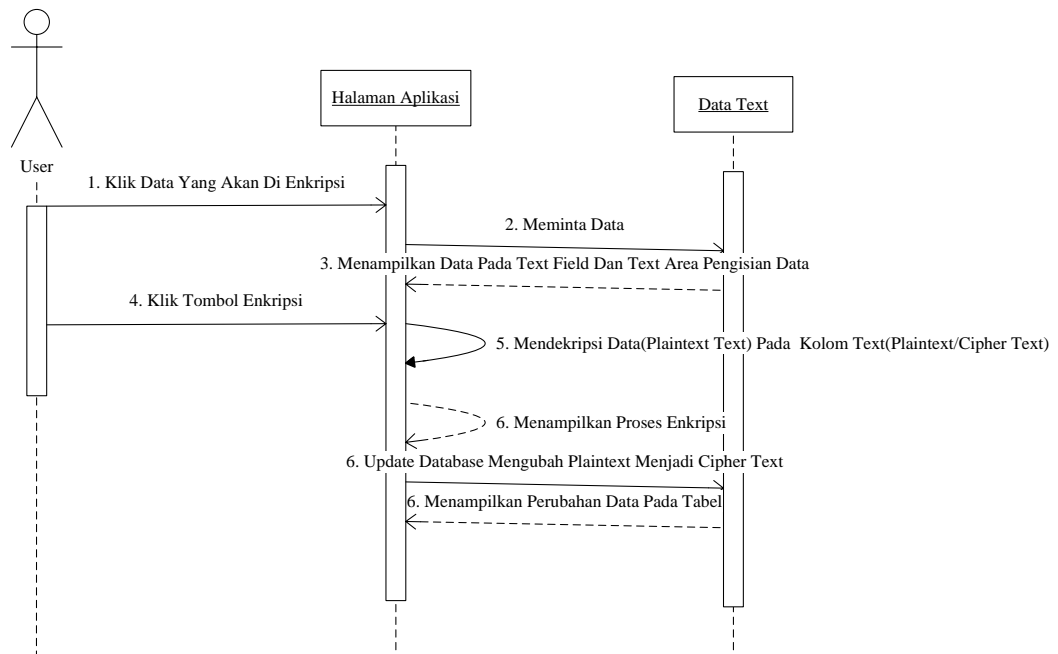


**Gambar III.6. Sequence Diagram Isi Data**

Keterangan :

- Pada tampilan interface halaman aplikasi user mengisi data yaitu berupa Nomor, Plaintext, dan key text
- Setelah itu User melakukan klik pada tombol Simpan.
- Setelah diklik tombol simpan, Sistem akan langsung menyimpannya kedalam database dan menampilkan isi database pada tabel yang ada dibagian bawah.

## 2. Sequence Diagram Enkripsi

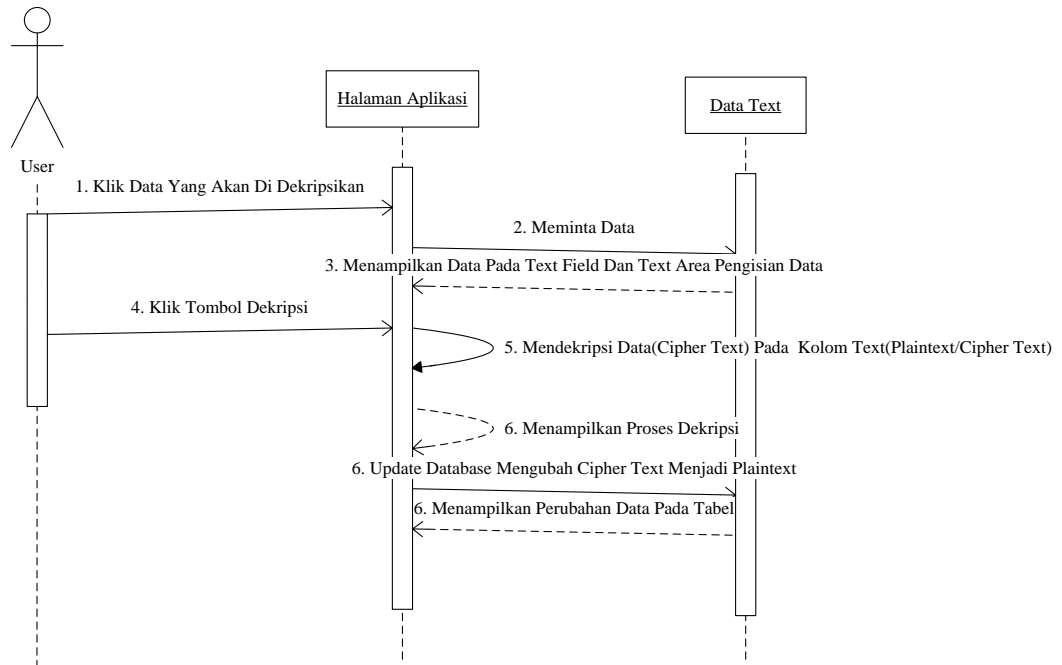


**Gambar III.7. Sequence Diagram Enkripsi**

Keterangan :

- a. User melakukan klik pada data yang ingin dienkripsi yang ada pada tabel tersebut, lalu secara otomatis Nomor, Plaintext dan Key Text akan muncul pada *Text Field* dan *Text Area* pengisian data.
- b. Lalu user melakukan klik pada tombol enkripsi.
- c. Kemudian sistem akan melakukan enkripsi pada *Plaintext* yang ada pada bagian kolom *Text (Plaintext/Cipher Text)*.
- d. Sistem pun akan menampilkan proses enkripsi sekaligus melakukan update pada data tersebut, dan menampilkan kembali isi database setelah di update pada tabel yang ada pada aplikasi.

### 3. Sequence Diagram Dekripsi



**Gambar III.8. Sequence Diagram Dekripsi**

Keterangan :

- a. Pada aplikasi terdapat tabel dimana pada tabel tersebut adalah tampilan dari apa yang ada didalam database.
- b. User melakukan klik pada data yang ingin didekripsikan yang ada pada tabel tersebut, lalu secara otomatis Nomor, cipher text dan key text akan muncul pada *Text Field*, dan *Text Area*.
- c. Lalu user melakukan klik pada tombol dekripsi.
- d. Kemudian *Cipher Text* pada kolom *Text (Plaintext/Cipher Text)* akan langsung terdekripsi.

- e. Sistem pun akan menampilkan proses dekripsi sekaligus melakukan update pada data tersebut, dan menampilkan kembali isi database setelah di update.

### III.4. Desain User Interface

Disini penulis akan memperlihatkan rancangan dari desain user interface atau rancangan tampilan aplikasi yang akan dibuat. Dapat dilihat pada gambar dibawah ini :

No.	Text (Plaintext)	Cipher Text	Key Text

**Gambar III.9. Desain User Interface Aplikasi**

Keterangan :

1. Text Field pengisian nomor, nomor ini nantinya akan digunakan sebagai primary key.

2. Text Area Pengisian Text (P/C), disini nantinya akan diisi Plaintext (P) yang akan di enkripsi, dan juga sekaligus nantinya Cipher Text akan ditampilkan disini pada saat pendekripsian.
3. Text Area Pengisian Key Text, disini nantinya akan diisikan Key Text sebagai kunci dari enkripsinya.
4. Tombol Simpan digunakan untuk menyimpan data yang ada pada Text Field dan Text Area pengisian data.
5. Tombol Enkripsi digunakan untuk mengupdate sekaligus mengenkripsi data Plaintext yang telah ditampilkan pada Text Field dan Text Area pengisian data.
6. Tombol Dekripsi digunakan untuk mengupdate sekaligus mendekripsi data Cipher Text yang telah ditampilkan pada Text Field dan Text Area pengisian data.
7. Tombol Bersihkan digunakan untuk membersihkan Text Field dan seluruh Text Area.
8. Tabel untuk menampilkan isi dari tabel database.
9. Text Area Proses, disini nantinya akan menampilkan proses dari enkripsi dan dekripsinya.