BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisis Masalah

PT. Permodalan Nasional Madani (Persero) atau PNM adalah perusahaan jasa keuangan yang hadir sebagai solusi peningkatan kesejahteraan melalui layanan pinjaman modal untuk perempuan prasejahtera pelaku usaha ultra mikro melalui program Membina Ekonomi Keluarga Sejahtera (PNM Mekaar) dan dilakukan secara berkelompok. Masalah keamanan dan kerahasiaan terhadap data nasabah pada PT. PNM yang disimpan dalam *database* merupakan hal yang sangat penting. Keamanan pada *database* dengan pembatasan hak akses sudah tidak lagi dapat menjamin keamanan data karena kebocoran data dapat disebabkan oleh pihak-pihak yang langsung berhubungan dengan *database*. Salah satu cara untuk mengamankan sebuah data adalah dengan menggunakan kriptografi. Oleh karena itu peneliti merekomendasikan algortima RC4 yang merupakan algoritma kriptografi kunci simetris dan bersifat *stream cipher* sehingga panjang karakter hasil enkripsi (*ciphertext*) mempunyai panjang karakter yang sama dengan data asli (*plaintext*). Maka dari itu peneliti membuat sebuah penerapan kriptografi menggunakan algoritma RC4 pada data nasabah PT. PNM. Proses enkripsi pada penelitian ini dilakukan dengan menentukan kolom pada *database* yang akan dienkripsi. Implementasi ini menerapkan metode sistem enkripsi dan dekripsi dengan implementasi algoritma RC4 dalam pengamanan data nasabah.

III.1.1. Strategi Pemecahan Masalah

Strategi dalam melakukan pemecahan masalah yang sedang dianalisa oleh penulis mengenai implementasi algoritma RC4 dalam melakukan pengamanan data nasabah pada PT. Permodalan Nasional Madani adalah sebagai berikut :

1. Dengan menerapkan teknik kriptografi pada data nasabah PT. PNM maka dapat mengatasi masalah keamanan data nasabah.
2. Merancang dan membangun sebuah aplikasi pengamanan data nasabah pada PT. PNM dengan implementasi algoritma RC4.

III.2. Penerapan Algoritma RC4

Algoritma RC4 mengenkripsi antara kombinasi *plaintext* dengan menggunakan *bit-wise* Xor (*Exclusive-or*). RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan *plaintext* untuk menghasilkan *ciphertext*. Masing-masing elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsinya dilakukan dengan cara yang sama (karena XOR merupakan fungsi simetris).

Untuk menghasilkan *keystream*, *cipher* menggunakan *state internal* yang meliputi dua bagian :

1. Tahap *key scheduling* dimana *state automaton* diberi nilai awal berdasarkan kunci enkripsi. *State* yang diberi nilai awal berupa *array* yang merepresentasikan suatu permutasi dengan 256 elemen, jadi hasil dari algoritma KSA adalah permutasi awal. KSA atau *Key Scheduling Algorithm* digunakan untuk menginisialisasi permutasi dalam *array* S. *Array* yang

mempunyai 256 elemen ini (dengan indeks 0 sampai dengan 255) dinamakan S. Berikut adalah algoritma KSA dalam bentuk *pseudo-code* dimana *key* adalah kunci enkripsi dan *keylength* adalah besar kunci enkripsi dalam *bytes* (untuk kunci 128 bit, *keylength* = 16).

```
for i = 0 to 255
  S[i] := i
j := 0
for i = 0 to 255
  j := (j + S[i] + key[i mod keylength]) mod 256
  swap(S[i],S[j])
```

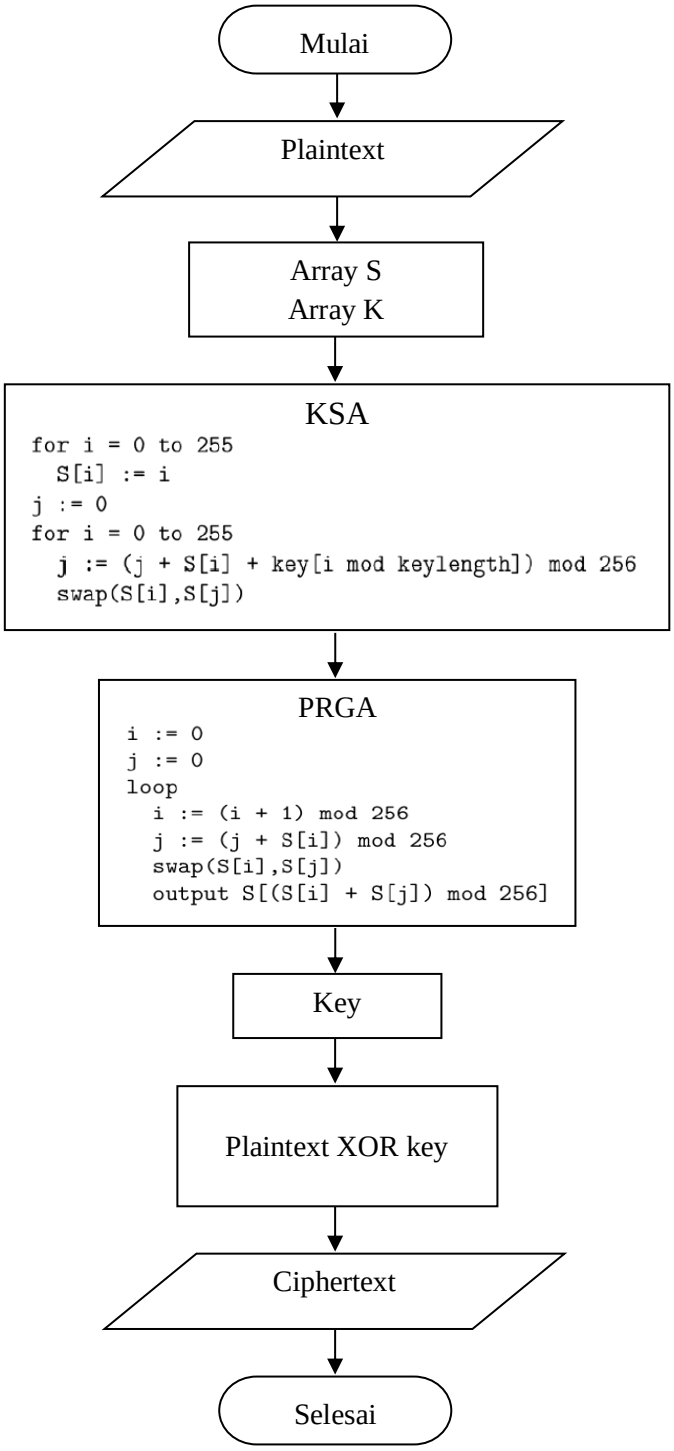
2. Tahap *pseudo-random generation* dimana *state automaton* beroperasi dan *output*nya menghasilkan *keystream*. Setiap putaran, bagian *keystream* sebesar 1 *byte* (dengan nilai antara 0 sampai dengan 255) di *output* oleh PRGA berdasarkan *state* S. Berikut adalah algoritma PRGA dalam bentuk *pseudo-code*.

```
i := 0
j := 0
loop
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap(S[i],S[j])
  output S[(S[i] + S[j]) mod 256]
```

Setelah terbentuk *keystream,* kemudian *keystream* tersebut dimasukkan dalam operasi XOR dengan *plaintext* yang ada, dengan sebelumnya pesan dipotong-potong terlebih dahulu menjadi *byte-byte*.

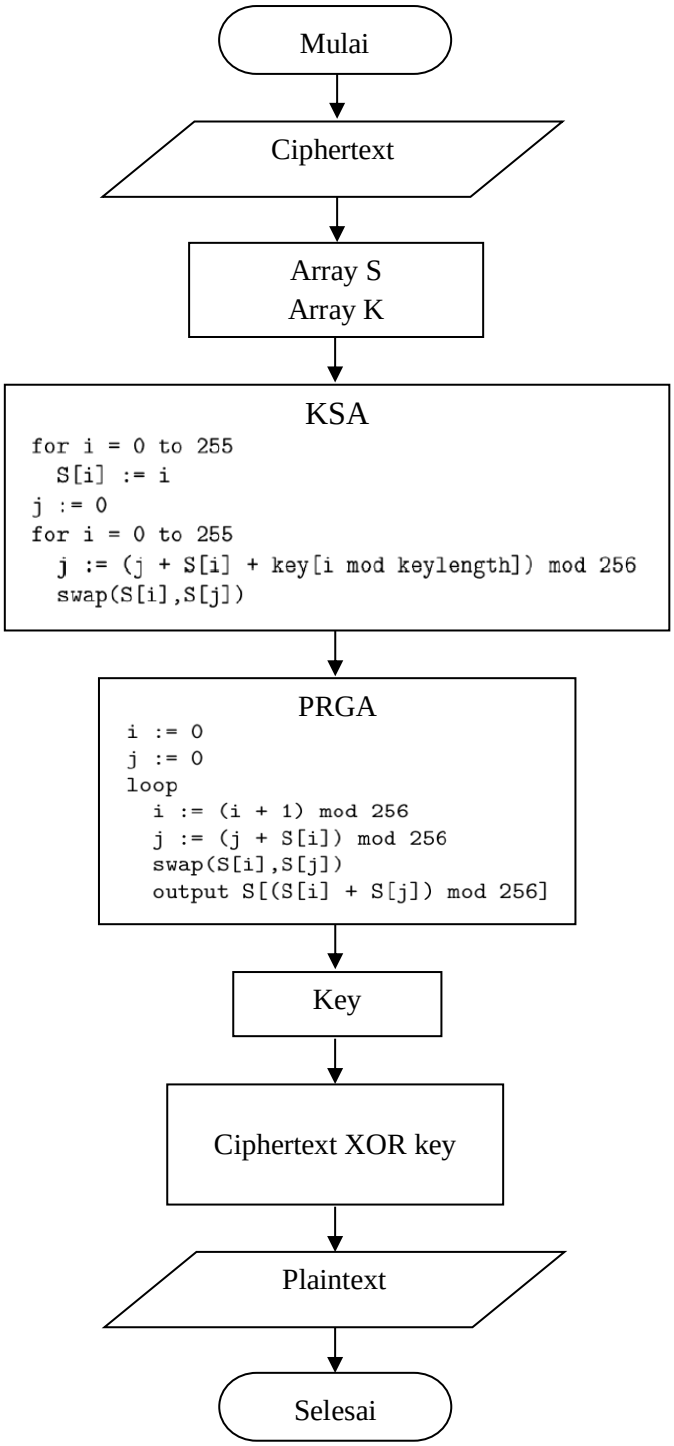
III.2.1.Flowchart Algoritma RC4

a. Enkripsi Algoritma RC4



Gambar III.1. *Flowchart* Enkripsi Algoritma RC4

b. Dekripsi Algoritma RC4



Gambar III.2. *Flowchart* Dekripsi Algoritma RC4

III.2.2.Studi Kasus Algoritma RC4

Tabel III.1. Data Nasabah

No. Akun	Nama Nasabah	Alamat	Jumlah Pinjaman
90361000130	Ayu Lestari	Jln. Pintu Air IV Gg Buntu No. 3	Rp. 2.000.000
90361000131	Wina Herwina	Jln. Pintu Air IV Gg Buntu No. 5	Rp. 2.000.000
90361000132	Yuni Siregar	Jln. Pintu Air IV Gg Buntu No. 6	Rp. 2.000.000
90361000133	Asri Larasati	Jln. Pintu Air IV Gg Buntu No. 8	Rp. 2.000.000
90361000134	Wati Puspita	Jln. Pintu Air IV Gg Buntu No. 11	Rp. 2.000.000
90361000135	Ayu Anggraini	Jln. Pintu Air IV Gg Buntu No. 14	Rp. 2.000.000
90361000136	Rini Arianti	Jln. Pintu Air IV Gg Buntu No. 15	Rp. 2.000.000
90361000137	Sumartini	Jln. Pintu Air IV Gg Buntu No. 19	Rp. 2.000.000
90361000138	Nurlela	Jln. Pintu Air IV Gg Buntu No. 21	Rp. 2.000.000
90361000139	Desi Ratna	Jln. Pintu Air IV Gg Buntu No. 22	Rp. 2.000.000
90361000140	Yati Suryani	Jln. Pintu Air IV Gg Buntu No. 23	Rp. 2.000.000

Berikut adalah implementasi algoritma RC4 dengan mengenkripsi mode 4 *byte* (untuk lebih menyederhanakan dalam perhitungan manual).

S-*Box* dengan panjang 4 *byte*, dengan S[0]=0, S[1]=1, S[2]=2 dan S[3]=3 sehingga *array* S menjadi : 0 1 2 3.

Inisialisasi 4 *byte* kunci *array*, K. Misalkan kunci ulang kunci sampai memenuhi seluruh adalah 3 2 1 4 , sehingga array K berisi 3 2 1 4 dan mencoba untuk mengenkripsikan nomor akun (0 1 3 9).

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA agar tercipta *state-array* yang acak. Penjelasan iterasi lebih lanjut dapat dijelaskan sebagai berikut:

Iterasi 1
i = 0
j = (0 + S[0] + K [0]) mod 4
  = (0 + 0 + 3) mod 4 = 3
Swap (S[0],S[3])
Hasil Array S
3 1 2 0

Iterasi 2
i = 1
j = (3 + S[1] + K [1]) mod 4
  = (3 + 1 + 2) mod 4 = 2
Swap (S[1],S[2])
Hasil Array S
3 2 1 0

Iterasi 3
i = 2
j = (2 + S[2] + K [2]) mod 4
  = (2 + 1 + 1) mod 4 = 0
Swap (S[2],S[0])
Hasil Array S
1 2 3 0

Iterasi 4
i = 3
j = (0 + S[3] + K [3]) mod 4
  = (0 + 0 + 4) mod 4 = 0
Swap (S[3],S[0])
Hasil Array S
0 2 3 1

Setelah melakukan KSA, akan dilakukan PRGA. PRGA akan dilakukan sebanyak 4 kali dikarenakan *plaintext* yang akan dienkripsi berjumlah 4 karakter. Hal ini disebakan karena dibutuhkan 1 kunci dan 1 kali pengoperasian XOR untuk tiap-tiap karakter pada *plaintext*. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

Array S : 0 2 3 1
Inisialisasi : i = 0, j = 0

Iterasi 1
i = (0 + 1) mod 4 = 1
j = (0 + S[1]) mod 4
  = (0 + 2) mod 4 = 2
swap (S[1],S[2])
0 3 2 1
K1 = S[(S[1]+S[2]) mod 4] = S[5 mod 4] = 1
K1 = 00000001

Iterasi 2
i = (1 + 1) mod 4 = 2
j = (2 + S[2]) mod 4
  = (2 + 2) mod 4 = 0
swap (S[2],S[0])
2 3 0 1
K2 = S[(S[2]+S[0]) mod 4] = S[2 mod 4] = 2
K2 = 00000010

Iterasi 3
i = (2 + 1) mod 4 = 3
j = (0 + S[3]) mod 4
  = (0 + 1) mod 4 = 1
swap (S[3],S[1])
2 1 0 3
K3 = S[(S[3]+S[1]) mod 4] = S[4 mod 4] = 0
K3 = 00000000

Iterasi 4
i = (3 + 1) mod 4 = 0
j = (1 + S[0]) mod 4
  = (1 + 2) mod 4 = 3
swap (S[0],S[3])
3 1 0 2
K4 = S[(S[0]+S[3]) mod 4] = S[5 mod 4] = 1
K4 = 00000001

Setelah menemukan kunci untuk tiap karakter, maka dilakukan operasi XOR antara karakter pada *plaintext* dengan kunci yang dihasilkan. Berikut adalah tabel ASCII untuk tiap-tiap karakter pada *plaintext* yang digunakan.

Kode ASCII	(*Binary* 8 bit)
0	00000000
1	00000001
3	00000011
9	00001001

Berikut adalah proses pengXORan dari *plaintext* dengan *key* yang telah didapat :

0 1 3 9 : 00000000 00000001 00000011 00001001

Key : 00000001 00000010 00000000 00000001

Ciphertext : 00000001 00000011 00000011 00001000

Proses dekripsi *ciphertext* menggunakan algoritma RC4 ini sama untuk proses *key-schedule*-nya. Untuk mendapatkan *plaintext*, *ciphertext* yang diperoleh di XORkan dengan *pseudo random byte* yang didapat sebelumnya. Maka hasilnya adalah *plaintext* atau teks asli.

Pesan dikirim dalam bentuk *ciphertext* sehingga setelah sampai di penerima pesan dapat kembali diubah menjadi *plaintext* dengan meng-XOR-kan dengan kunci yang sama. Pemrosesan pesan setelah sampai pada penerima dapat dilihat pada dibawah ini.

Proses XOR *pseudo random byte* dengan *ciphertext* pada dekripsi yaitu:

Ciphertext	: 00000001	00000011	00000011	00001000
pseudo random byte	: 00000001	00000010	00000000	00000001
Plaintext	: 00000000	00000001	00000011	00001001
	0	1	3	9

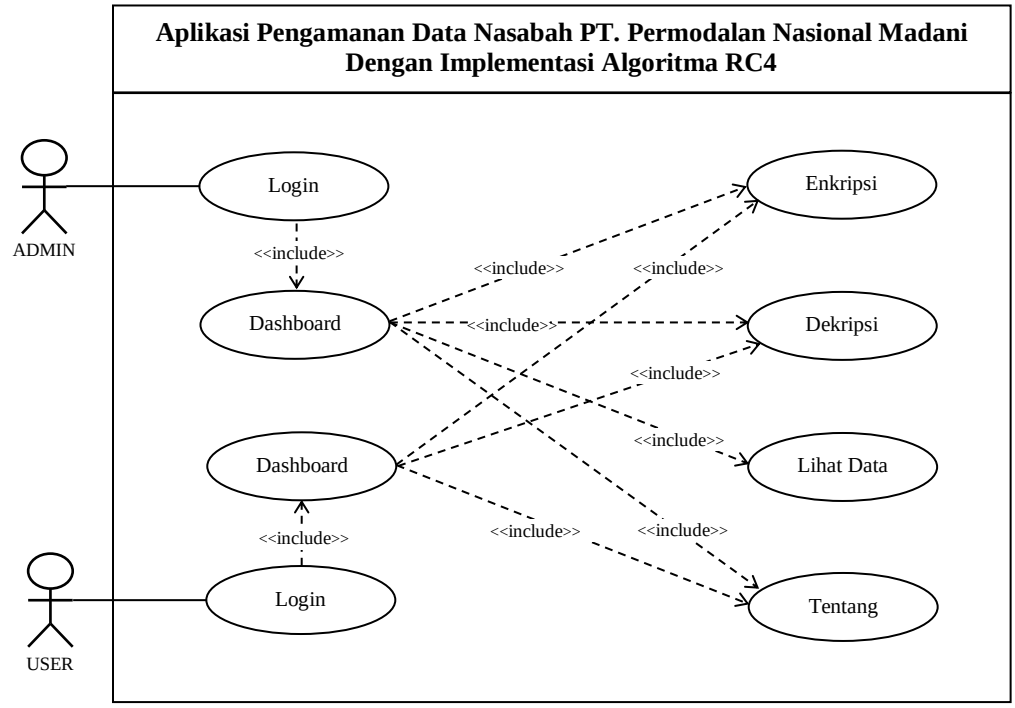
III.3. Desain Sistem

III.3.1. Desain Sistem Pemodelan UML

Desain sistem yang akan dibuat menggunakan beberapa bentuk diagram dari *Unified Modeling Language* (UML) yaitu *Use Case Diagram*, *Acitvity Diagram* dan *Sequence Diagram*.

III.3.1.1. *Use Case Diagram*

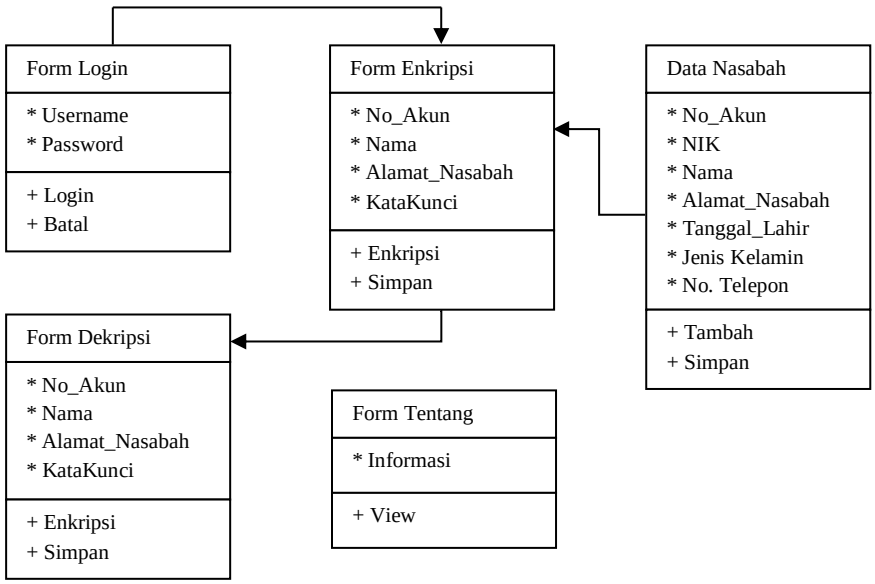
Use case diagram aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4, dapat dilihat pada gambar III.3 sebagai berikut :



Gambar III.3. *Use Case Diagram* Perancangan Aplikasi

III.3.1.2. *Class Diagram*

Class diagram aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.4 sebagai berikut :

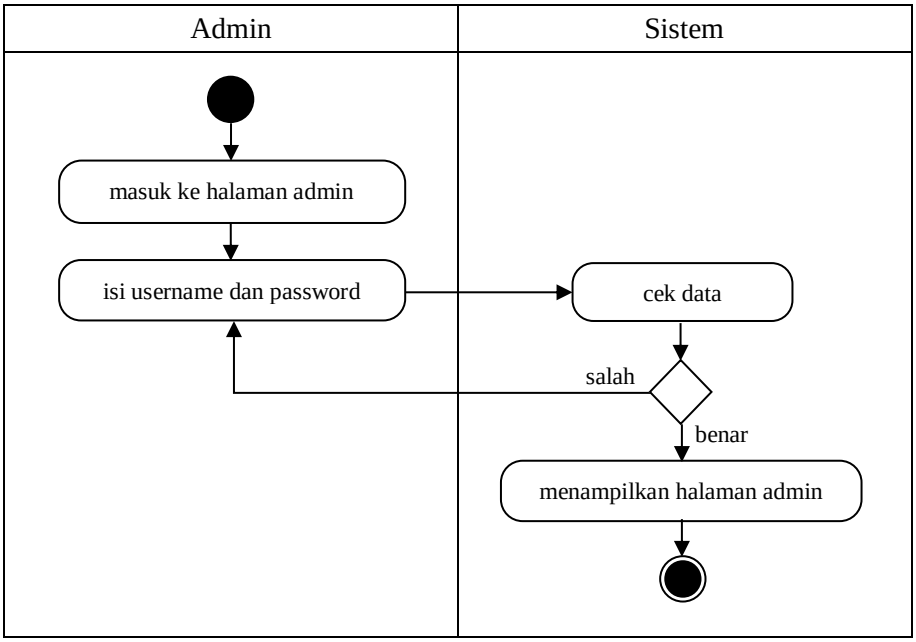


Gambar III.4. *Class Diagram* Perancangan Aplikasi

III.3.1.3. *Acitvity Diagram*

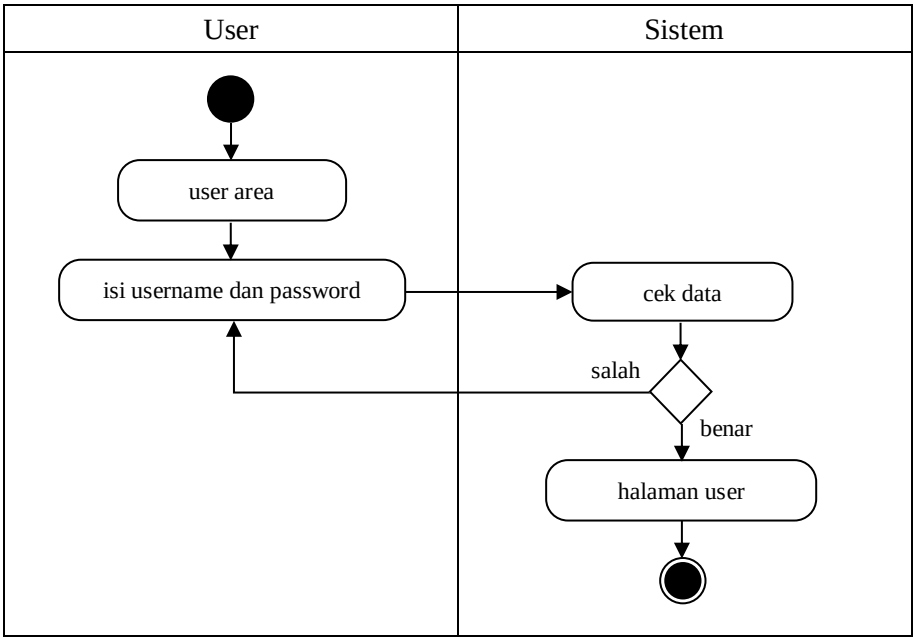
Activity diagram aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat sebagai berikut :

1. *Activity Diagram Login* Admin



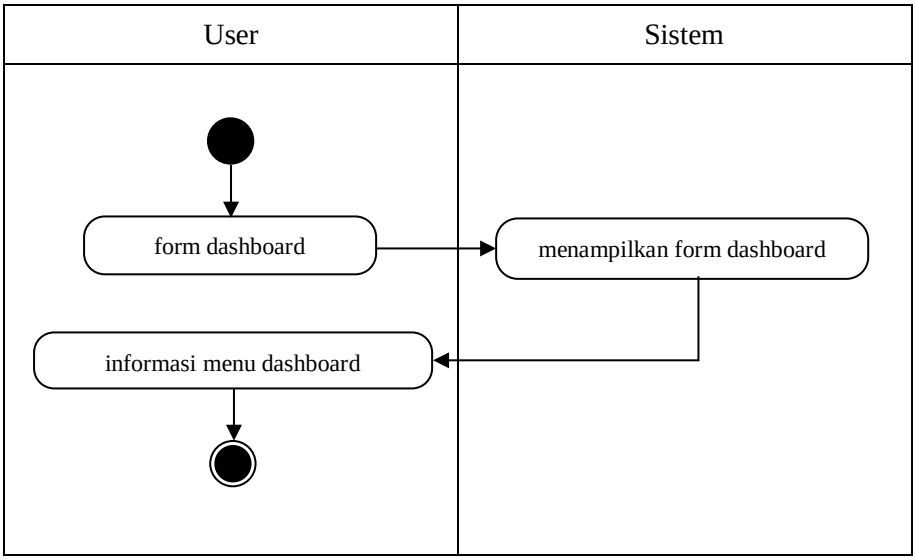
Gambar III.5. *Activity Diagram Login* Admin

2. *Activity Diagram Login User*



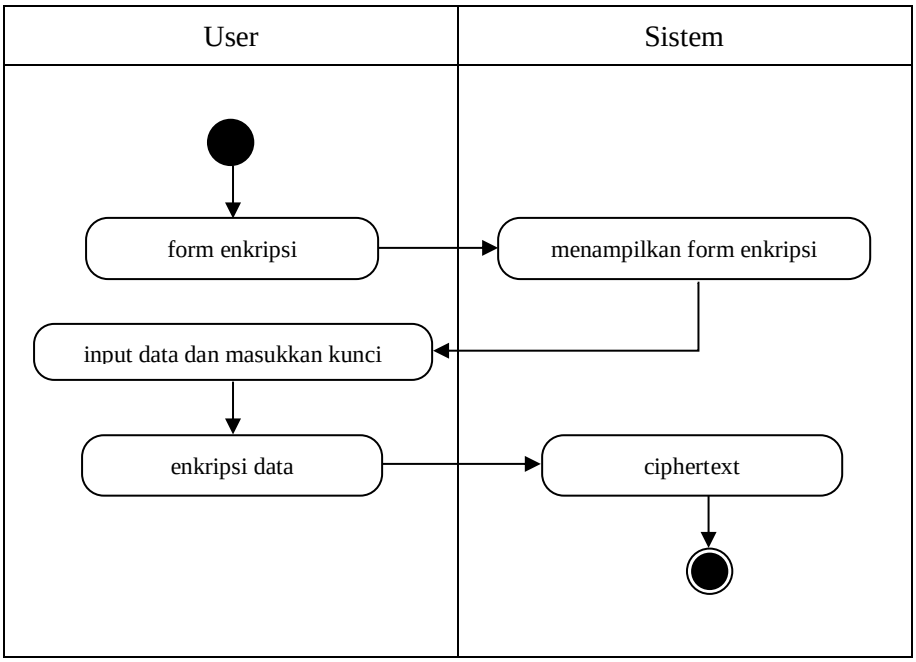
Gambar III.6. *Activity Diagram Login User*

3. *Activity Diagram Form Dashboard*



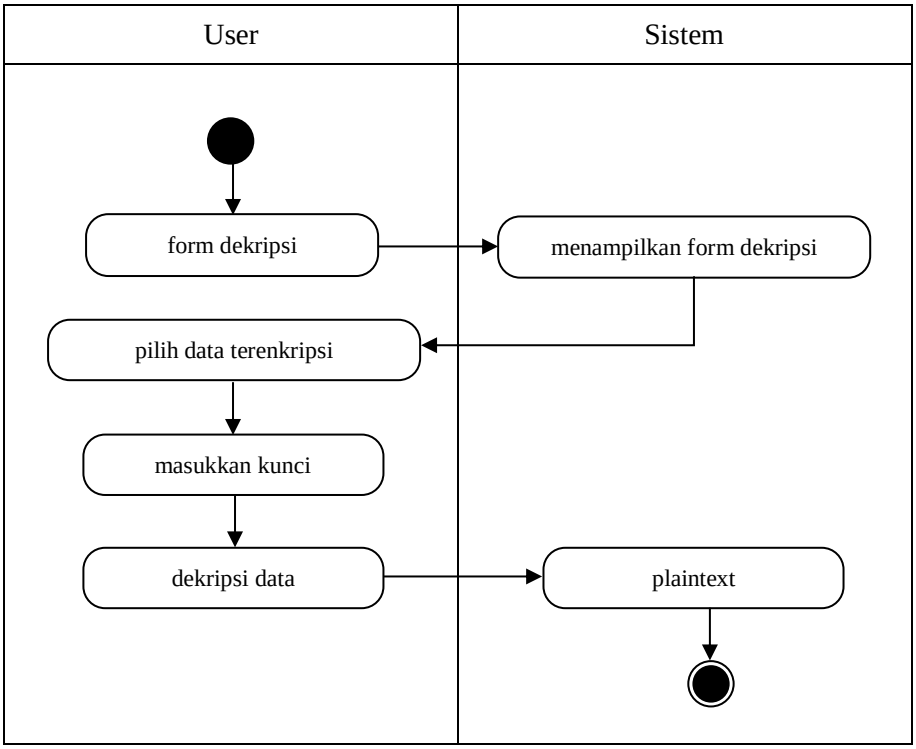
Gambar III.7. *Activity Diagram Form Dashboard*

4. *Activity Diagram Form* Enkripsi



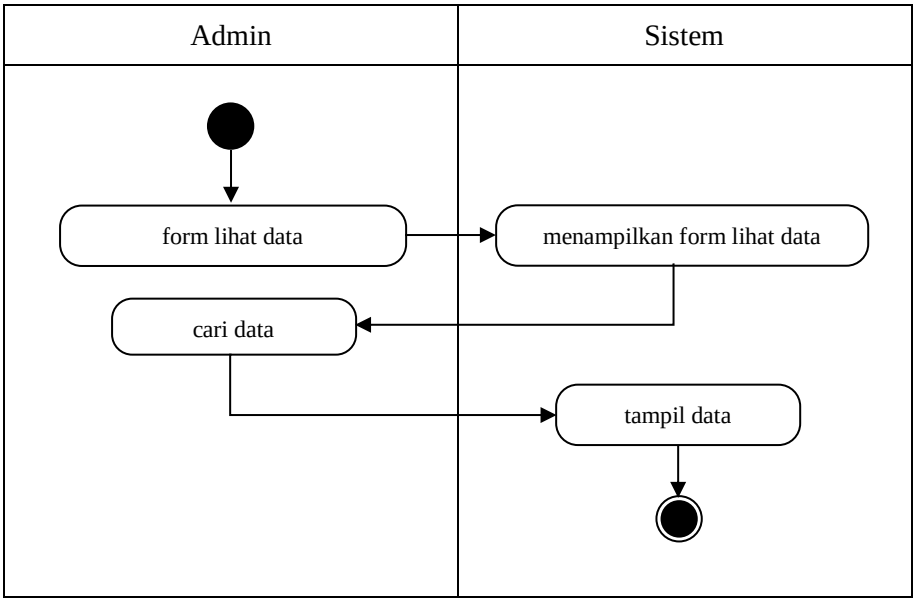
Gambar III.8. *Activity Diagram Form* Enkripsi

5. *Activity Diagram Form* Dekripsi



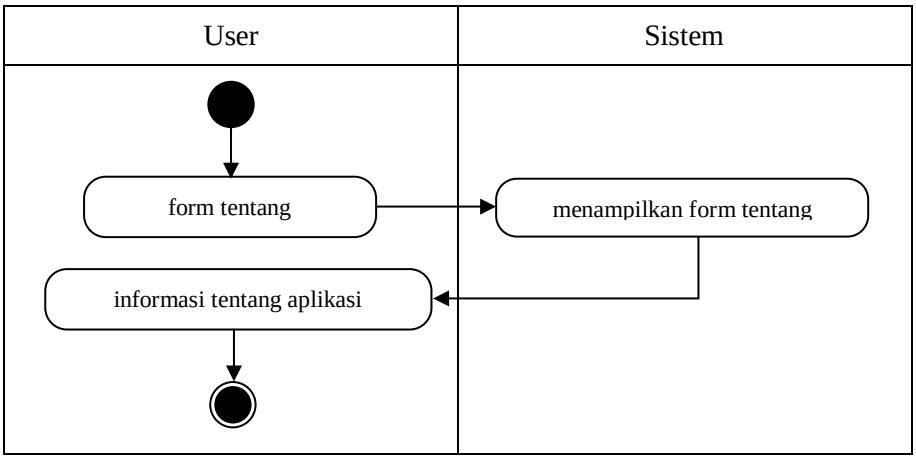
Gambar III.9. *Activity Diagram Form* Dekripsi

6. *Activity Diagram Form* Lihat Data



Gambar III.10. *Activity Diagram Form* Lihat Data

7. *Activity Diagram Form* Tentang

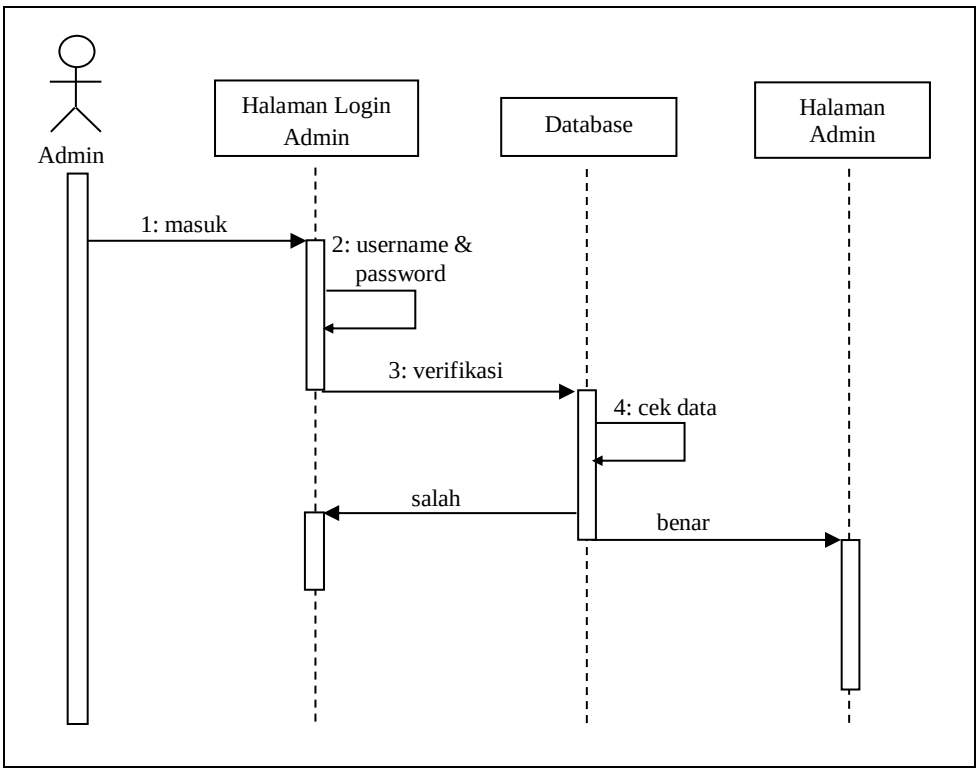


Gambar III.11. *Activity Diagram Form* Tentang

III.3.1.4. *Sequence* Diagram

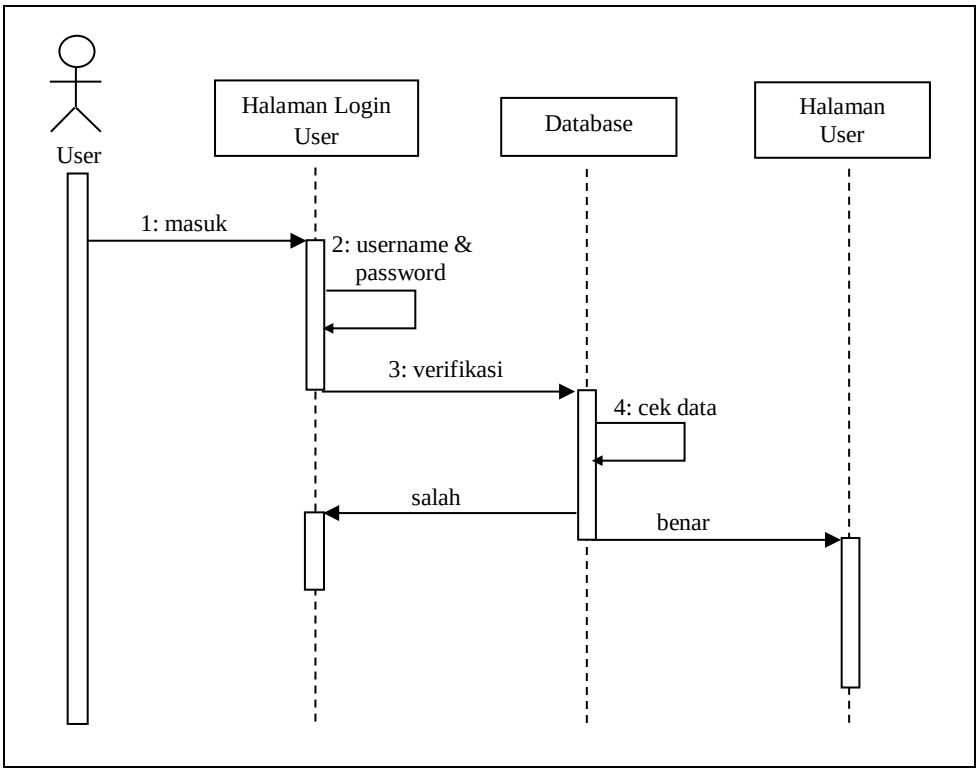
Sequence diagram aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 sebagai berikut :

1. *Sequence Diagram Login* Admin



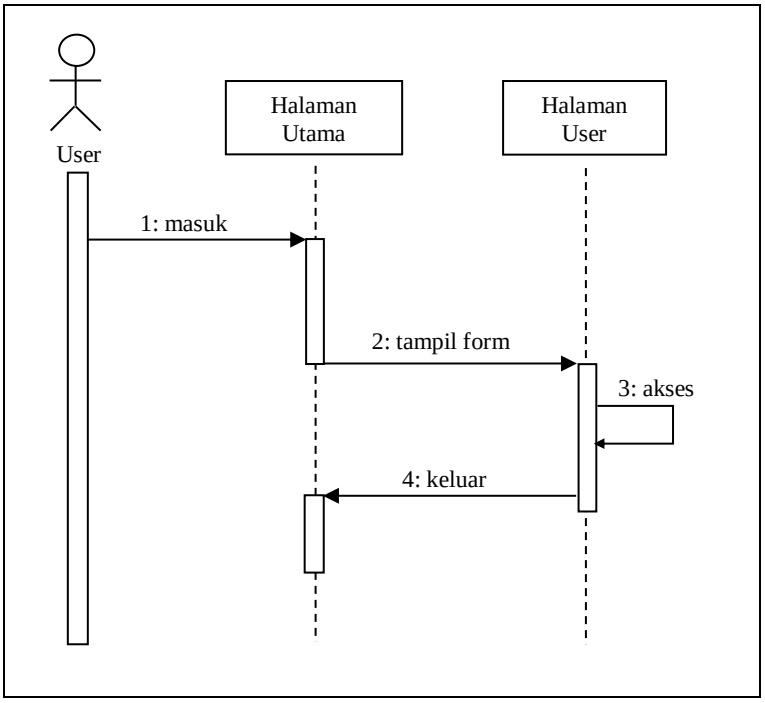
Gambar III.12. *Sequence Diagram Login* Admin

2. *Sequence Diagram Login User*



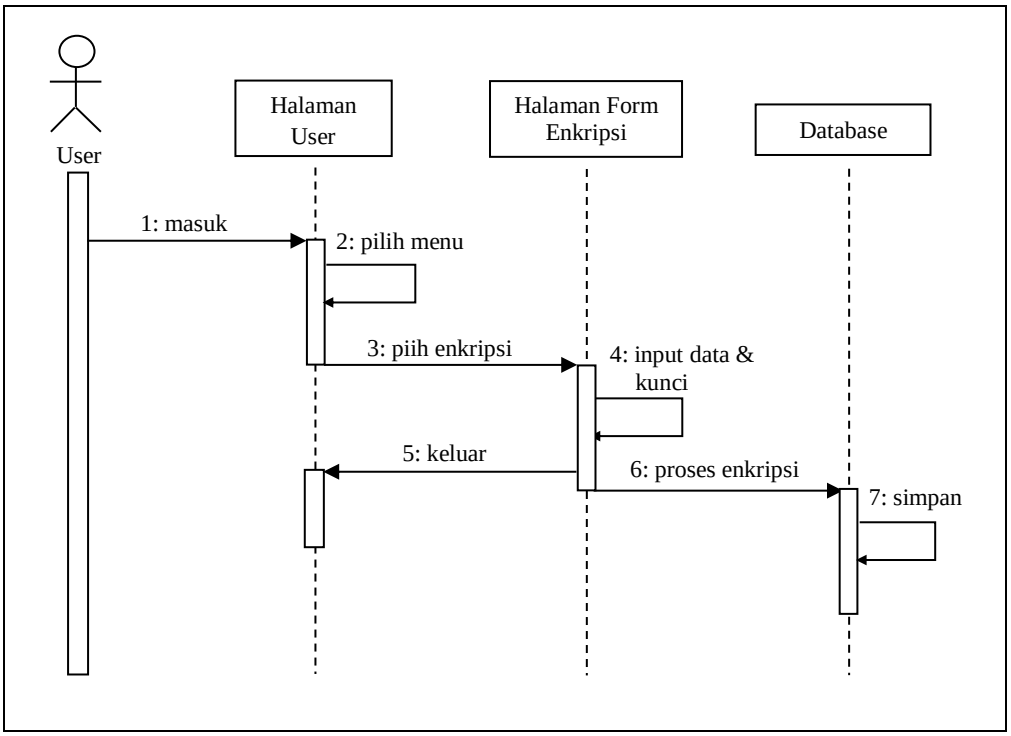
Gambar III.13. *Sequence Diagram Login User*

3. *Sequence Diagram Form Dashboard*



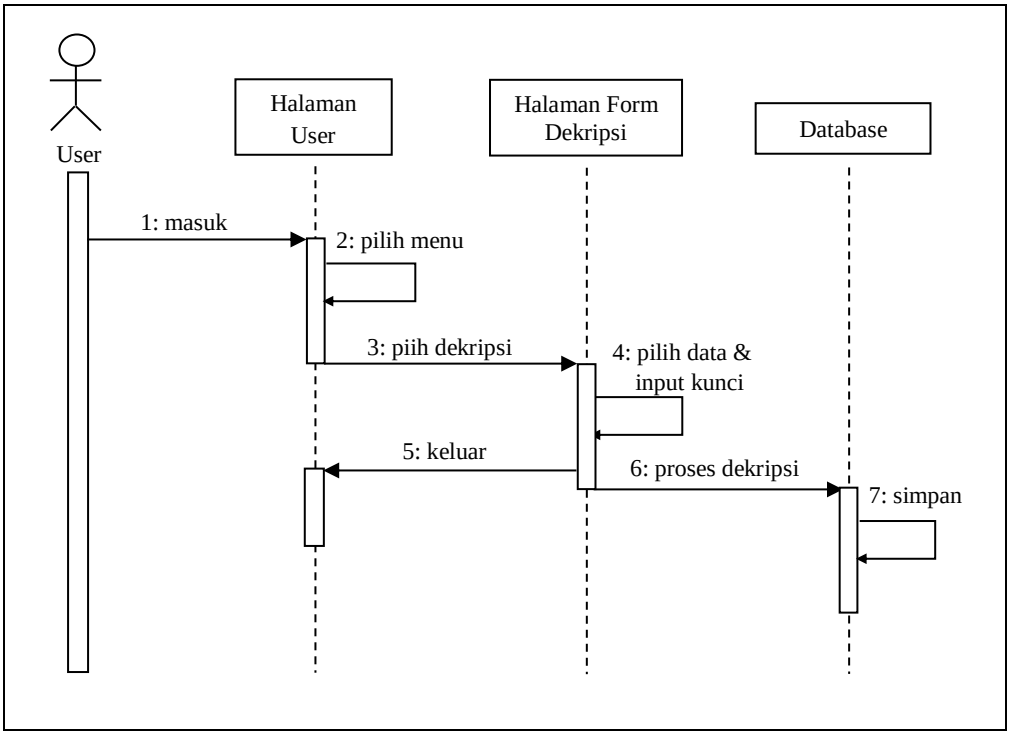
Gambar III.14. *Sequence Diagram Form Dashboard*

4. *Sequence Diagram Form* Enkripsi



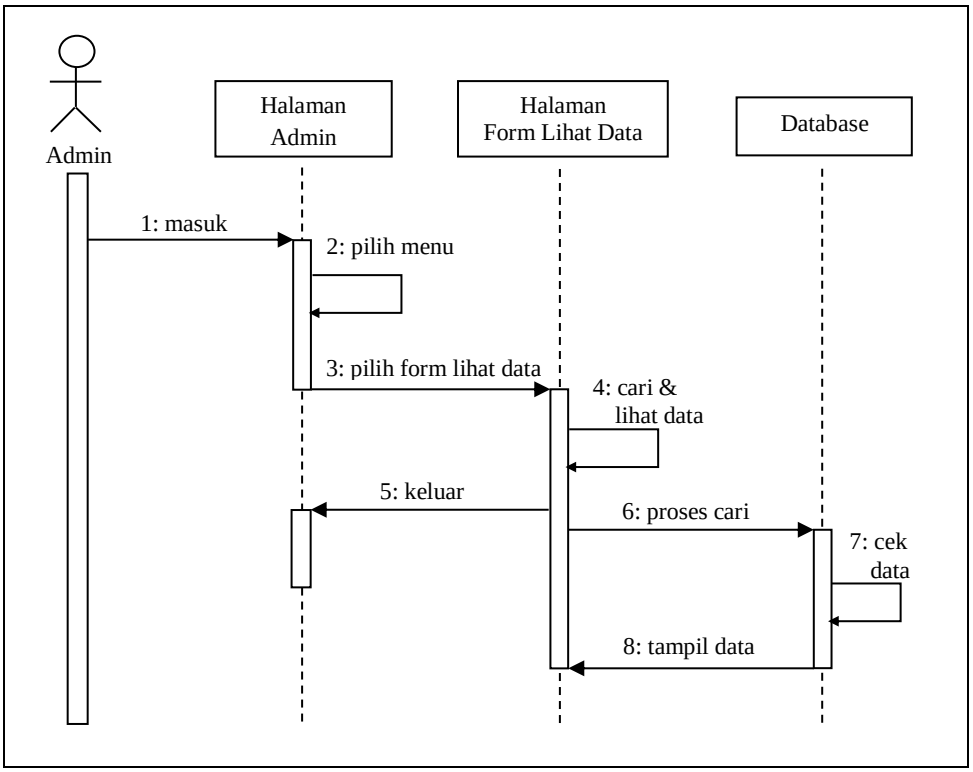
Gambar III.15. *Sequence Diagram Form* Enkripsi

5. *Sequence Diagram Form* Dekripsi



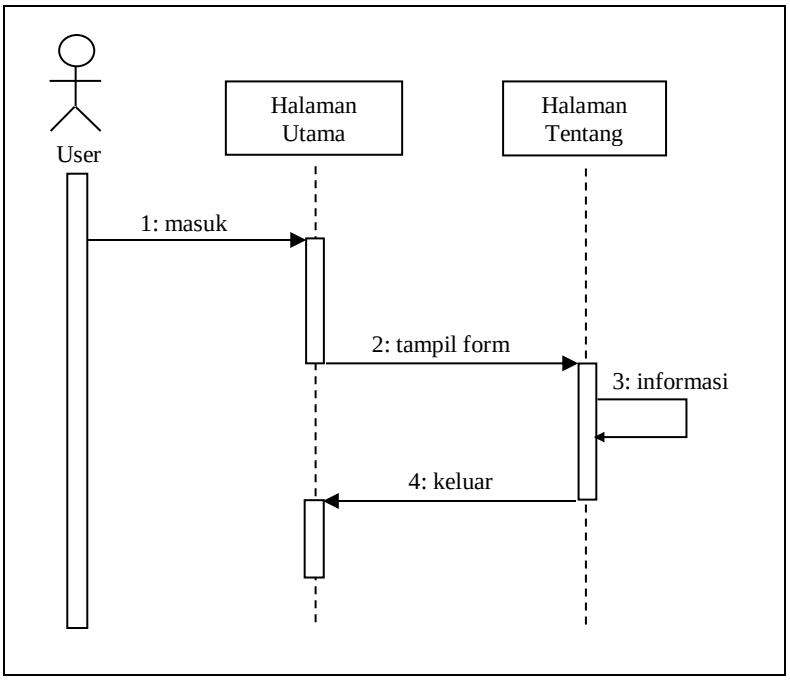
Gambar III.16. *Sequence Diagram Form* Dekripsi

6. *Sequence Diagram Form* Lihat Data



Gambar III.17. *Sequence Diagram Form* Lihat Data

7. *Sequence Diagram Form* Tentang



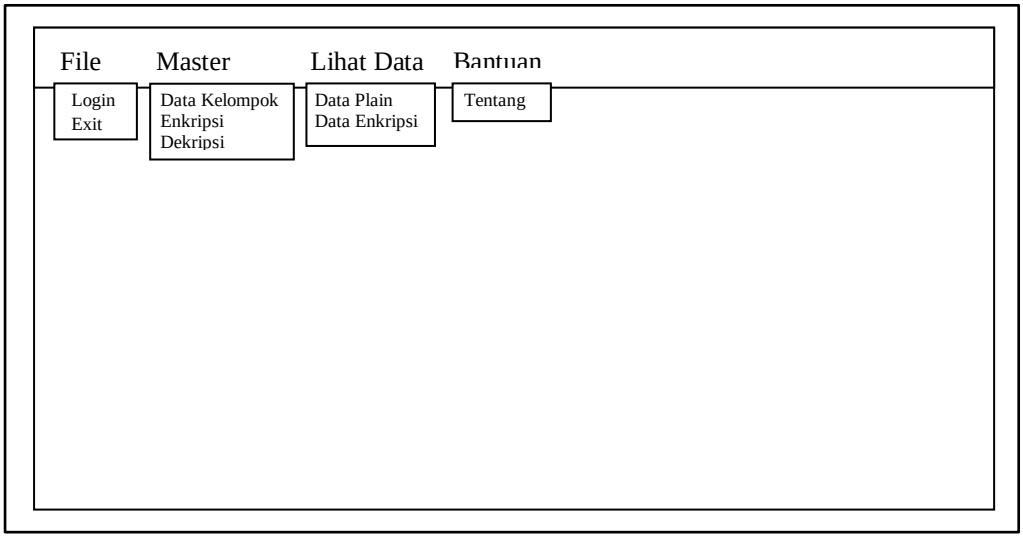
Gambar III.18. *Sequence Diagram Form* Tentang

III.3.2. Desain Sistem Aplikasi

Desain sistem aplikasi yang telah dibuat digambarkan dan dijabarkan sebagai berikut :

1. Desain *Dashboard*

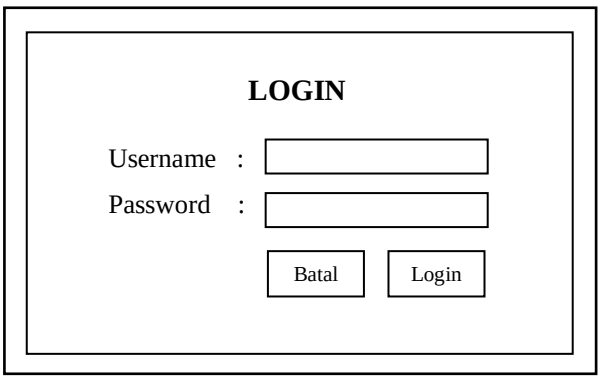
Desain *dashboard* aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.19 sebagai berikut:



Gambar III.19. Desain *Form Dashboard*

2. Desain *Form Login*

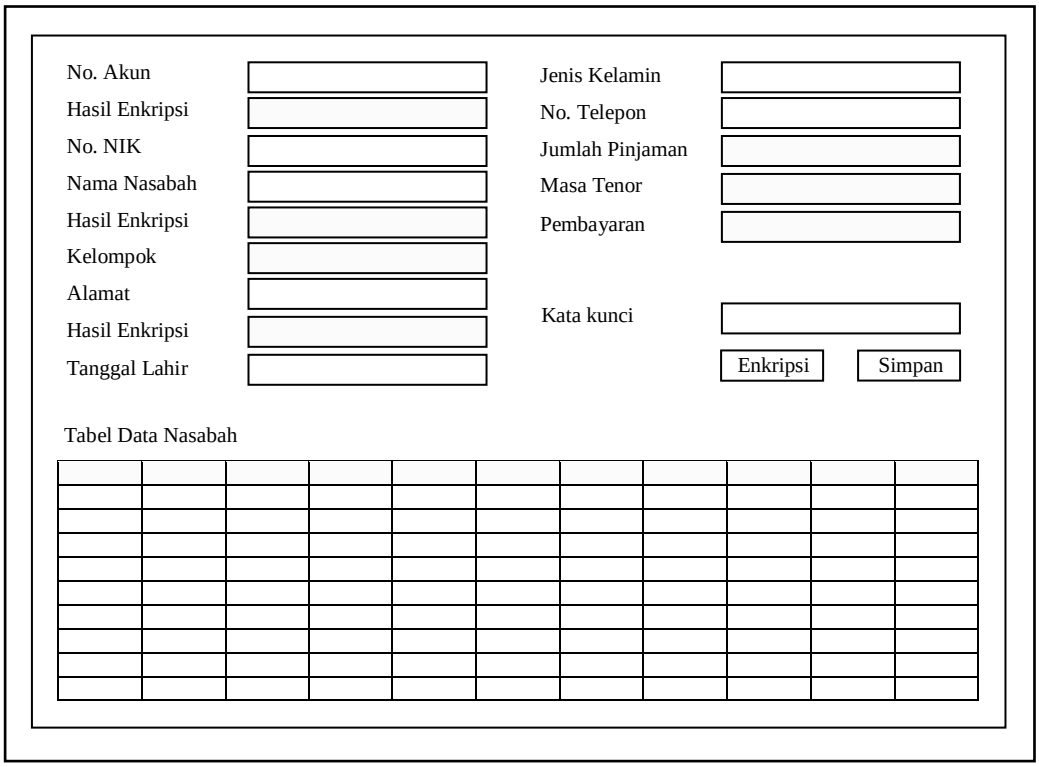
Desain *form login* aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.20 sebagai berikut:



Gambar III.20. Desain *Form Login*

3. Desain *Form Enkripsi*

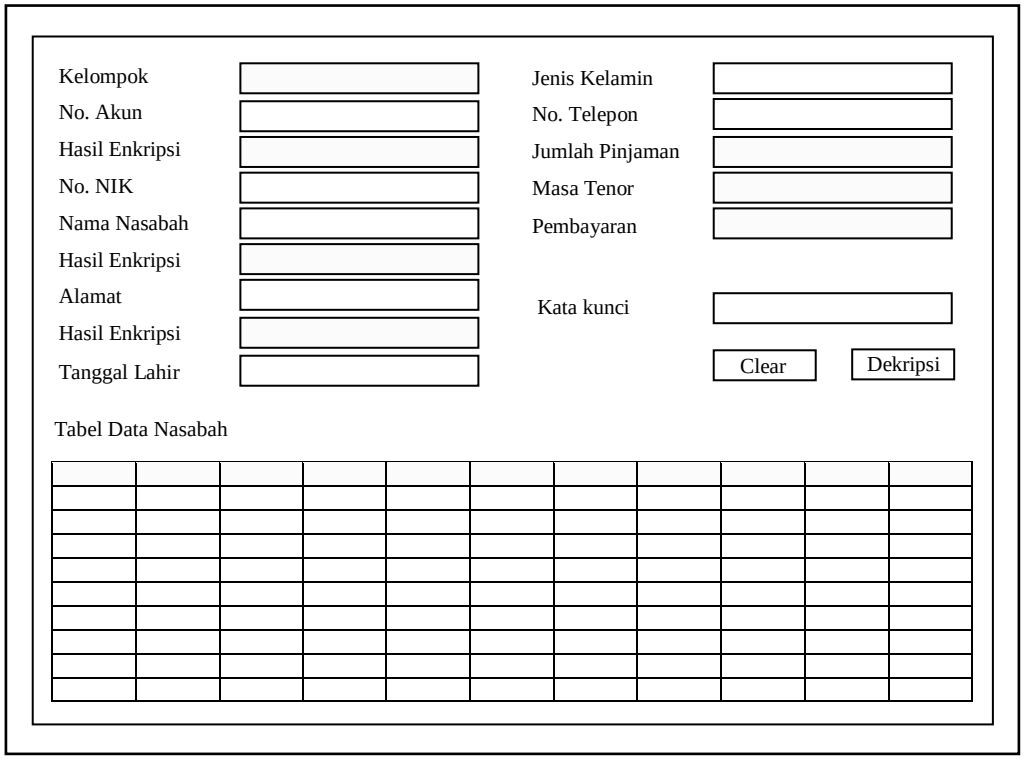
Desain *form enkripsi* aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.21 sebagai berikut:



Gambar III.21. Desain *Form* Enkripsi

4. Desain *Form* Dekripsi

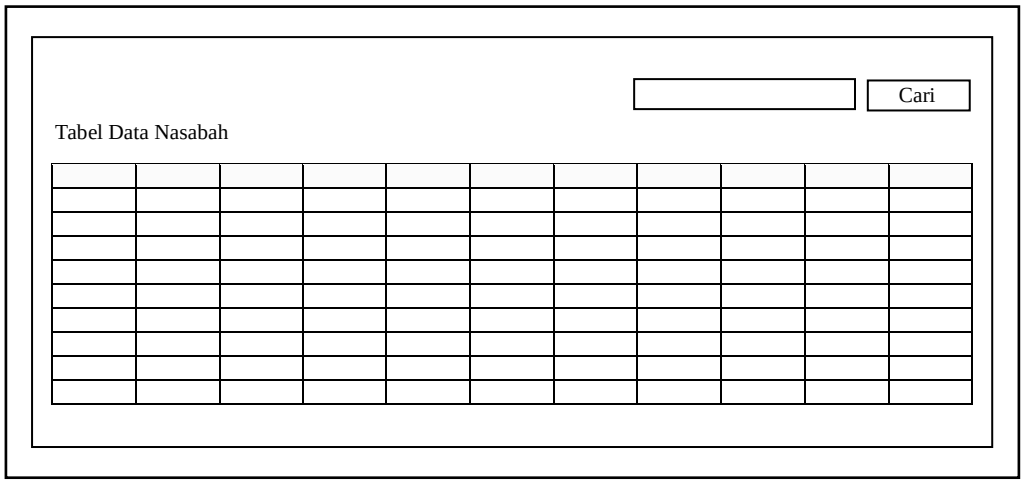
Desain *form dekripsi* aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.22 sebagai berikut:



Gambar III.22. Desain *Form* Dekripsi

5. Desain *Form* Lihat Data

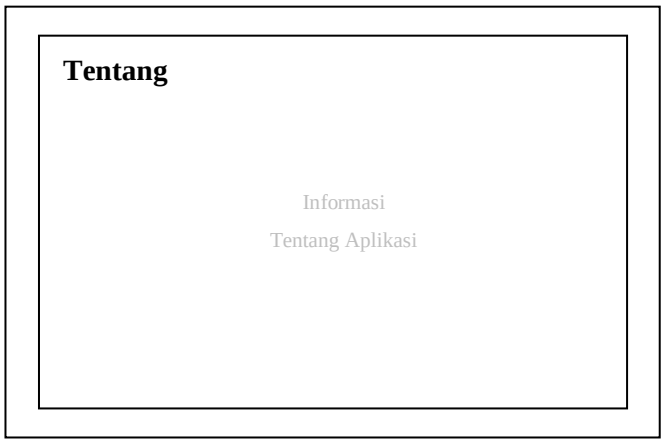
Desain *form* lihat data aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.23 sebagai berikut:



Gambar III.23. Desain *Form* Lihat Data

6. Desain *Form* Tentang

Desain *form* tentang aplikasi pengamanan data nasabah PT. PNM dengan implementasi algoritma RC4 dapat dilihat pada gambar III.24 sebagai berikut:



Gambar III.24. Desain *Form* Tentang