BAB III
ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai Aplikasi Keamanan Database Menggunakan Algoritma Vigenere dan Algoritma RC4 yang meliputi analisa sistem dan desain sistem.

III.1. Analisis Masalah

Adapun analisa masalah pada Aplikasi Keamanan Database Menggunakan *Algoritma Vigenere* dan *Algoritma RC4* yaitu :

1. Banyaknya pihak-pihak yang melakukan modifikasi data atau *database* yang bertujuan mengubah data yang ada sehingga merugikan pihak-pihak tertentu.
2. Terjadinya interupsi yang dapat mengganggu ketersediaan *database* yaitu data yang ada dapat dihapus sehingga pihak yang membutuhkan data atau informasi tersebut tidak dapat menemukan data atau informasi tersebut
3. Seringnya terjadi ancaman intersepsi yaitu merupakan ancaman terhadap kerahasiaan data atau informasi yang ada dalam *database*.

Berdasarkan analisa diatas maka penulis telah melakukan evaluasi dari sistem yang sedang berjalan dan penulis menemukan kelemahan sistem yang ada. Dengan demikian penulis memberikan suatu solusi yang diharapkan dapat mengatasi kelemahan sistem yang ada. Adapun solusi yang ditawarkan adalah membangun Aplikasi Keamanan *Database* Menggunakan *Algoritma Vigenere* dan *Algoritma RC4.* Aplikasi ini adalah salah satu alat yang diyakini mampu memberikan konstribusi positif dalam menjamin keamanan *database*.

III.2. Algoritma Vigenere

Vigenère Cipher dibuat oleh *Blaise de Vigenère* pada abad 16, yang merupakan metode menyandikan teks *alfabet* dengan menggunakan deretan sandi *Caesar* berdasar-kan deretan huruf-huruf pada kata kunci. *Algoritma Vigenère* dapat dinyatakan secara matematis, dengan menggunakan penjumlahan dan operasi modulus:

Algoritma enkripsi dinyatakan:

$$C_i = (P_i + K_i) \bmod 26$$

Algoritma dekripsi dinyatakan:

$$P_i = (C_i - K_i) \bmod 26$$

C adalah nilai desimal karakter *cipherteks* ke-i, P adalah nilai desimal karakter plainteks ke-i, dan K adalah nilai desimal karakter kunci ke-i, dengan asumsi angka desimal karakter A = 0, B = 1, ..., Z = 25. Jika hasil dekripsi bernilai negatif, maka nilai tersebut ditambah dengan angka 26 untuk mendapatkan plainteks.

III.2.1 Sistem Sandi RC4

Sistem sandi RC4 dikembangkan oleh Ronald Rivest pada tahun 1984 merupakan sistem sandi stream yang paling banyak digunakan minsalnya pada protokol SSL/TLS. RC4 merupakan sistem sandi stream berorientasi *byte*. Masukkan algoritma enkripsi RC4 merupaka sebuah *byte,* kemudian dilakukan operasi XOR dengan sebuah *byte* kunci, dan menghasilkan sebuah *byte* sandi. ‘

2. Penjadwalan Kunci RC4

Sistem sandi RC4 menggunakan *state*, yaitu larik *byte* berukuran 256 yang terpermutasi, dan tercampur oleh kunci. Kunci *enkripsi* juga merupakan larik *byte* berukuran 256. Sebelum

melakukan *enkripsi*, dan *dekripsi*, sistem sandi RC4 melakukan inisialisasi terhadap *state* dengan Algoritma, algoritma ini dosebut dengan penjadwalan kunci *(key scheduling).*

Input: kunci Output: {S[1],...,S[N]}

For i=0 255 do S[i]=i End for J=0 For i=0 255 do

J=(j+S[i]+Kunci[imod [kumci]]) mod 256 swap(S[i],S[j]) end

3. Enkripsi RC4

Setelah *state* sandi RC4 merupakan *state*, yaitu larik *byte* pada teks asli dikenakan operasi XOR dengan kunci *byte* untuk menghasilkan *byte* pada teks sandi, Kunci *byte* yang digunakan pada *enkripsi* dibangitkan denagn memanfaatkan *state* S. Algoritma *enkripsi* RC4: *Input*: P {Stream teks asli} Output: C{Stream teks sandi} *i*=0, j=0 {bisa diisi nilai lain} *While* P masih memiliki *byte do i*=(i+1) mod 256 j=(j+S[i]) mod 256 swap (S[i], S[j]) k= S[S[i]+S[j]] mod 256 C=Pok End while

4. Dekripsi RC4

Algoritma *dekripsi* sistem sandi RC4 serupa dengan algoritma *enkripsi* sistem RC4.

5. Implementasi Sistem Sandi RC4

6. Ukuran kunci sandi RC4 sangat berpengaruh terhadap keamanan sistem sandi RC4. Rc4 telah dibuktikan tidak aman untuk ukuran kunci yang kecil, yaitu dibawah 5 *byte*. Rekomendasi penggunaan sistem sandi RC4 agar memiiki keamanan yang kuat adalah:

a. Ukuran kunci sama atau lebih besar daripada 256 bit(16*byte*)
b. Setiap sesi baru membangkitkan kunci yang baru (dengan pembangkitan kunci yang baru menghindari penyerang untuk melakukan analisis sandi diferensial pada sistem sandi).

7. Implementasi sistem Sandi RC4

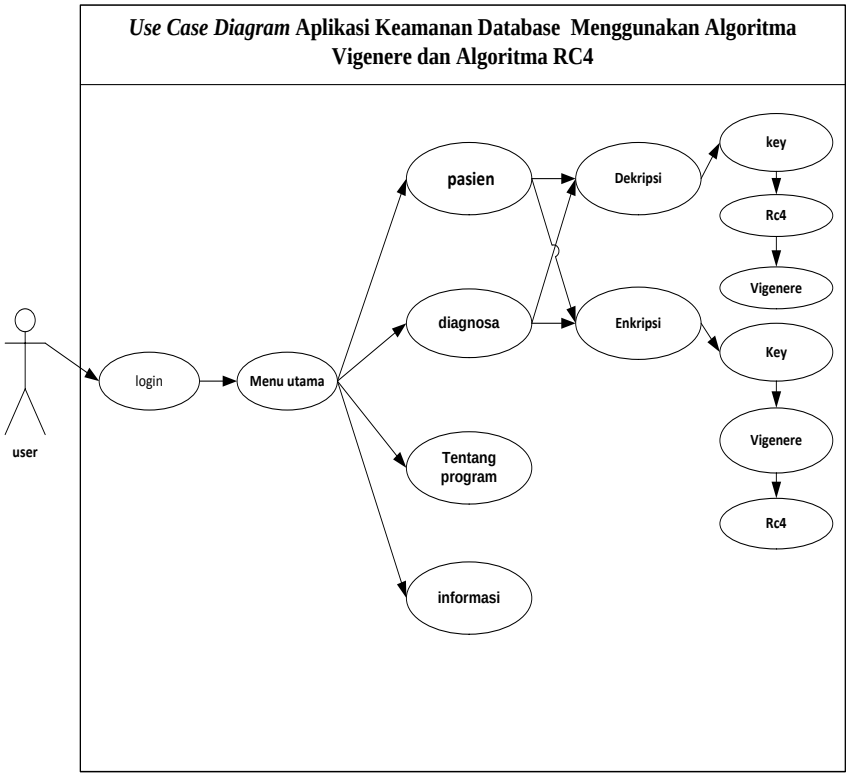
Sistem sandi RC4 mudah diimplementasikan dalam perangkat lunak karena beroperasi pada tipe data *byte* (Kurniadi:2015;6).

III.3. Desain Sistem Baru

Desain Sistem Baru menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram*.

III.3.1. *Usecase* Diagram

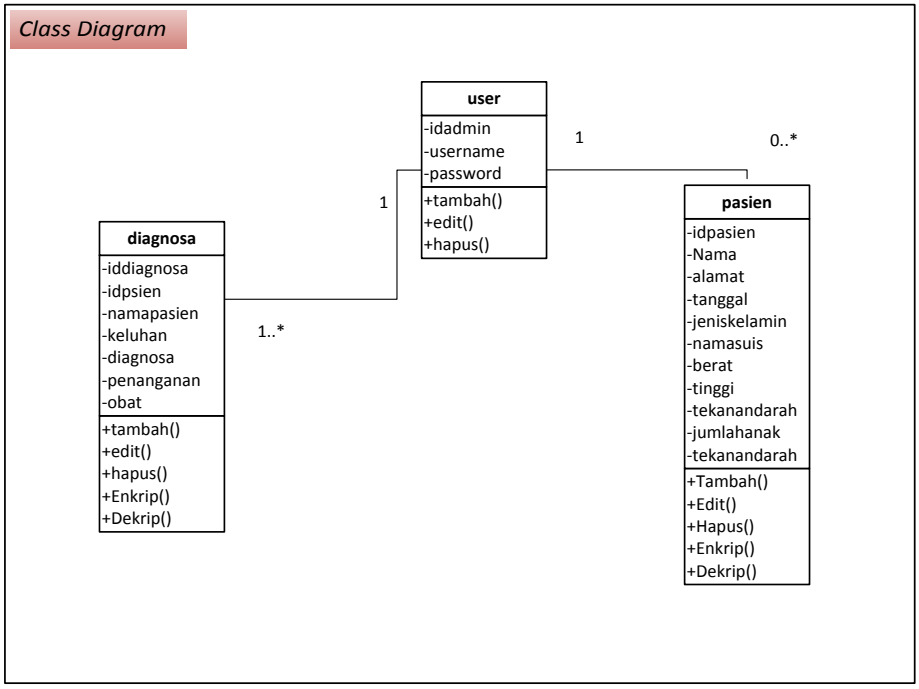
Secara garis besar, proses sistem yang akan dirancang digambarkan dengan *usecase diagram* yang terdapat pada Gambar III.1 :



Gambar III.1. *Use Case* Diagram Aplikasi Keamanan Database Menggunakan Algoritma Vigenere dan Algoritma RC4

III.3.2. *Class* Diagram

Rancangan kelas-kelas yang akan digunakan pada sistem yang akan dirancang dapat dilihat pada gambar III.2 :



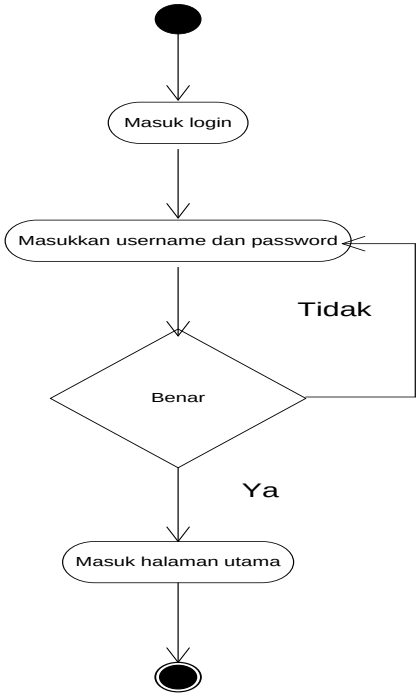
Gambar III.2. *Class Diagram* Aplikasi Keamanan Database Menggunakan Algoritma Vigenere dan Algoritma RC4

III.3.3. *Activity Diagram*

Diagram aktivitas menggambarkan suatu urutan proses yang terjadi pada sistem dari dimulainya aktivitas hingga aktivitas berhenti. Diagram aktivitas hampir mirip dengan diagram *flowchart*. Diagram aktivitas merupakan salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use-case*. Berikut *activity* diagram yang ditunjukkan pada gambar ini:

1. *Activity Diagram Login*

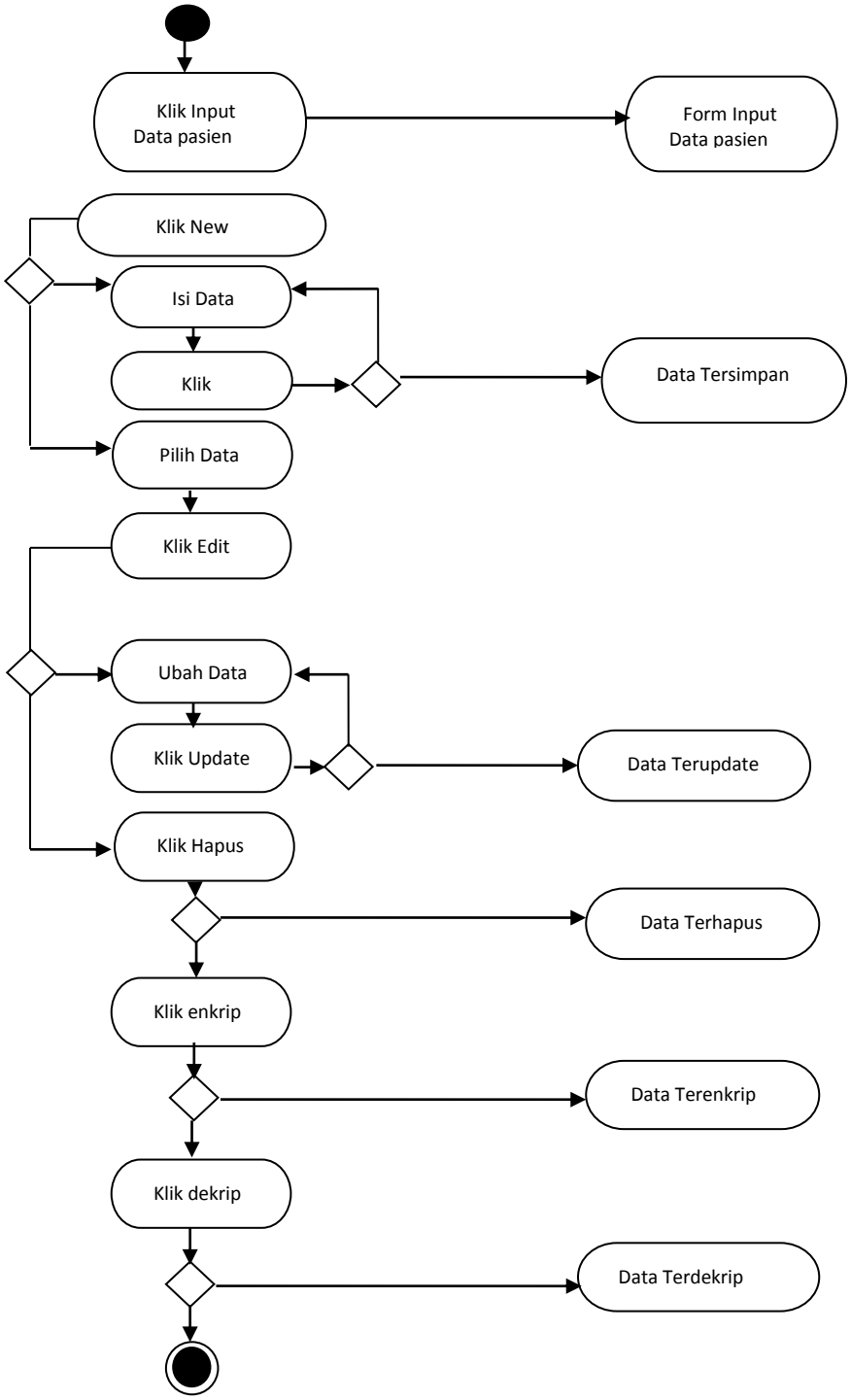
Activity diagram login merupakan *activity diagram* untuk proses *login user*. *Activity diagram login* ditunjukkan pada gambar berikut ini:



Gambar III.3. *Activity Diagram Login*

2. *Activity* Diagram data pasien

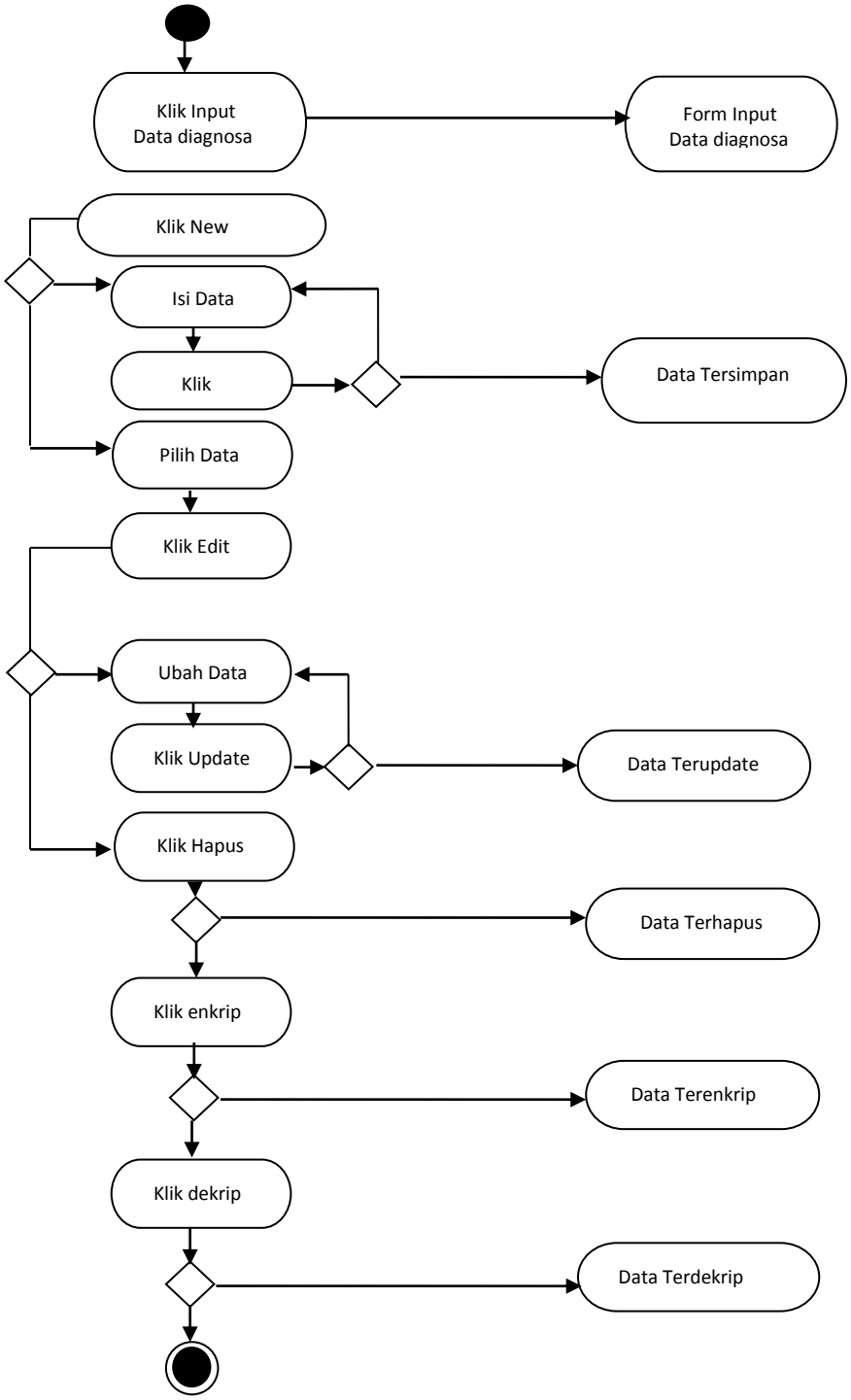
Activity diagram data pasien merupakan *activity diagram* untuk inputan data pasien . *Activity diagram* data pasien ditunjukkan pada gambar berikut ini:



Gambar III.4. *Activity Diagram* Data Pasien

3. *Activity* Diagram data diagnosa

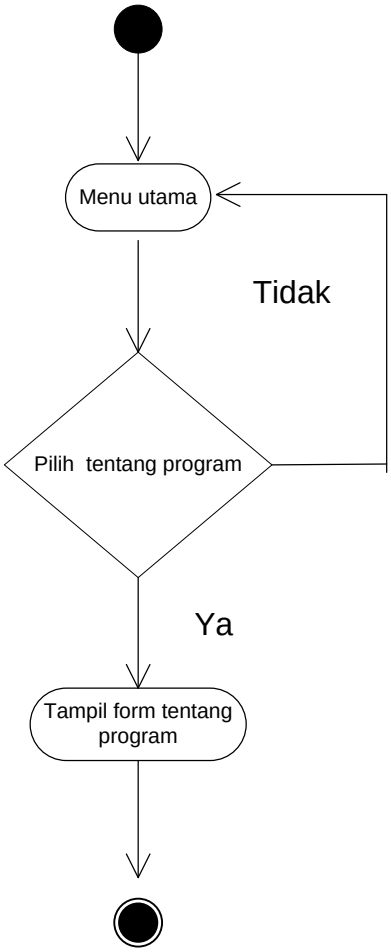
Activity diagram data diagnosa merupakan *activity diagram* untuk proses diagnosa . *Activity diagram* data diagnosa ditunjukkan pada gambar berikut ini:



Gambar III.5. *Activity Diagram* Data Pasien

4. *Activity* Diagram Tentang Program

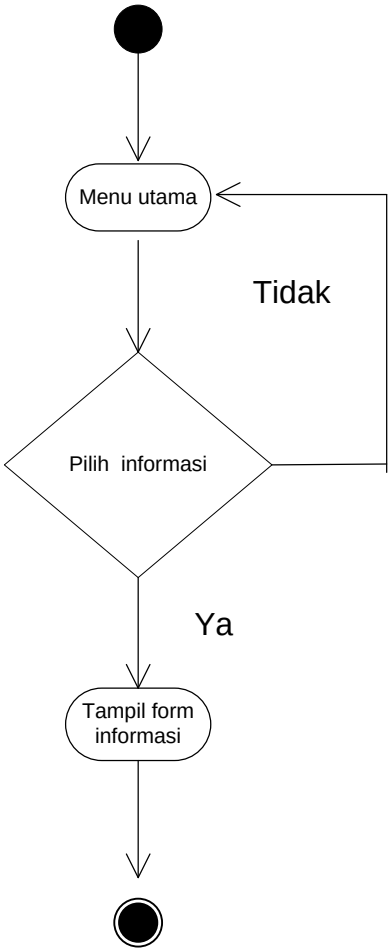
Activity diagram tentang program merupakan *activity diagram* untuk melihat *form* tentang Program. *Activity diagram* tentang program ditunjukkan pada gambar berikut ini:



Gambar III.6. *Activity Diagram* Tentang Program

5. *Activity* Diagram Informasi

Activity diagram informasi merupakan *activity diagram* untuk melihat *form* informasi cara menggunakan aplikasi yang tealah dibangun. *Activity diagram* informasi ditunjukkan pada gambar berikut ini:



Gambar III.7. *Activity Diagram* Informasi

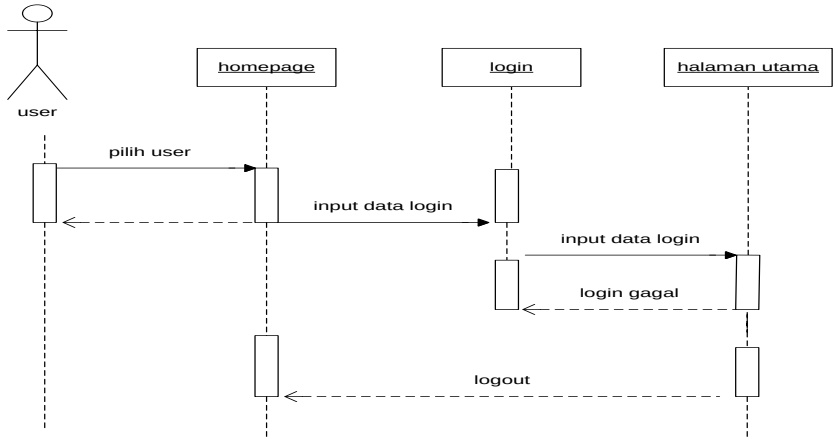
III.3.4. *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-

objek yang terkait). Serangkaian kegiatan saat terjadi *event* pada aplikasi ini dapat dilihat pada gambar dibawah:

1. *Sequence Diagram Login*

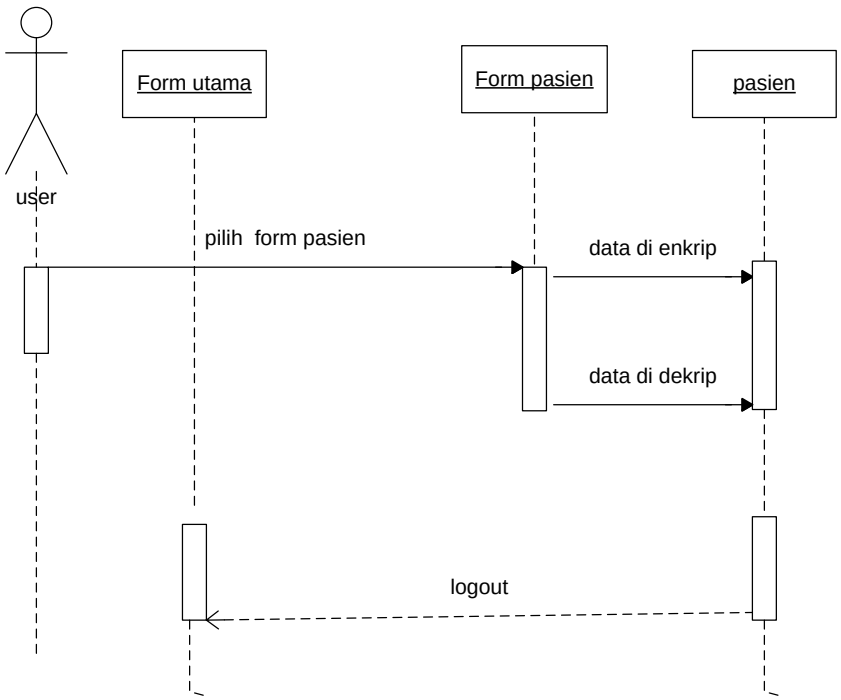
Proses *sequence login* adalah *user* memasukkan *username* dan *password* pada *form login*, dari *form login* data akan di kirim ke sistem untuk di cek kevalidan data. Jika data *valid* maka akan ditampilkan *form* utama *Sequence diagram Login* ditunjukan pada gambar berikut ini :



Gambar III.8. *Diagram Sequence* Login

2. *Sequence Diagram* Data Pasien

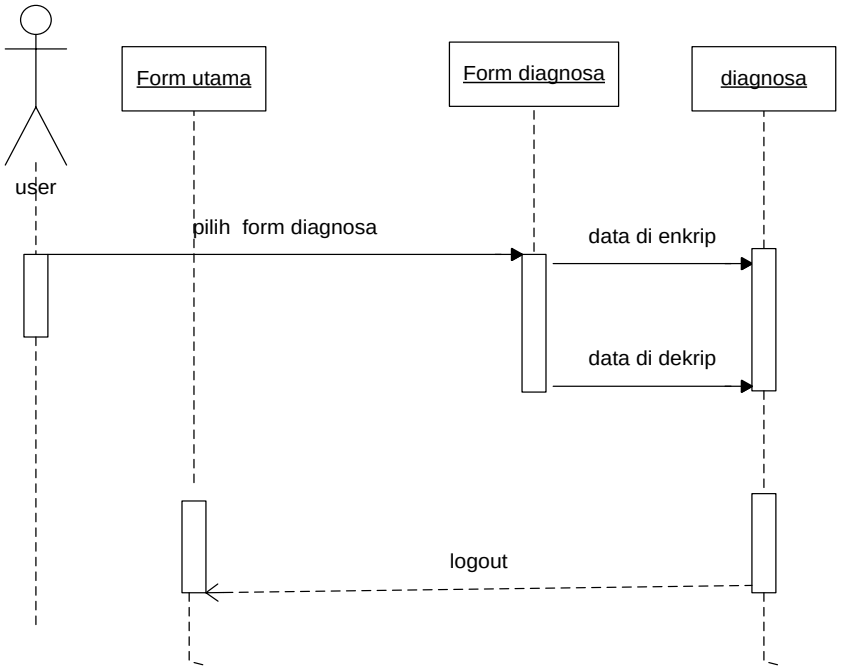
Sequence diagram data pasien menggambarkan interaksi antara objek pada proses data pasien dan mengenkrip, serta dekrip data pasien . *Sequence* diagram data pasien ditunjukkan pada gambar dibawah ini:



Gambar III.9. *Sequence Diagram* Data Pasien

3. *Sequence* Diagram Data Diagnosa

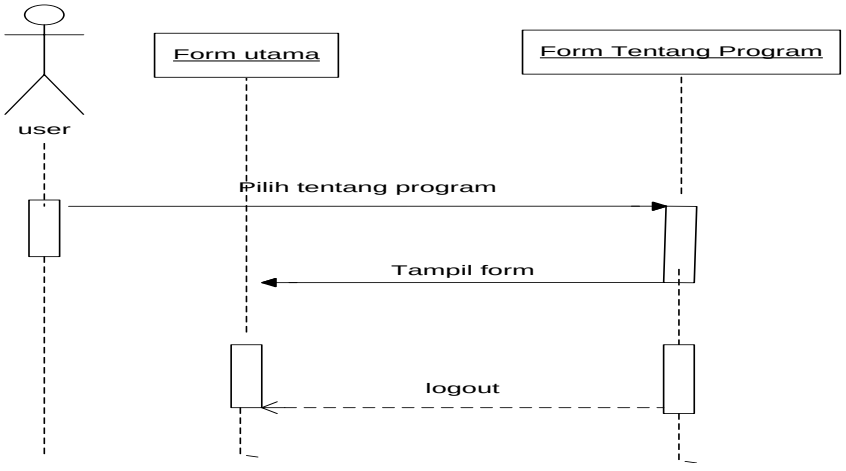
Sequence diagram data diagnosa menggambarkan interaksi antara objek pada proses data diagnosa dan mengenkrip, serta dekrip data diagnosa . *Sequence* diagram data diagnosa ditunjukkan pada gambar berikut ini:



Gambar III.10. *Sequence Diagram* Data Diagnosa

4. *Sequence Diagram* Tentang Program

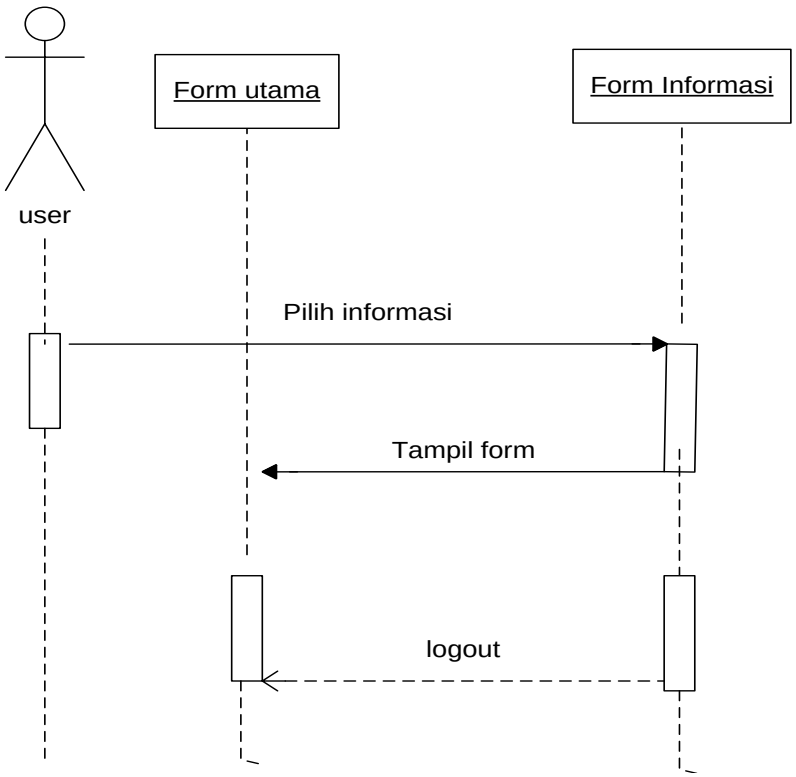
Sequence diagram tentang program menggambarkan interaksi antara *user* pada *form* tentang program. *Sequence* diagram tentang program ditunjukkan pada gambar dibawah ini:



Gambar III.11. *Sequence Diagram* Tentang Program

5. *Sequence Diagram* Informasi

Sequence diagram Informasi menggambarkan interaksi antara *user* pada *form* Informasi. *Sequence diagram* Informasi ditunjukkan pada gambar dibawah ini:



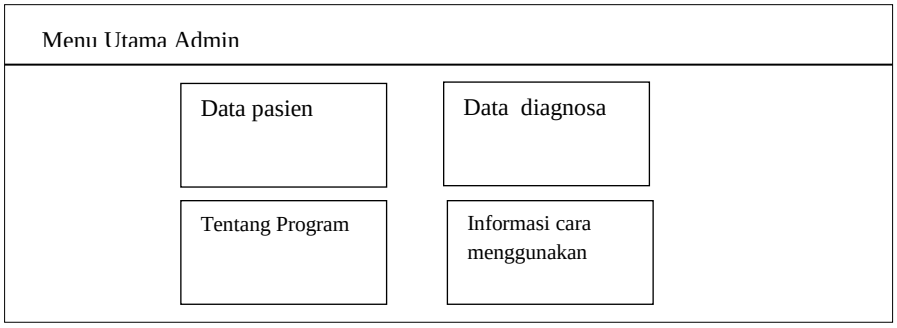
Gambar III.12. *Sequence Diagram* Informasi

III.4. *Desain User Interface*

1. Rancangan Antar Muka Menu Utama Admin

   Antar muka ini merupakan antar muka Admin yang berisi sedikit penjelasan tentang Aplikasi Keamanan *Database* Menggunakan Algoritma Vigenere dan

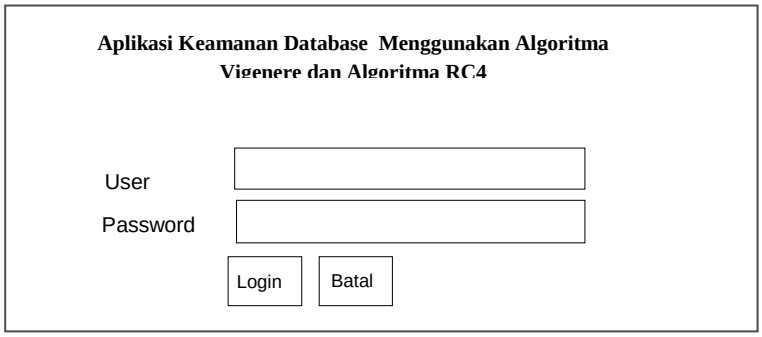
Algoritma RC4. Rancangan Antar muka beranda ditunjukkan pada gambar III.13 berikut ini :



Gambar III.13. Rancangan antar muka *form* Utama

2. Form Login

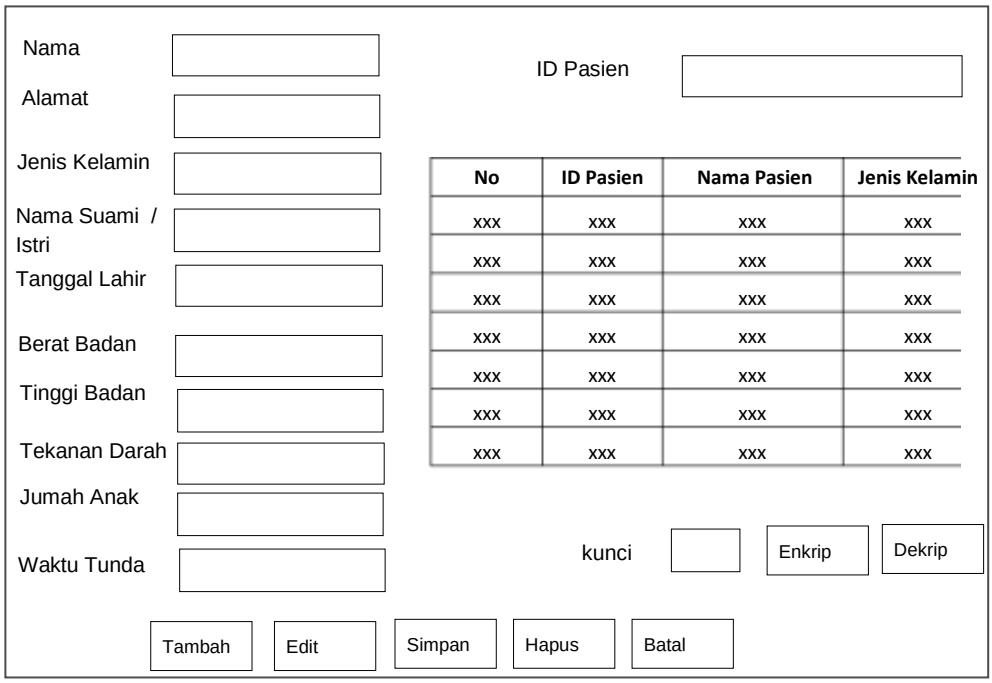
Disaat user pertama sekali membuka program maka akan dihadapkan oleh *form login* ini. Dimana user diminta untuk memasukkan *user id* dan *password* agar dapat mengakses ke menu utama program.



Gambar III.14. Rancangan antar muka *form login*

3. *Form* pasien

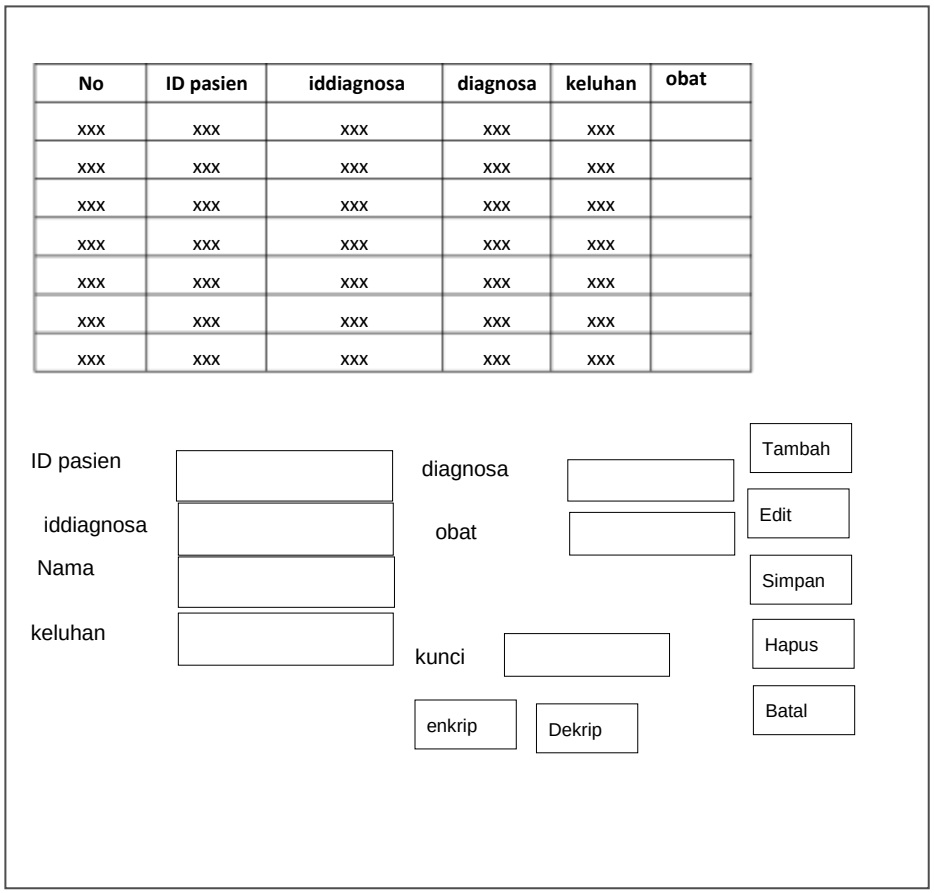
Form ini dirancang untuk menambah menghapus dan merubah data pasien selanjutnya untuk dilakukan enkripsi dan dekripsi yang. Data-data pasien yang melakukan konsultasi di inputkan berdasarkan data dari tempat penulis melakukan riset.



Gambar III.15. Rancangan antar muka *form* Pasien

4. *Form* diagnosa

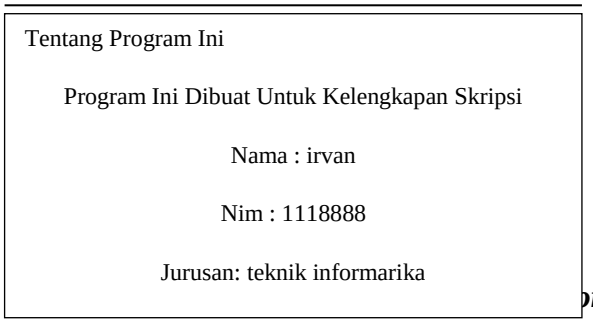
Rancangan *form* diagnosa berguna untuk menambah menghapus dan merubah data diagnosa selanjutnya untuk dilakukan enkripsi dan dekripsi. Data yang dinputkan berdasarkan dengan perusahaan penulis melakukan riset.



Gambar III.16. Rancangan antar muka *form* Diagnosa

5. Form tentang

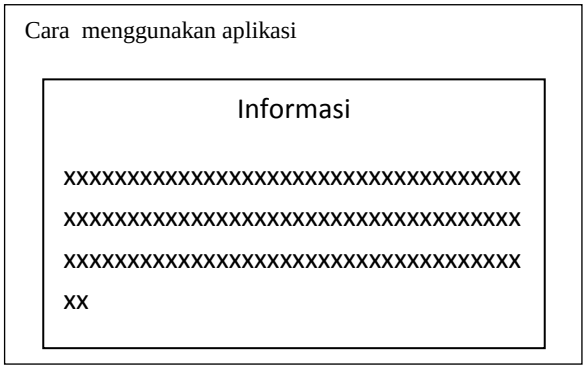
Form ini dirancang untuk menjelaskan penulis dan pembuat Aplikasi Keamanan *Database* Menggunakan Algoritma Vigenere dan Algoritma RC4.



***orm* Tentang**

6. Form Informasi

Form ini dirancang untuk menjelaskan cara penggunaan aplikasi Keamanan *Database* Menggunakan Algoritma Vigenere dan Algoritma RC4:



Gambar III.18. Rancangan antar muka *form* Informasi

III.5. Desain *Database*

Perancangan *database* berguna untuk menyimpan data-data yang saling berhubungan satu dengan yang lainnya. Dalam perancangan *database* di bentuk satu *file* yang berguna untuk menyimpan tabel-tabel yang diperlukan sebagai basis penyimpanan suatu data.

III.5.1. Kamus Data

Dibawah ini adalah kamus data atau referensi data yang ada pada basis data sistem yang akan dibangun :

1. Admin ={(**idadmin** +username+ pass)}
2. Diagnosa ={(**iddiagnosa**+idpasien+namapasien+keluhan+diagnose+ penanganan+obat)}
3. Pasien ={(**idpasien** + nama + alamat + tempat+tanggal +jeniskelamin + namasuis+berat+tinggi+jumlahanak+tekanandarah+waktutunda)}

Normalisasi *database* biasanya jarang dilakukan dalam *database* skala kecil dan dianggap tidak diperlukan pada penggunaan personal. Namun seiring dengan berkembangnya informasi yang dikandung dalam sebuah *database*, proses normalisasi akan sangat membantu dalam menghemat ruang yang digunakan oleh setiap tabel di dalamnya, sekaligus mempercepat proses permintaan data. Pada tahap ini semua data direkam tanpa *format* tertentu dan data bisa jadi mengalami duplikasi.

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)

   a. Tabel Normal Pertama

Idpasien	namapasien	Id diagnose	diagnosa	keluhan	Obat

2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

   a. Tabel pasien

Idpasien	namapasien	Alamat	jniskelamin	telp

   b. Tabel diagnosa

Iddiagnosa	idpasien	Diagnose	Keluhan	obat

   c. Tabel *user*

Nama_email	Password

Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

a. Tabel pasien

Idpasien*	namapasien	Alamat	jniskelamin	telp

b. Tabel diagnosa

Iddiagnosa*	idpasien	Diagnose	Keluhan	obat

c. Tabel *user*

Nama_email*	Password

III.5.2. Desain Tabel/ *File*

Pada sistem ini, digunakan *database SQL Server* dengan namadblisa menggunakan 3 tabel, yaitu tabel user, tabel pasien dan tabel diagnosa . Adapun struktur data dari tabel-tabel tersebut adalah sebagai berikut :

III.5.2.1. Struktur Tabel pasien

Tabel pasien digunakan untuk menyimpan *record* data pasien. Tabel pasien ditunjukkan pada tabel III.1 berikut ini :

Tabel III.1. Tabel Pasien

No	Field Name	Type	Width	Keterangan
1	Idpasien	Nchar	10	Idpasien
2	Nama	Nchar	10	Nama
3	Alamat	Nchar	25	Tempat

4	Tanggal	Date	Date	Tanggal
5	Jeniskelamin	Nchar	10	Alamat
6	Namasuis	Nchar	25	Namasuis
7	Berat	Int		Berat
8	Tinggi	Int		Tinggi
9	Tekanandarah	Nchar	25	Tekanan darah
10	Jumlahanak	Int		Jumlah anak
11	Waktutunda	Ncar	10	Waktu tunda

III.5.2.2. Struktur Tabel admin

Tabel admin digunakan untuk menyimpan *record* data *user* dengan properti atribut idadmin, *username* dan *password*. Tabel admin ditunjukkan pada tabel III.2 berikut ini ;

Tabel III.2. Tabel Admin

Field	*Type*	*Size*	Keterangan
Idadmin	Varchar	50	Idadmin
Username	Nchar	10	Username
Pass	Nchar	10	Password

III.5.2.3. Struktur Tabel diagnosa

Tabel diagnosa digunakan untuk menyimpan *record* data diagnosa pasien. Tabel diagnosa ditunjukkan pada tabel III.3 berikut ini :

Tabel III.3. Tabel Diagnosa

No	Field Name	Type	Width	Keterangan

1	Iddiagnosa	Nchar	10	Iddiagnosa
2	Idpasien	Nchar	10	Idpasien
3	Namapasien	Nchar	25	Namapasien
4	Keluhan	Nchar	100	Keluhan
5	Diagnose	Nchar	100	Diagnose
6	Penanganan	Nchar	100	Penanganan
7	Obat	Nchar	100	Obat