BAB IV

PERANCANGAN

4.1 Perancangan

Pada Bab III telah dijelaskan bahwa algoritma RSA memiliki 3 buah proses utama yaitu pembentukan kunci, proses enkripsi dan proses dekripsi. Diasumsikan proses pembentukan kunci berada pada sisi administrator untuk memutuskan menggunakan berapa panjang kunci dan juga menentukan kunci itu sendiri, proses enkripsi terjadi saat transaksi data ketika suatu *record* dimasukkan kedalam basis data oleh pengguna, proses dekripsi berada di sisi administrator basis data juga yang mempunyai wewenang melihat data yang dirahasiakan dalam keadaan tertentu.

Pada perancangan RSA pada oracle kali ini, digunakan beberapa kondisi tambahan sebagai berikut :

1. Penentuan kedua bilangan prima menggunakan fungsi random dengan kombinasi pengecekan nilai random harus bilangan prima.
2. Nilai e ditentukan dengan formula pencarian bilangan prima e dimulai dari nilai m dikurang 1 hingga e sama dengan 0. Dimana nilai e yang memenuhi adalah e yang bernilai prima. Digunakan pencarian nilai e dimulai dari nilai m dikurang satu agar didapatkan nilai e yang lebih besar jika dibandingkan dengan pencarian nilai e dimulai dari nilai 1 hingga m.
3. Nilai d ditentukan dengan k bernilai 1 hingga ditemukan nilai d yang integer dan nilai d tidak sama dengan nilai e. Ditambahkan kondisi nilai d

tidak sama dengan e untuk menghindari kedua pasangan kunci mempunyai nilai yang sama.

4. Tabel konversi yang digunakan untuk mengonversi pesan (M) menjadi bentuk bilangan dan sebaliknya adalah tabel yang nilai konversinya disusun sendiri oleh penulis bukan berdasarkan tabel yang telah ada, tabel ASCII contohnya.

   Tabel konversi dari *char* ke bilangan dan dari bilangan ke *char* ditunjukkan pada tabel 4.1 dan 4.2.

Tabel 4.1 Konversi dari *char* ke bilangan

char	bilangan	char	bilangan
a	32	s	52
b	33	t	53
c	34	u	54
d	35	v	55
e	36	w	56
f	37	x	57
g	38	y	58
h	39	z	59
i	41	1	72
j	42	2	73
k	43	3	74
l	44	4	75
m	45	5	76
n	46	6	77
o	47	7	78
p	48	8	79
q	49	9	81
r	51	0	82

Tabel 4.2 Konversi dari bilangan ke *char*

bilangan	char	bilangan	char
32	a	52	s
33	b	53	t
34	c	54	u
35	d	55	v
36	e	56	w
37	f	57	x
38	g	58	y
39	h	59	z
41	i	72	1
42	j	73	2
43	k	74	3
44	l	75	4
45	m	76	5
46	n	77	6
47	o	78	7
48	p	79	8
49	q	81	9
51	r	82	0

4.1.1 Perancangan Algoritma Pembentukan Kunci RSA di Oracle

Pembentukan kunci pada RSA sepenuhnya dapat diatur oleh pengguna dalam hal panjang kunci yang dipakai (tergantung kemampuan panjang rentang tipe data yang dipakai) dan bilangan yang digunakan untuk membentuk kunci. Kali ini algoritma pembentukan kunci dicoba diterapkan pada level PL/SQL di Oracle dengan kedua bilangan prima ditentukan dengan secara random pada *range* 2 sampai 300 dengan tipe data integer.

Dalam penghitungan nilai e, diterapkan formula pencarian bilangan prima dengan aturan pencarian nilai e adalah m dikurang 1 hingga e = 0. Dimana nilai e yang terpilih akan diuji bilangan prima ataukah bukan bilangan prima, jika nilai e

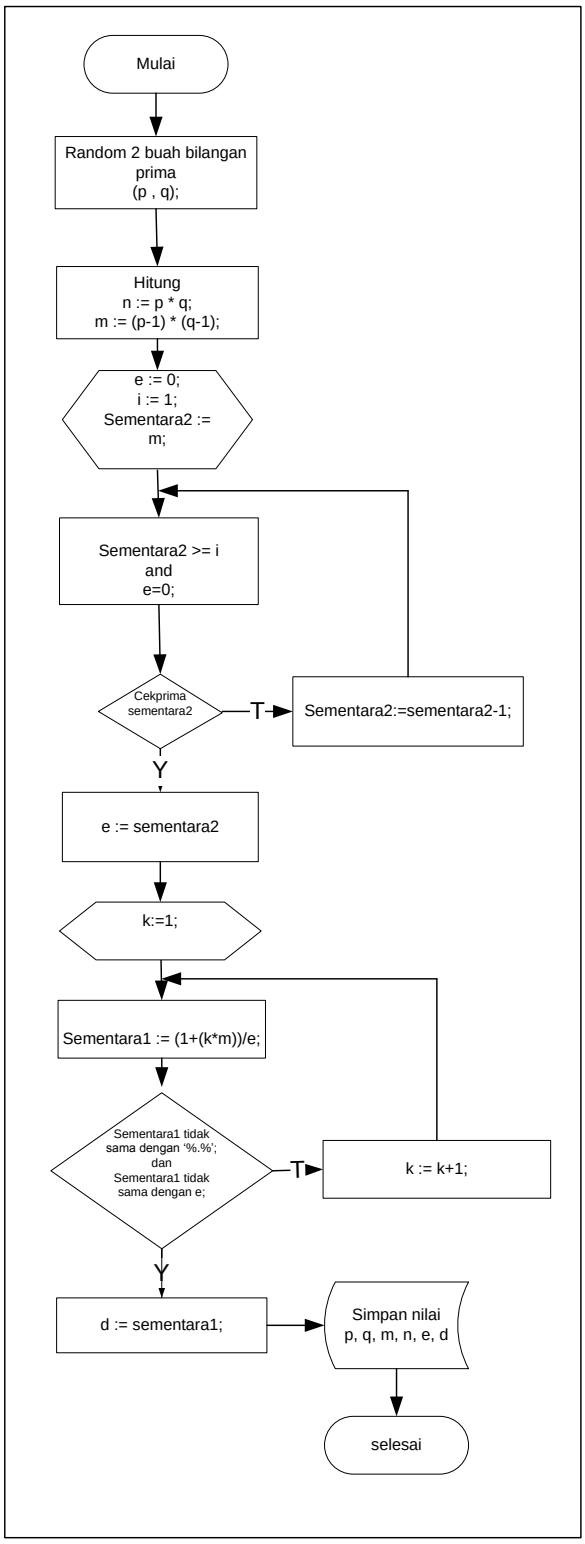
tersebut bilangan prima, maka nilai tersebut memenuhi dan diambil sebagai nilai e.

Sedangkan untuk pencarian nilai d, kondisi yang diterapkan adalah nilai d ditentukan dengan nilai awal k adalah 1 hingga ditemukan d yang integer dan nilai d tidak sama dengan nilai e.

Alur proses pembentukan kunci RSA sebagai berikut :

1. Tentukan dua bilangan prima yang diambil secara random, diwakilkan pada variabel p dan q.
2. Hitung nilai modulus (n) dan m.
3. Hitung nilai *enciphering exponent* (e), dengan pengecekan nilai e dimulai dr m-1 hingga e=0.
4. Hitung nilai deciphering exponent (d), dengan nilai k sebagai integer sembarang dimulai dari 1 hingga ditemukan nilai d yang integer dan d tidak sama dengan e.
5. Simpan nilai n, m, e, d kedalam basis data.

Flowchart Pembentukan Kunci RSA terdapat pada gambar 4.1.



Gambar 4.1

Flowchart Algoritma Pembentukan Kunci RSA di Oracle 10g

4.1.2 Perancangan Algoritma Enkripsi RSA di Oracle

Proses enkripsi diatur untuk aktif secara otomatis ketika pengguna memasukkan data pada kolom tertentu yang telah diberi *triggers* untuk membangkitkan fungsi enkripsi, kolom yang dienkripsi adalah kolom password dan kolom nomor CC. Secara umum, proses enkripsi sebagai berikut :

a. Konversi string ke dalam bentuk bilangan.

b. Lakukan pemenggalan menjadi beberapa blok.

c. Jalankan proses perhitungan enkripsi masing-masing blok

d. Samakan panjang setiap blok.

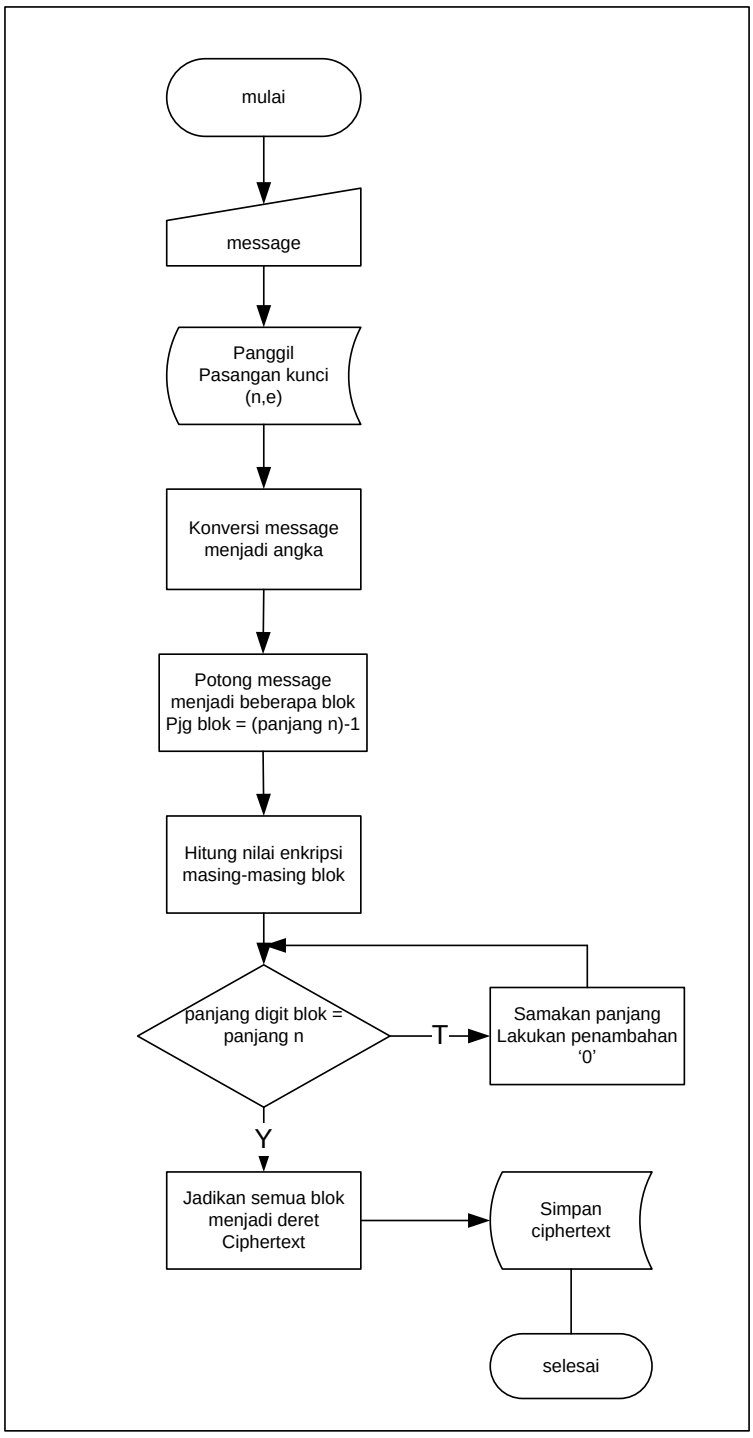
e. Semua blok disatukan menjadi satu deret ciphertext.

Alur yang terjadi dalam proses enkripsi jika dijabarkan secara detail sebagai berikut :

1. Pasangan kunci enkripsi diambil dr tabel kunci dan diwakilkan ke dalam suatu variabel.
2. Konversi data yang akan dienkrip menjadi bentuk bilangan dengan menggunakan *function* yang telah dibuat terlebih dahulu. *Function* untuk mengonversi ini bisa dibuat seseuai dengan ketentuan sendiri, untuk setiap huruf digantikan dengan angka yang ditentukan sendiri oleh penulis bukan berdasarkan ASCII sebagai contoh.
3. Lakukan pemotongan menjadi beberapa blok dengan terlebih dahulu menghitung panjang maksimal untuk setiap blok berdasarkan panjang modulus dikurang 1 (n-1) untuk menjaga selalu terpenuhinya syarat “number < modulus”.

4. Lakukan pemotongan deret menjadi blok, kemudian masing-masing blok dienkrip di kelas java menggunakan kunci enkripsi yang tersimpan pada basis data. Dalam kasus ini, untuk melakukan perhitungan perpangkatan digunakan sebuah fungsi yang memanggil kelas java agar menghindari ketidakmampuan tipe data pada PL/SQL menampung nilai dalam proses perpangkatan. Pada kelas java, proses yang terjadi hanya perhitungan formulas enkripsi dan dekripsi dengan variabel yang bertipe data BigInteger, kemudian hasilnya dikembalikan ke PL/SQL. Kelas java menerima 3 buah nilai, angka yang akan dienkripsi, *enciphering exponent* dan *modulus.*
5. Setelah masing-masing blok dienkrip, maka lakukan pengecekan panjang digit setiap bloknya. Panjang digit setiap blok harus sama dengan panjang digit modulus (n). Blok yang panjang digit kurang dari panjang digit modulus, ditambahkan angka 0 didepannya sebanyak kekurangan digitnya untuk menjaga konsistensi panjang blok.
6. Blok yang telah disesuaikan panjang digitnya kemudian dijadikan satu deret yang disebut deret *Ciphertext* untuk selanjutnya disimpan kedalam tabel pada basis data.

Flowchart proses enkripsi ditunjukkan pada gambar 4.2.



Gambar 4.2

Flowchart Algoritma Enkripsi RSA di Oracle 10g

4.1.3 Perancangan Algoritma Dekripsi RSA di Oracle

Secara umum, proses dekripsi data pada RSA adalah sebagai berikut:

a. Ambil data yang akan didekrip.
b. Lakukan pemenggalan deret menjadi blok.
c. Jalankan proses perhitungan dekripsi masing-masing blok.
d. Satukan hasil dekripsi menjadi satu deret plaintext.
e. Konversi *plaintext* menjadi *message*.

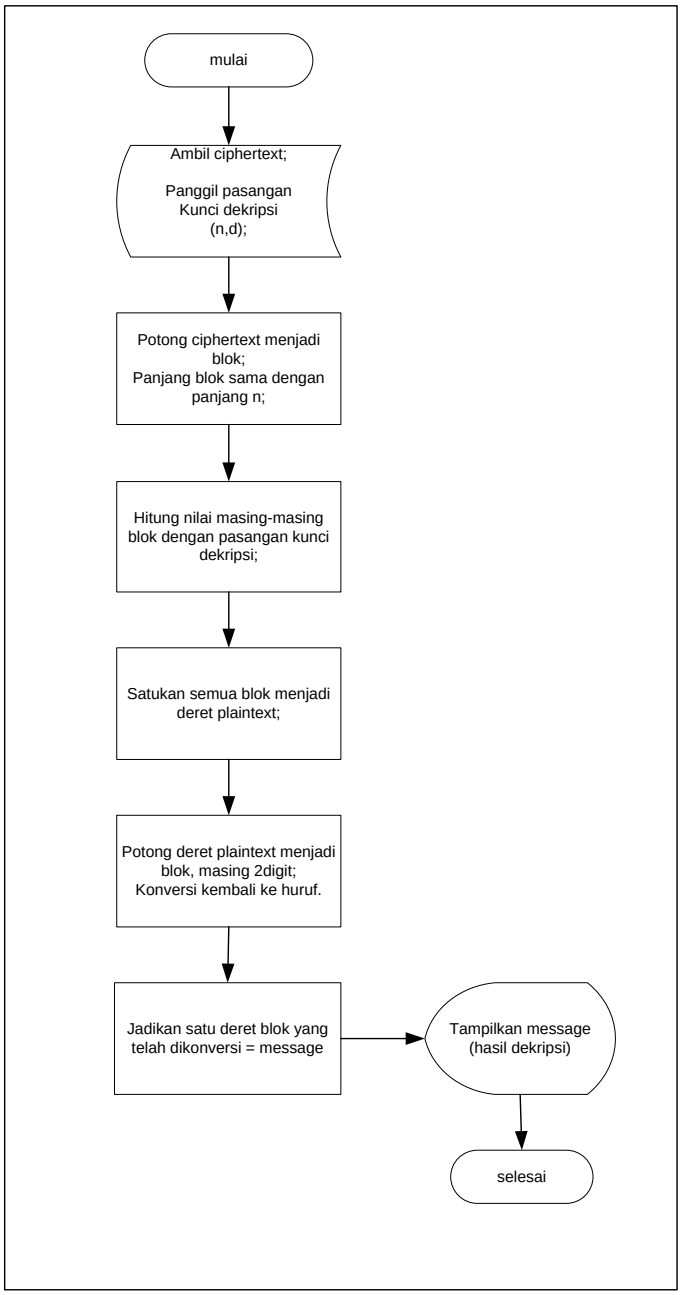
Proses perhitungan dekripsi dijalankan pada kelas java agar dapat menampung hasil dari penghitungan perpangkatan yang kemungkinan nilainya melebihi nilai yang mampu ditampung oleh tipe data integer PL/SQL Oracle.

Secara detail, proses dekripsi data menggunakan algoritma RSA pada Oracle adalah sebagai berikut :

1. Tentukan data yang akan didekrip, pada kasus ini data yang akan didekrip adalah data password dan nomor-cc.
2. Panggil pasangan kunci dekripsi dan diwakilkan pada variabel tertentu.
3. Bagi deret data *CipherText* menjadi blok berdasarkan panjang maksimal digit modulus (n).
4. Lakukan proses perhitungan masing-masing blok dengan mengirimkan masing-masing nilai blok dan pasangan kunci dekripsi ke kelas java yang telah disediakan sebagai tempat perhitungan perpangkatan.
5. Hasil perhitungan dari kelas java kemudian dijadikan deret yang disebut deret *plaintext,*

6. Konversi deret *plaintext* menjadi *message* untuk kemudian ditampilkan kepada pengguna.

Proses dekripsi diatas jika gambarkan kedalam flowchart ditunjukkan pada gambar 4.3.



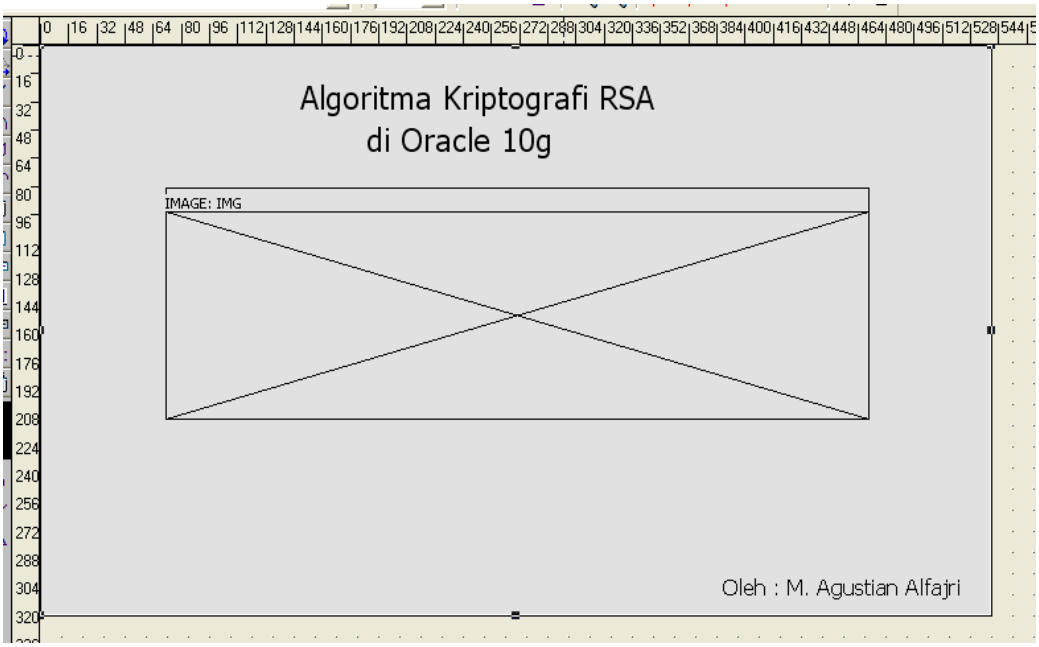
Gambar 4.3
Flowchart Algoritma Dekripsi RSA di Oracle 10g

4.2 Rancangan Antarmuka

Rancangan antar muka ini akan dibuat menggunakan Form Developer PL/SQL sebagai simulasi dari proses-proses yang yang terdapat pada RSA untuk implementasi sebenarnya pada aplikasi program yang menerapkan RSA.

4.2.1 Rancangan Menu Utama

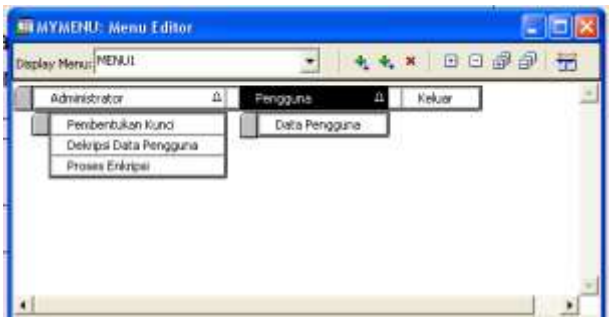
Menu utama ini dirancangan pada sisi administrator yang bertugas menjaga basis data. Termasuk salah satunya bertugas untuk membentuk pasangan kunci, saat pertama kali basis data akan digunakan.



Gambar 4.4
Rancangan Antarmuka Menu Utama

Pada bagian atas *canvas* terdapat panel menu yang dikelompokkan seperti pada gambar 4.5. Pada gambar 4.5 ditunjukkan pada panel menu, terdapat menu Administrator yang di dalamnya terdapat sub menu Pembentukan Kunci, Dekripsi Data Pengguna, dan Proses Enkripsi.

Kemudian terdapat menu Pengguna yang di dalamnya terdapat sub menu Data Pengguna untuk menampilkan data pengguna yang ada di dalam basis data.



Gambar 4.5

Rancangan Antarmuka Panel Menu

4.2.2 Rancangan Antarmuka Pembentukan Kunci

Untuk membentuk pasangan kunci RSA, disediakan satu form bagi administrator membentuk pasangan kunci. Pembentukan pasangan kunci RSA dilakukan dengan pemilihan bilangan prima secara acak (random). Rancangan tampilan awal form pembentukan kunci terlihat pada gambar 4.6 :

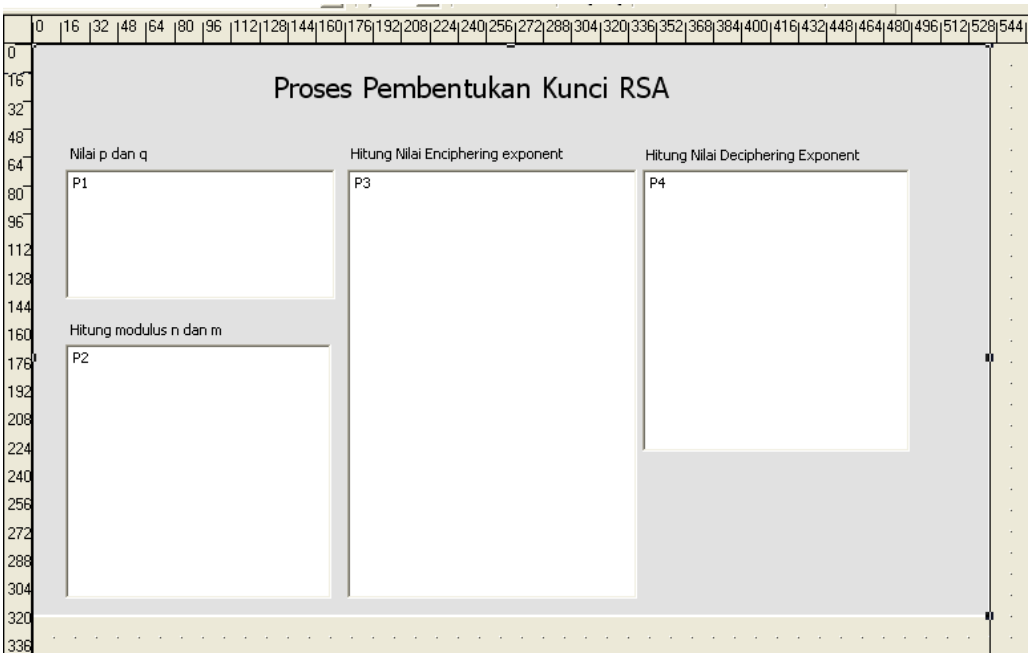


Gambar 4.6

Rancangan antarmuka pembentukan kunci

Gambar 4.6 merupakan tampilan form bagi administrator untuk proses pembentukan kunci. Terdapat satu buah tombol yaitu tombol “Buat Kunci” yang meng-*generate* dua buah bilangan prima yang akan diproses menjadi pasangan kunci enkripsi dan dekripsi. Pada *frame* detail kunci terdapat tabel yang terdiri dari enam kolom sebagai kolom untuk menampilkan detail hasil pembentukan kunci yang nantinya disimpan kedalam tabel kunci.

Tombol “Proses” merupakan tombol pemanggilan interface penjelasan mendetail bagaimana langkah sehingga dihasilkan pasangan kunci publik (n, e) dan pasangan kunci privat (n, d) seperti yang ditunjukkan pada gambar 4.7.

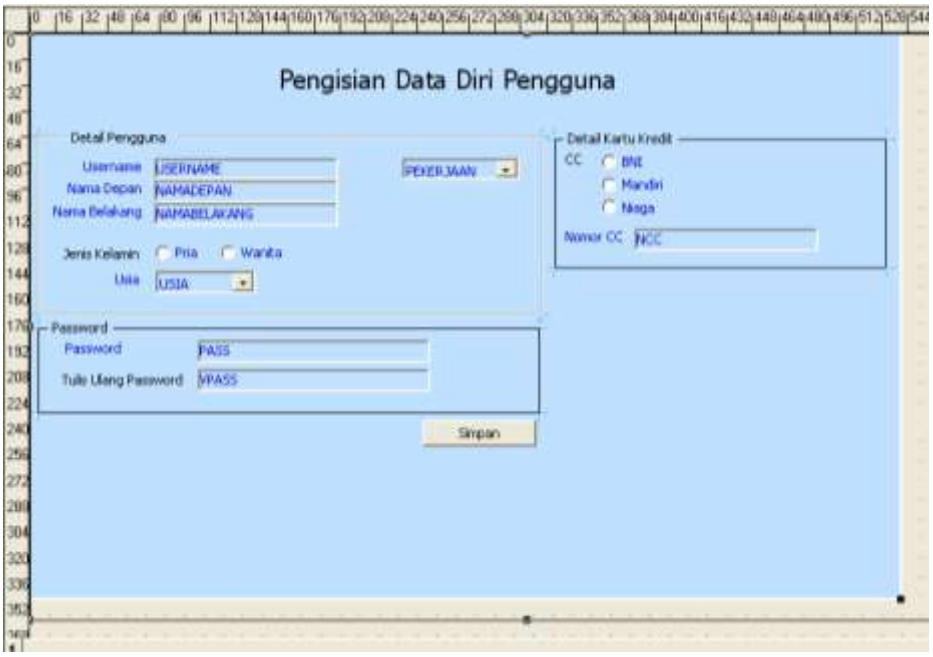


Gambar 4.7

Rancangan Antarmuka Proses Pembentukan Kunci

4.2.3 Rancangan Antarmuka Form Data Diri Client

Gambar 4.8 berikut merupakan rancangan antarmuka form pengisian data diri *client* yang data di dalamnya akan dirahasiakan (dienkrip) menggunakan algoritma RSA.



Gambar 4.8

Rancangan antarmuka form data diri client

Gambar 4.8 rancangan form masukan data diri pengguna, terdiri dari 3 buah *frame* yaitu : Detail Pengguna , Password dan Detail Kartu Kredit.

Detail Pengguna akan digunakan sebagai *frame* yang didalamnya terdapat kolom untuk mengisi username, nama depan, nama belakang, jenis kelamin, golongan umur dan jenis pekerjaan.

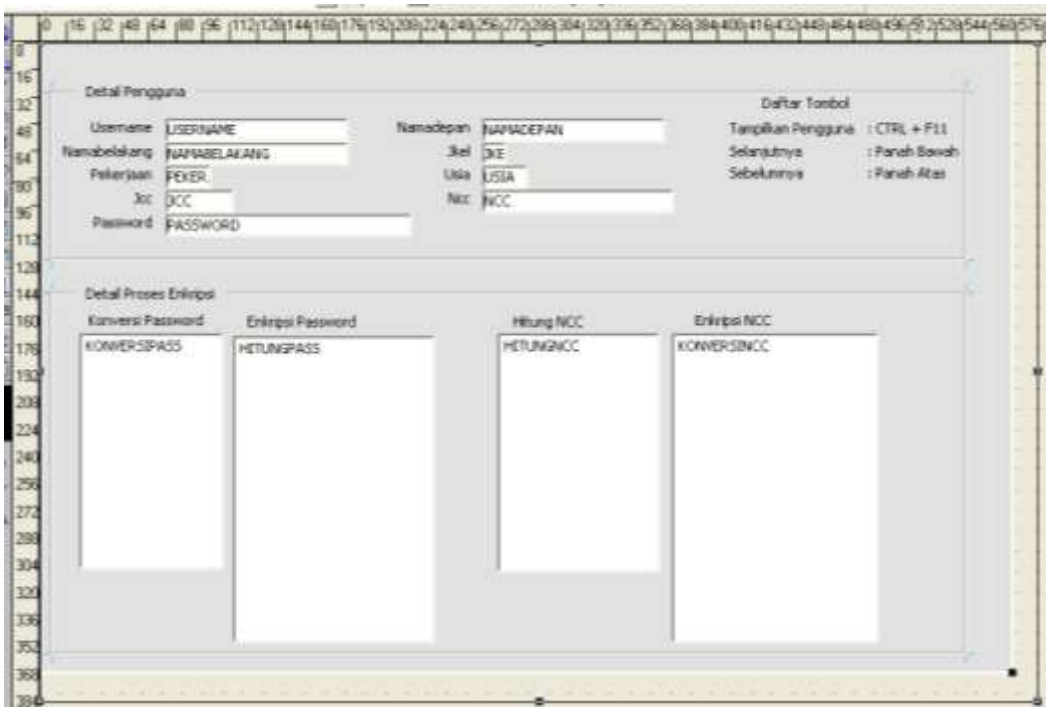
Frame Password berisi dua buah kolom sebagai tempat masukan password dan Tulis Ulang Password yang mana data password ini adalah salah

satu data yang akan dienkrip. Ditambahkan kolom tulis ulang password untuk memastikan bahwa password yang diketik sesuai dengan yang diinginkan.

Frame Detail Kartu Kredit berisi data kartu kredit, di dalamnya terdapat pemilihan jenis kartu kredit serta kolom inputan nomor kartu kredit. Nomor kartu kredit merupakan salah satu data yang juga akan dienkrip.

4.2.4 Rancangan Antarmuka Proses Enkripsi

Antarmuka untuk melihat proses enkripsi data diri pengguna berada di sisi administrator. Pada antarmuka proses enkripsi ditunjukkan detail pengguna dan detail proses data diri pengguna dienkrip.



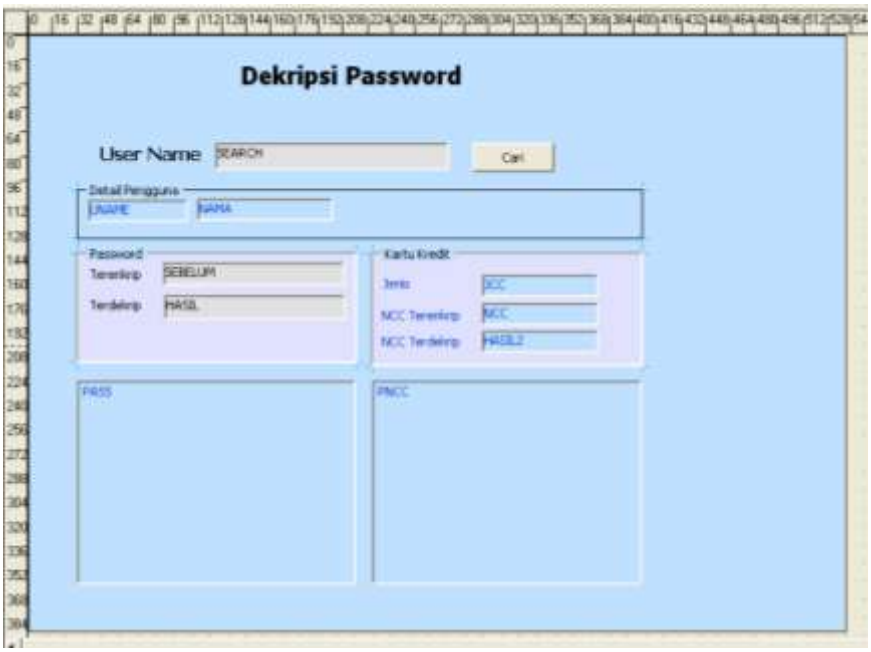
Gambar 4.9

Rancangan antar muka Detail User dan Proses Enkripsi

Pada gambar 4.9 frame detail pengguna menjadi *master detail* yang akan menjadi acuan dalam pemunculan data detail pada *frame* proses dibawahnya yaitu Detail Proses Enkripsi. Sedangkan yang menjadi penghubung kedua blok adalah kolom username.

4.2.5 Rancangan Antarmuka Proses Dekripsi

Proses dekripsi dijalankan dengan sistem pencarian data berdasarkan username pengguna, yang jika data ditemukan akan ditampilkan beberapa data pengenal pengguna seperti username dan nama lengkap. Perancangan antarmuka proses dekripsi ditunjukkan pada gambar 4.10.



Gambar 4.10

Rancangan antarmuka Dekripsi

Pada gambar 4.10 , *search* merupakan kolom inputan yang dijadikan acuan dalam pencarian data pada basis data, yang mana sebagian data pengguna tersebut akan ditampilkan pada kolom uname, nama, jcc, ncc dan pass yang

berada pada detail pengguna sebagai informasi dari pengguna yang datanya akan didekrip. Sedangkan pada *frame* Password dan *frame* Kartu Kredit sendiri adalah kelompok kolom yang menampikan data yang telah dienkrip dan disimpan di basis data (kolom sebelum, kolom ncc) dan data yang telah didekrip kembali (kolom hasil, kolom hasil2).

Dua buah kolom dibawahnya (kolom PASS dan kolom PNCC) adalah kolom dimana dijelaskan langkah yang terjadi selama proses dekripsi berlangsung hingga menghasilkan data yang "asli" dari data yang tersimpan di basis data.

4.3 Analisa Perancangan

Dalam proses pengimplementasian algoritma kriptografi RSA di Oracle 10g, dilakukan ujicoba untuk menggunakan P dan Q yang nilainya lebih dari 2 digit, maka dihasilkan pasangan kunci yang memiliki panjang digit lebih dari 3, ketika dilakukan perhitungan perpangkatan, tipe data integer pada PL/SQL Oracle tidak dapat menampung nilai hitungannya, oleh karenanya untuk melakukan perhitungan perpangkatan pada proses enkripsi dan dekripsi dengan bilangan eksponen lebih dari 3 digit, dibutuhkan tipe data yang mampu menampung perhitungan tersebut dimana akan menghasilkan nilai yang sangat besar melebihi *range* nilai tipe data integer, -32768 s/d 32767.

Untuk mengatasi kendala keterbatasan *range* tipe data integer pada PL/SQL, maka digunakan satu kelas Java sebagai tempat dilakukan proses perhitungan perpangkatan dan modulus, dengan menggunakan tipe data bigInteger

yang bersifat *unbounded* atau tak berbatas, yang nilai akhirnya dikembalikan pada PL/SQL untuk dilanjutkan pada proses pembagian blok dan penderetan.