BAB III
ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai Aplikasi Keamanan Database Menggunakan Metode elgamal yang meliputi analisa sistem dan desain sistem.

III.1. Analisis Masalah

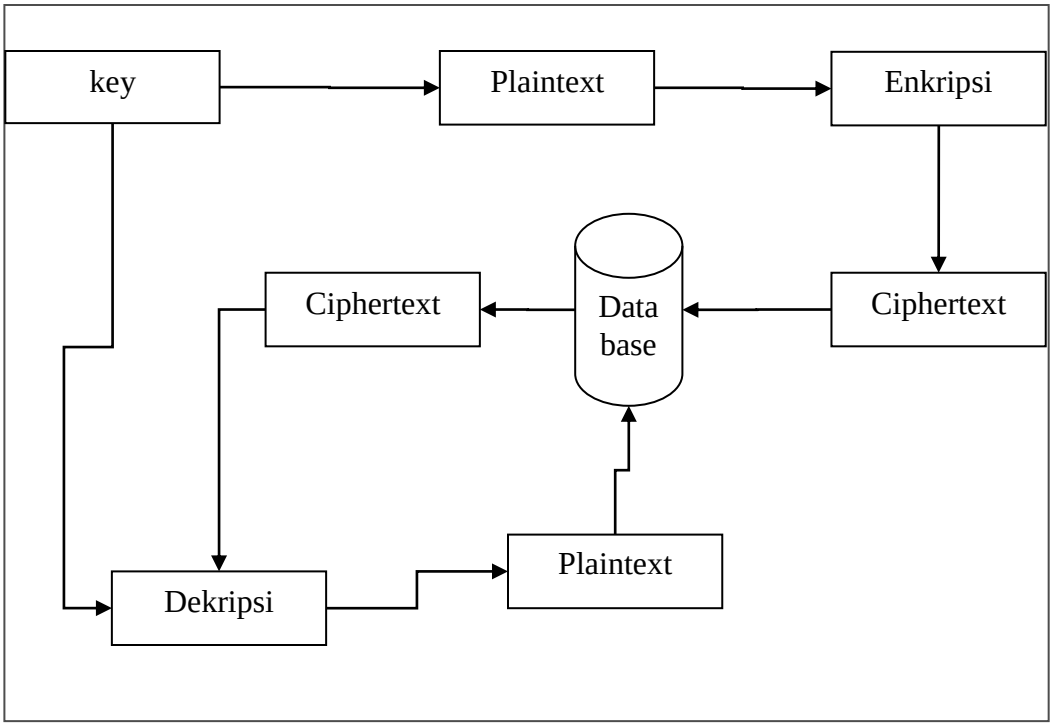
Adapun analisa masalah pada Aplikasi Keamanan Database Menggunakan Metode Elgamal yaitu :

1. Sering terjadinya pengaksesan data oleh pihak-pihak tertentu yang tidak berkepentingan sehingga data tidak lagi terjamin keasliannya.
2. Banyaknya pemalsuan suatu data informasi yang ada dalam *database* sehingga dapat merugikan pemilik *database* tersebut.

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 Metode Elgamal. Aplikasi ini adalah salah satu alat yang diyakini mampu memberikan konstribusi positif dalam menjamin keamanan *database*.

III.1.1. Blok Diagram

Blok diagram adalah diagram dari sebuah system, di mana bagian utama atau fungsi yang diwakili oleh blok dihubungkan dengan garis, yang menunjukkan hubungan dari blok.



Gambar III.1. Blok Diagram

III.2. Metode Elgamal

Algoritma ElGamal merupakan algoritma dalam kriptografi yang termasuk dalam kategori algoritma asimetris. Keamanan algoritma ElGamal terletak pada kesulitan penghitungan logaritma diskret pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar.

Algoritma ElGamal mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini mempunyai kerugian pada *cipherteksnya* yang mempunyai panjang dua kali lipat dari *plainteksnya*. Akan tetapi, algoritma ini mempunyai kelebihan pada enkripsi. Untuk *plainteks* yang sama,algoritma ini memberikan *cipherteks* yang berbeda (dengan

kepastian yang dekat) setiap kali plainteks dienkripsi. Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini merupakan *cipher* blok, yaitu melakukan proses enkripsi pada blok-blok *plainteks* dan menghasilkan blok-blok *ciphertek*s yang kemudian dilakukan proses dekripsi dan hasilnya digabungkan.

III.2.1. Proses Pembentukan Kunci

Pembentukan kunci terdiri atas pembentukan kunci publik dan kunci rahasia. Pada proses ini dibutuhkan sebuah bilangan prima p yang digunakan untuk membentuk grup Zp*, elemen primitif α dan sembarang $a \in \{0,1,..., p-2\}$.

Kunci publik algoritma ElGamal terdiri atas pasangan 3

bilangan (p,α ,β) di mana

$$\beta = \alpha^a \bmod p \quad \text{.....(1)}$$

Sedangkan kunci rahasianya adalah bilangan a tersebut.

Proses pembentukan kunci untuk algoritma ElGamal terdiri atas:

a. Penentuan bilangan prima aman yang bernilai besar

b. Penentuan elemen primitif

c. Pembentukan kunci berdasarkan bilangan prima aman dan elemen primitif

III.2.2. Proses Enkripsi

Proses enkripsi menggunakan kunci publik (p,α ,β) dansebuah bilangan integer acak k ($k \in \{0,1,..., p-1\}$) yang dijaga kerahasiaannya oleh penerima yang mengenkripsi pesan. Untuk setiap karakter dalam pesan dienkripsi dengan menggunakan bilangan k yang berbeda-beda. Satu

karakter yang direpresentasikan dengan menggunakan bilangan bulat ASCII akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai (*r*, *t*).

Langkah proses enkripsi:

a. Ambil sebuah karakter dalam pesan yang akan dienkripsi dan transformasi karakter tersebut ke dalam kode ASCII sehingga diperoleh bilangan bulat *M*.

b. Hitung nilai *r* dan *t* dengan persamaan berikut:

$$r = \alpha \text{ k } (\text{mod } p) \quad \text{..... (2)}$$

$$t = \beta \text{ k } M \ (\text{mod } p) \quad \text{.....(3)}$$

c. Diperoleh cipherteks untuk karakter *M* tersebut dalam blok (*r*, *t*)

d. Lakukan proses di atas untuk seluruh karakter dalam pesan termasuk karakter spasi.

III.2.3. Proses Dekripsi

Dekripsi dari *cipherteks* ke *plainteks* menggunakan kunci rahasia *a* yang disimpan kerahasiaannya oleh penerima pesan.

Teorema:

Diberikan (p,α ,β) sebagai kunci public dan *a* sebagai kunci rahasia pada algoritma ElGamal. Jika diberikan cipherteks (*r*, *t*),

$$\text{maka } M = t\,(ra)\ \text{-1 mod } p \quad \text{..... (4)}$$

dengan *M* adalah plainteks.

$$\text{Di mana nilai } (ra)\ \text{-1} = r\ \text{-}a = r\ p\text{-}1\text{-}a \ \text{mod } p \quad \text{..... (5)}$$

Langkah proses dekripsi:

a. Ambil sebuah blok cipherteks dari pesan yang telah dienkripsikan pengirim.

b. Dengan menggunakan *a* yang dirahasiakan oleh penerima, hitung nilai Plainteks dengan menggunakan “persamaan (4)” dan “persamaan(5)”. (Faqihuddin Al-Anshori; 2014).

Studi Kasus:

Untuk melakukan proses elgamal maka perlu dilakukan pembentukan kunci (*generate key*). Langkah-langkah dalam pembuatan kunci adalah sebagai berikut :

Pilih sembarang bilangan prima p, dengan syarat p > 255.

Pilih bilangan acak g dengan syarat g < p.

Pilih bilangan acak x dengan syarat $1 \leq x \leq p - 2$.

Hitung y = g ^ x mod p.

Perhitungan Pembentukan Kunci.

Misalkan A membangkitkan pasangan kunci dengan memilih bilangan :

p = 257, g = 11, x = 13, kunci acak k = 182.

Kemudian p, g, x digunakan untuk menghitung y :

y = g * x mod p

y = 1113 mod 257 = 22

jadi kunci public A adalah y = 22, g = 11, p = 257 dan kunci private A adalah x =13, p = 257.

Perhitungan enkripsi:

Dalam melakukan enkripsi plainteks atau string yang digunakan perlu dirubah menjadi kode ASCII untuk studi kasus ini kitaakan menggunakan kata "TES".

berikut nilai ASCII nya T = 84, E = 69, S = 83. kemudian plaintekstersebut akan dimasukkan kedalam blok nilai m yaitu m1 = 84, m2 = 69 dan m3 = 83.

kemudian kita dapat menghitung enkripsi untuk mendapatkan nilai ai dan bi dari masing-masing blok seperti berikut.

Perhitungan a1 :

ai = g ^ k mod p

a1 = 3414 mod 257 = 189

Perhitungan b1 :

b1 = y ^ k * m1 mod p

b1 = 2093 * 84 mod p = 143

Perhitungan a2 :

ai = g ^ k mod p

a2 = 3414 mod 257 = 189

Perhitungan b2 :

bi = y ^ k * m2 mod p

b2 = 2093 * 69 mod p = 145

Perhitungan a3 :

ai = g ^ k mod p

a3 = 3414 mod 257 = 189

Perhitungan b3 :

bi = y ^ k * m3 mod p

b3 = 2093 * 83 mod p = 126

maka hasil dari enkripsi nilai m1 m2 m3 menghasilkan a1, b1, a2, b2, a3, b3

ebagai berikut :

hasil enkripsi = 189 143 189 145 189 126

perhitungan dekripsi :

untuk melakukan dekripsi kita akan merubah nilai *chippertext* yang sudah di enkripsi sebelumnya dan memasukkan kedalam blok msebagai berikut :

hasil enkripsi = 189 143 189 145 189 126

m1 = 189, m2 = 143, m3 = 189, m4 = 145, m5 = 189, m6 = 126.

perhitungan dekripsi:

mi = bi * (ai ^ p-1-x) mod p

m1 = 143 * (189 ^ 257 -1 - 13) mod 189 = 84

m3 = 145 * (189 ^ 257 -1 - 13) mod 189 = 69

m4 = 126 * (189 ^ 257 -1 - 13) mod 189 = 83

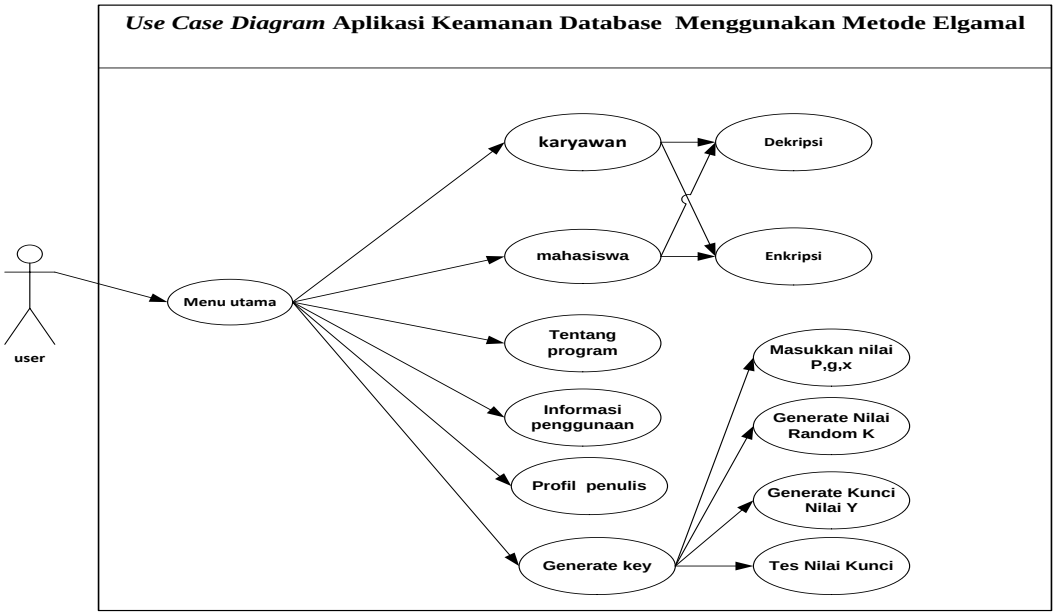
maka hasil dekripsi kita akan mendapatkan nilai ASCII 84, 69 dan 83. dimana plainteks nya adalah 84 = T, 69= E dan 83 = S.

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

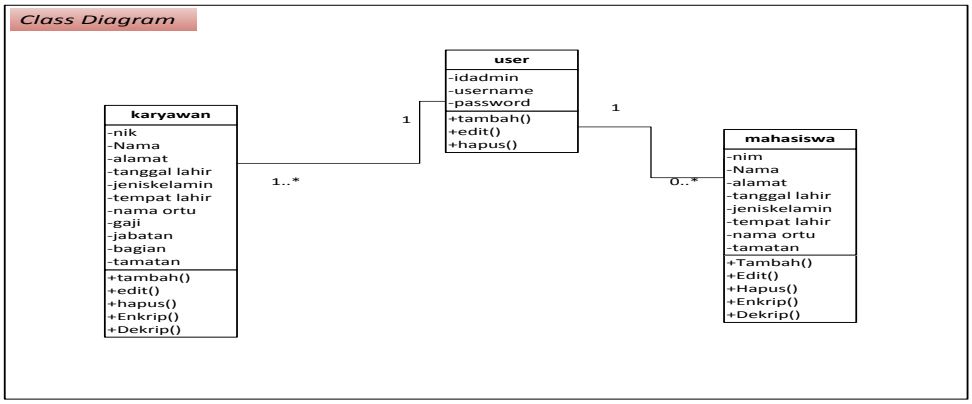
Secara garis besar, proses sistem yang akan dirancang digambarkan dengan *usecase diagram* yang terdapat pada Gambar III.2 :



Gambar III.2. *Use Case* Diagram Aplikasi Keamanan Database Menggunakan Metode Elgamal

III.3.2. *Class* Diagram

Rancangan kelas-kelas yang akan digunakan pada sistem yang akan dirancang dapat dilihat pada gambar III.3 :



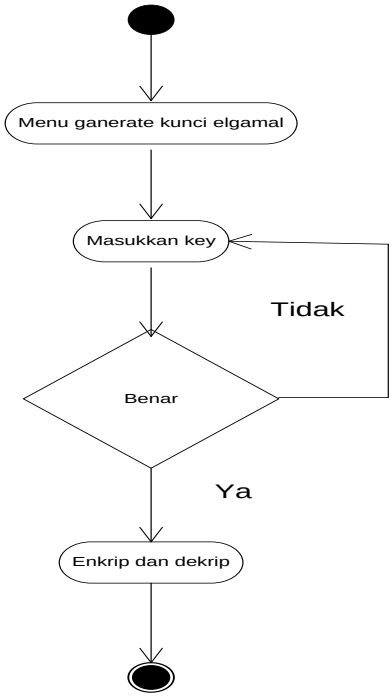
Gambar III.3. *Class Diagram* Aplikasi Keamanan Database Menggunakan Metode Elgamal

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 Generate kunci Elgamal

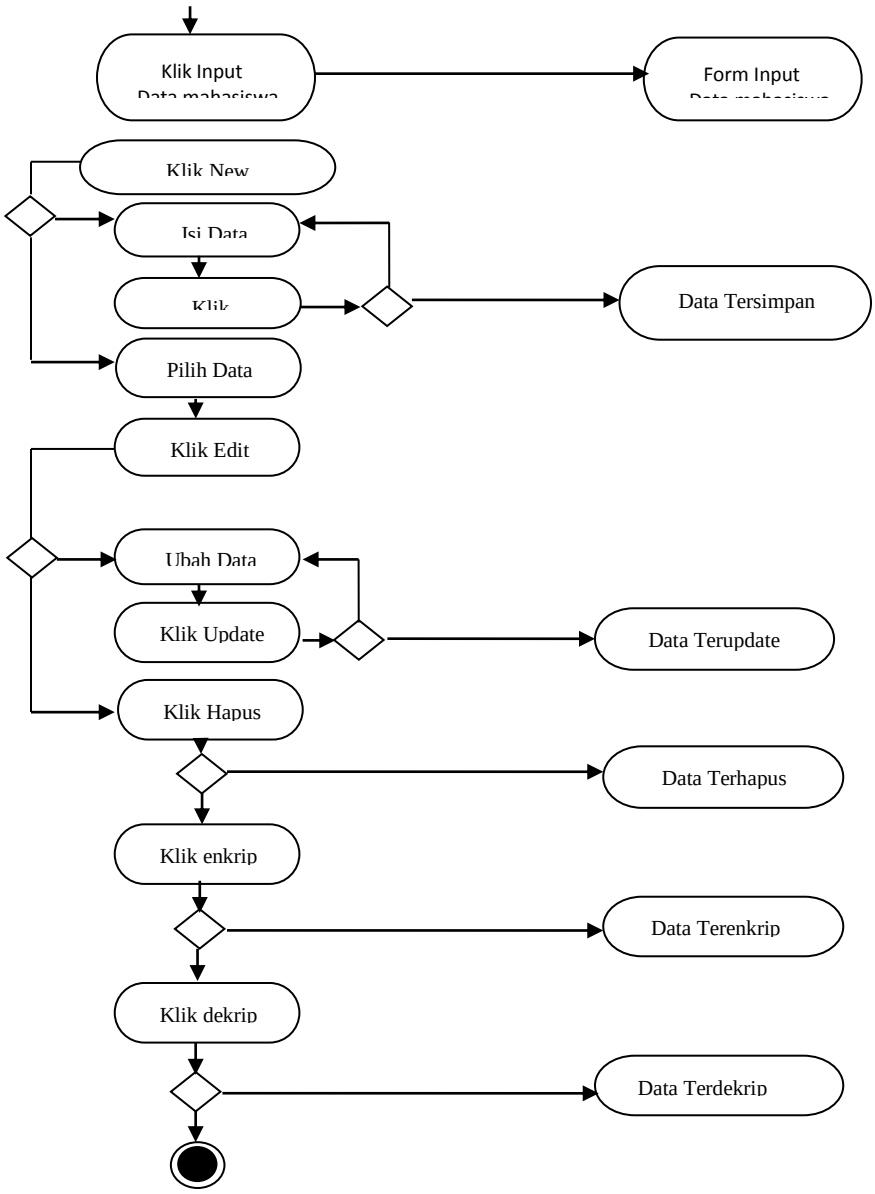
Activity diagram Generate kunci Elgamal merupakan *activity diagram* untuk proses Generate kunci elgamal. *Activity diagram Generate* kunci elgamal ditunjukkan pada gambar berikut ini:



Gambar III.4. *Activity Diagram* Generate Kunci Elgamal

2. *Activity* Diagram data mahasiswa

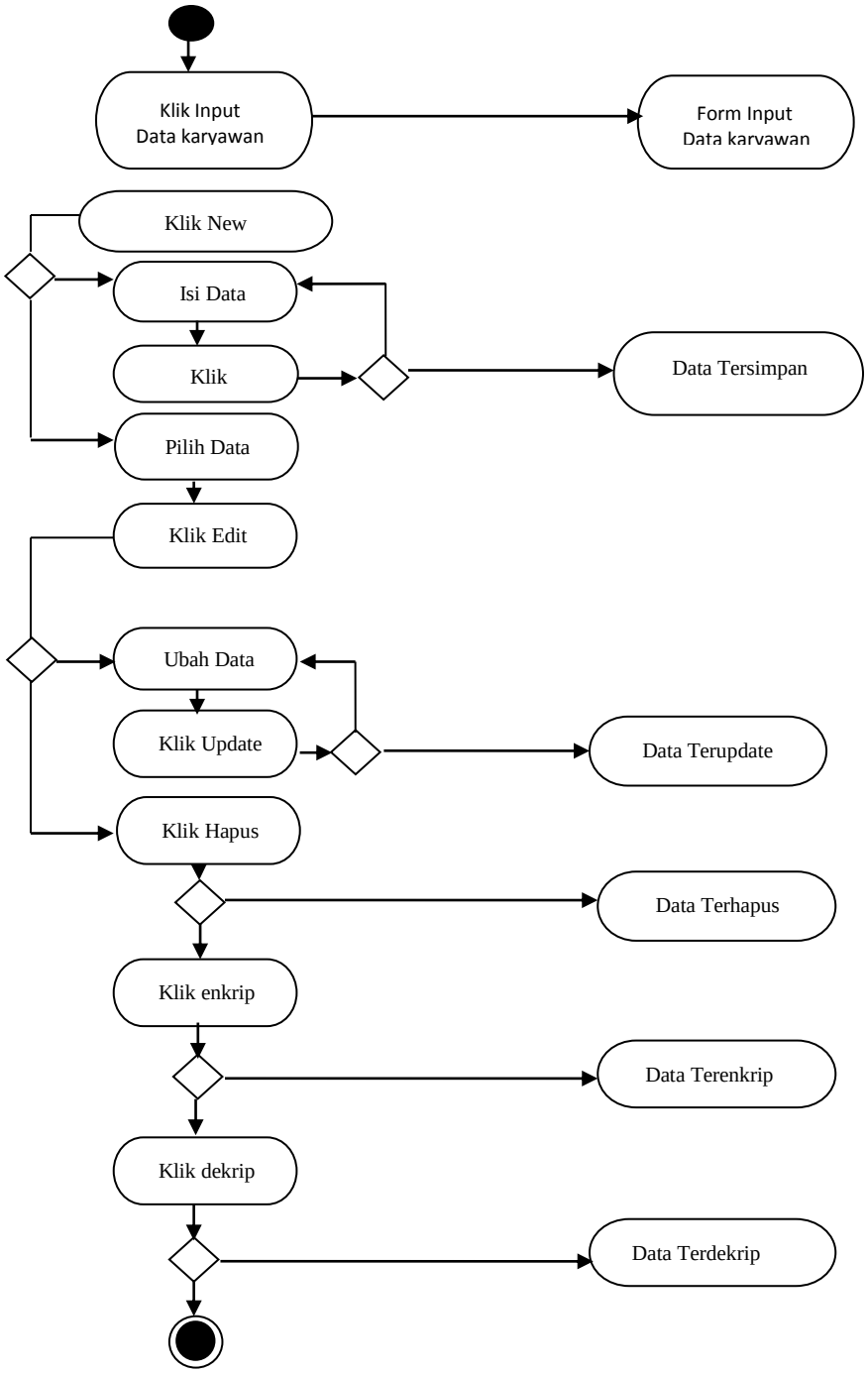
Activity diagram data mahasiswa merupakan *activity diagram* untuk inputan data mahasiswa. *Activity diagram* data mahasiswa ditunjukkan pada gambar berikut ini:



Gambar III.5. *Activity Diagram* Data Mahasiswa

3. *Activity* Diagram data karyawan

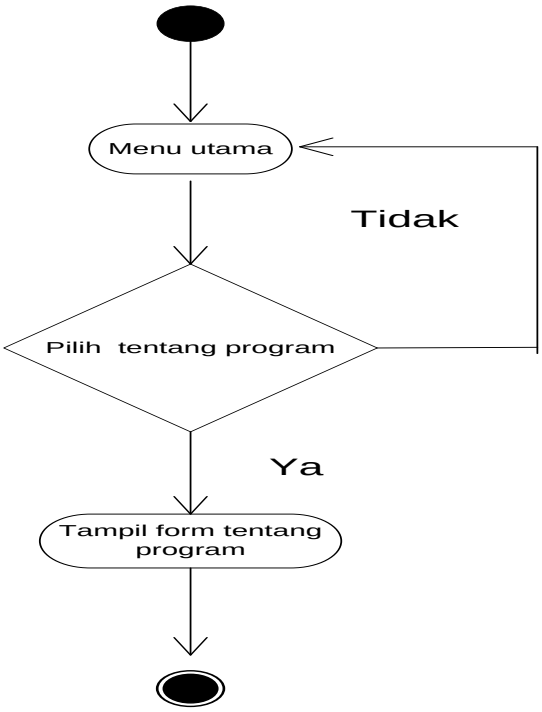
Activity diagram data karyawan merupakan *activity diagram* untuk proses karyawan. *Activity diagram* data karyawan ditunjukkan pada gambar berikut ini:



Gambar III.6. *Activity Diagram* Data Karyawan

4. *Activity* Diagram Tentang Program

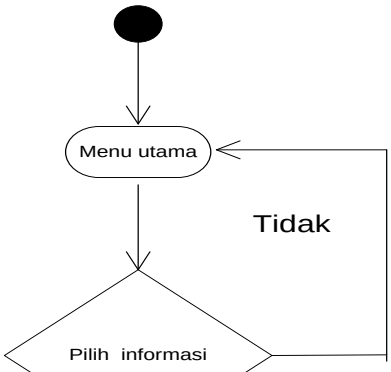
Activity diagram tentang program merupakan *activity diagram* untuk melihat form tentang Program. *Activity diagram* tentang program ditunjukkan pada gambar berikut ini:



Gambar III.7. *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:



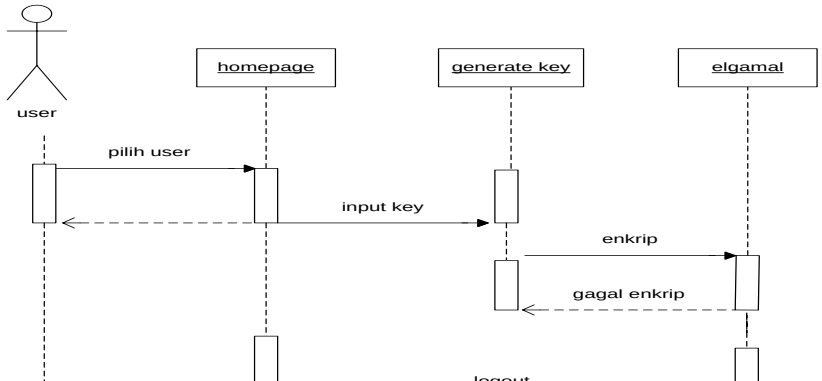
Gambar III.8. *Activity Diagram* Informasi

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 Generate Key*

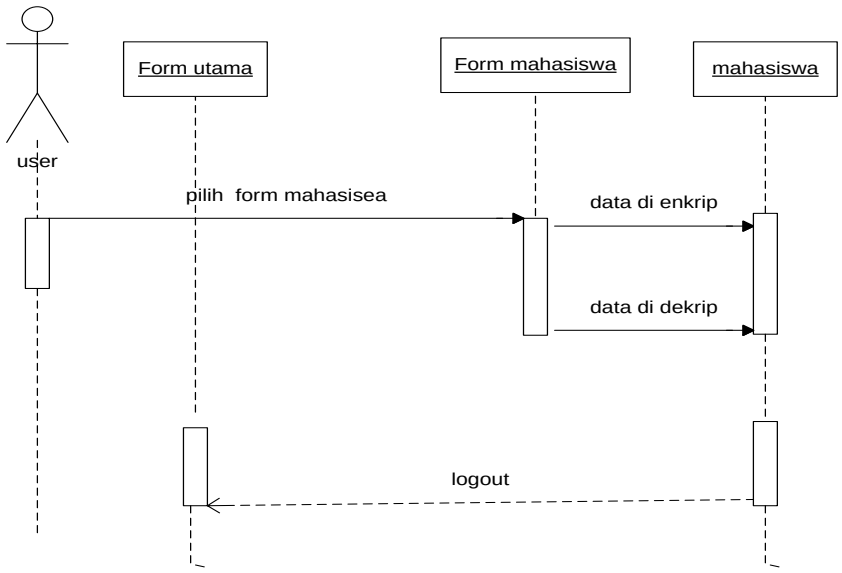
Proses *sequence generate Key*adalah user memasukkan *key* pada *form generate key*, maka data akan di enkrip dan dekrip. *Sequence diagram generate key* ditunjukan pada gambar berikut ini :



Gambar III.9. *Diagram Sequence* Generate Key

2. *Sequence Diagram* Data Mahasiswa

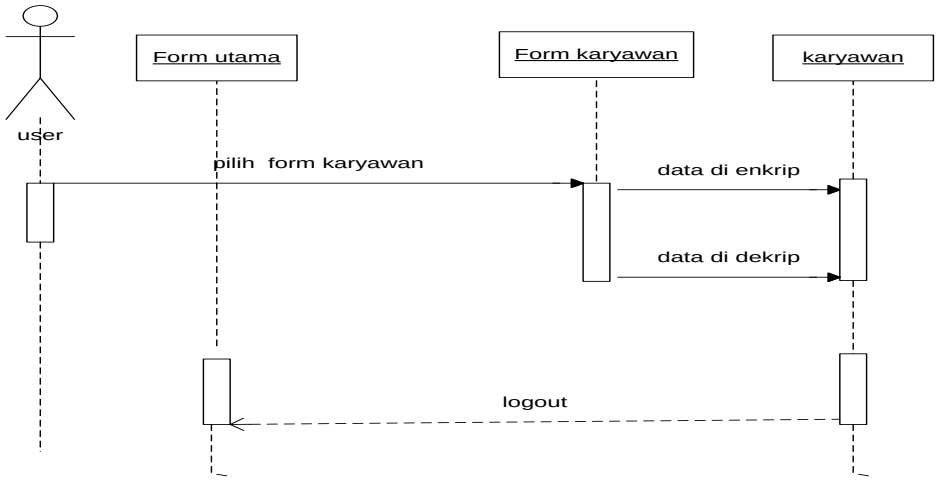
Sequence diagram data mahasiswa menggambarkan interaksi antara objek pada proses data mahasiswa dan mengenkrip, serta dekrip data mahasiswa . *Sequence diagram* data mahasiswa ditunjukkan pada gambar dibawah ini:



Gambar III.10. *Sequence Diagram* Data Mahasiswa

3. *Sequence Diagram* Data Karyawan

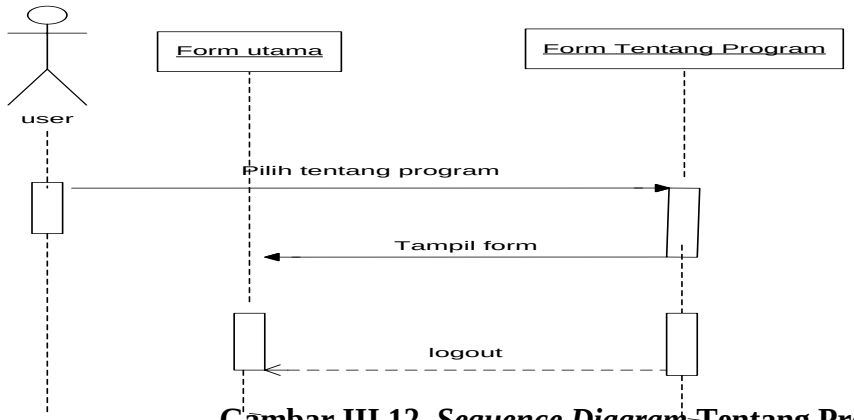
Sequence diagram data karyawan menggambarkan interaksi antara objek pada proses data karyawan dan mengenkrip, serta dekrip data karyawan . *Sequence diagram* data karyawan ditunjukkan pada gambar dibawah ini:



Gambar III.11. *Sequence Diagram* Data Karyawan

4. *Sequence Diagram* Tentang Program

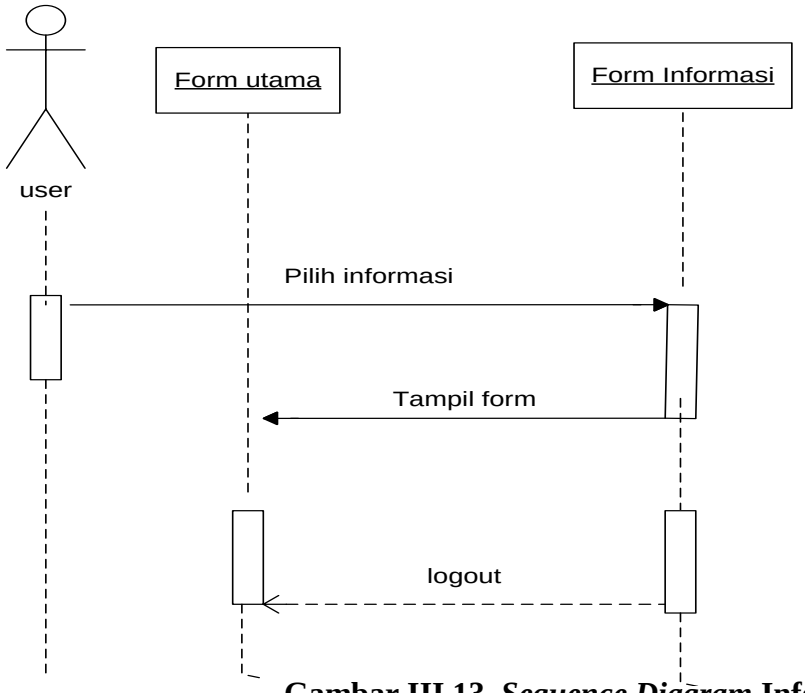
Sequence diagram tentang program menggambarkan interaksi antara user pada form tentang program. *Sequence diagram* tentang program ditunjukkan pada gambar dibawah ini:



Gambar III.12. *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:

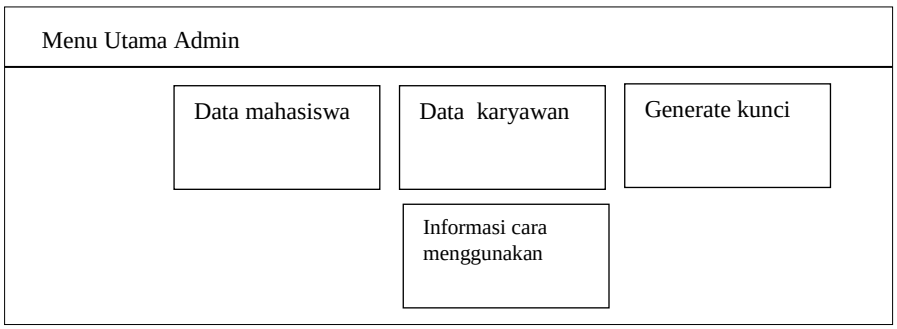


Gambar III.13. *Sequence Diagram* Informasi

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 Metode Elgamal. Rancangan Antar muka beranda ditunjukkan pada gambar III.14 berikut ini :



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

Adapun pada perancangan *form* Layar *Menu* Utama dapat dilihat pada uraian berikut:

Pada *data mahasiswa* menggunakan *label*

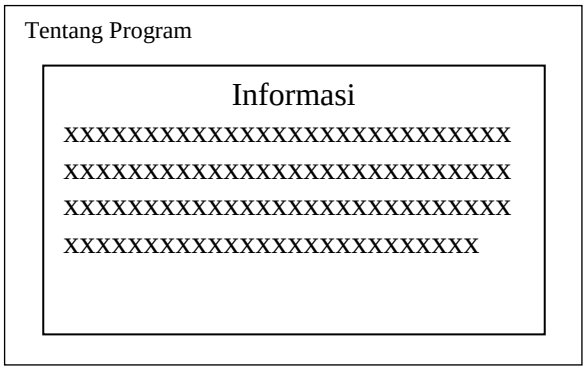
Pada *diagnose* menerima *file* menggunakan *label*

Pada informasi Cara penggunaan menggunakan *label*

Pada Tentang programi menggunakan *label*

2. Form Tentang Program

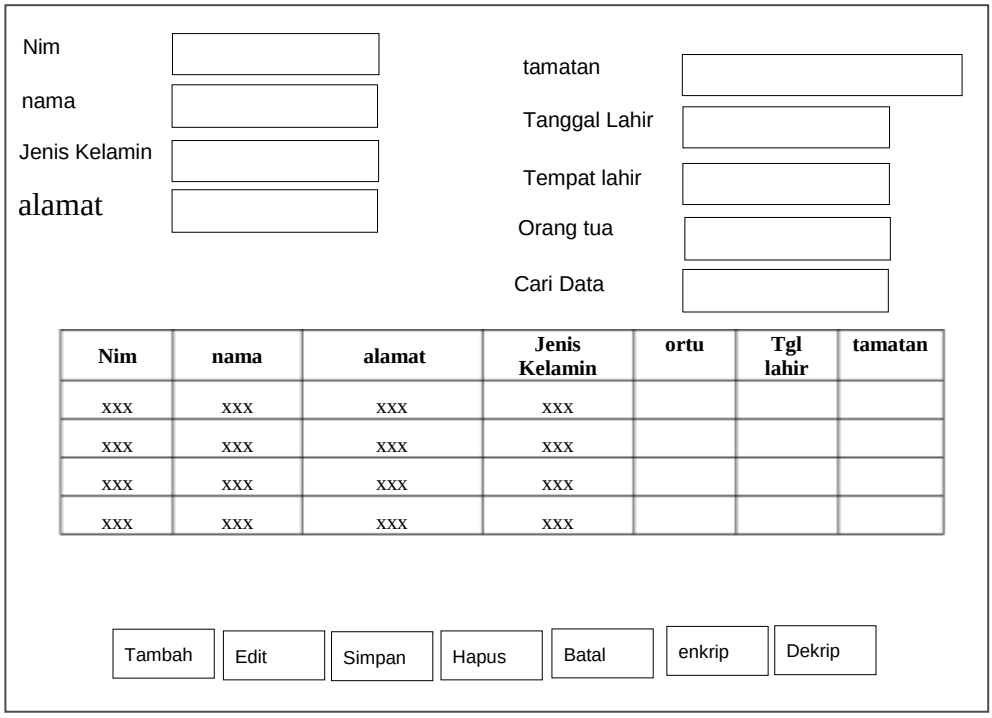
Form ini dirancang untuk menjelaskan untuk apa aplikasi Keamanan Database Menggunakan Metode Elgamal ini di buat:



Gambar III.15. Rancangan antar muka *form* Tentang Program

3. Form mahasiswa

Form ini dirancang untuk menambah menghapus dan merubah data mahasiswa selanjutnya untuk dilakukan enkripsi dan dekripsi.



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

Adapun pada perancangan *form* Layar *mahasiswa* dapat dilihat pada uraian berikut:

Pada nama, alamat, jenis kelamin menggunakan *label*

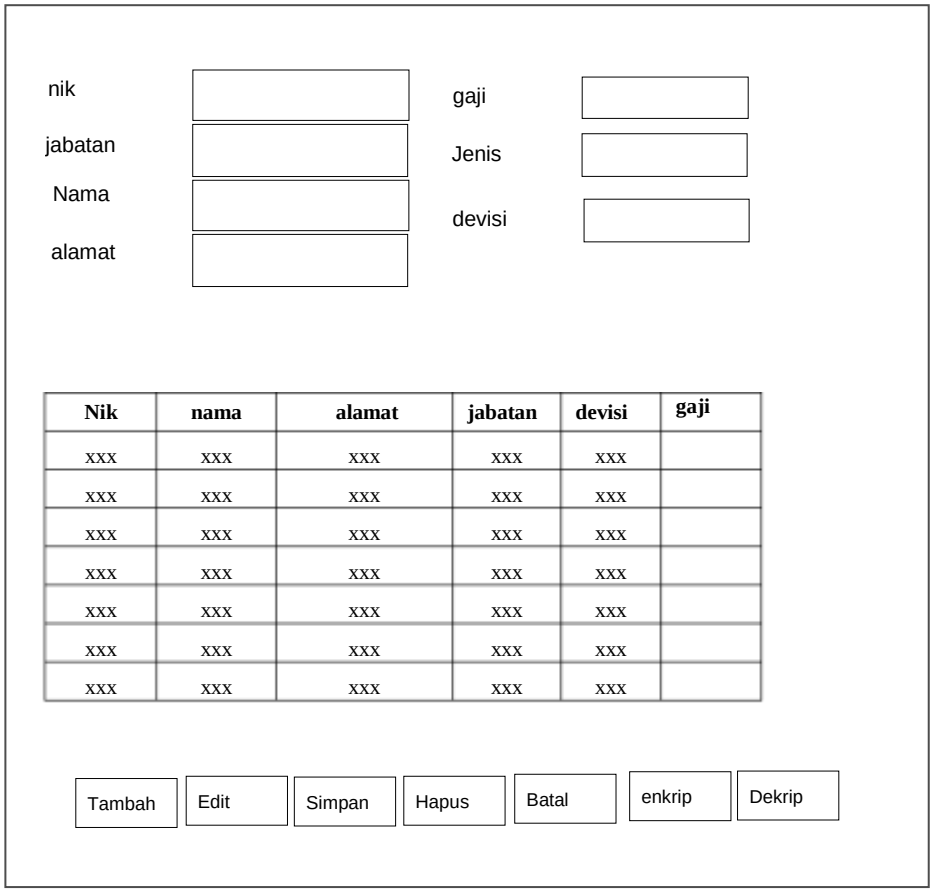
Pada tanggal lahir nim dan ortu menggunakan *label*

Pada tombol simpan, edit, tambah, hapus, batal, menggunakan *Button*

Pada tombol enkrip dan dekrip menggunakan *Button*

4. Form karyawan

Rancangan form karyawan berguna untuk menambah menghapus dan merubah data karyawan selanjutnya untuk dilakukan enkripsi dan dekripsi.



Gambar III.17. Rancangan antar muka *form* Karyawan

Adapun pada perancangan *form* Layar *karyawan* dapat dilihat pada uraian berikut:

Pada nik, alamat menggunakan *label*

Pada nama, gaji menggunakan *label*

Pada jabatan menggunakan *label*

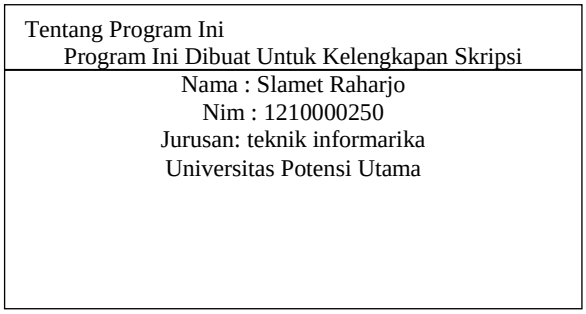
Pada tombol simpan, edit, tambah, hapus, batal, menggunakan *Button*

Pada devisi menggunakan *label*

Pada tombol enkrip dan dekrip menggunakan *Button*

5. Form profil pembuat

Form ini dirancang untuk menjelaskan penulis dan pembuat Aplikasi Keamanan Database Menggunakan Metode Elgamal.



Gambar III.18. Rancangan antar muka *form* Profil Pembuat

Adapun pada perancangan *form* Layar Pembuat dapat dilihat pada uraian berikut:

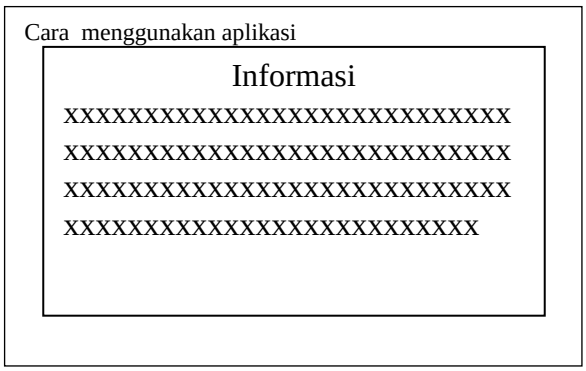
Pada nama menggunakan *label*

Pada nim menggunakan *label*

Pada jurusan menggunakan *label*

6. Form Informasi

Form ini dirancang untuk menjelaskan cara penggunaan aplikasi Keamanan Database Menggunakan Metode Elgamal:



Gambar III.19. Rancangan antar muka *form* Informasi

Adapun pada perancangan *form* Layar *informasi* adalah: Pada informasi menggunakan *label*

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. Karyawan={(**nik**+nama+alamat+jeniskelamin+gaji+
   tempatlahir+jabatan+devisi+namaortu+tamatan)}
3. Mahasiswa ={(**nim**+nama+alamat+jeniskelamin+tempatlahir +namaortu+tamatan)}

III.5.2. Desain Tabel/ File

Pada sistem ini, digunakan *database SQL Server* dengan nama dbelgamal menggunakan 3 tabel, yaitu tabel user, tabel mahasiswa dan tabel karyawan . Adapun struktur data dari tabel-tabel tersebut adalah sebagai berikut :

III.5.2.1. Struktur Tabel mahasiswa

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

Tabel III.1. Tabel Mahasiswa

No	Field Name	Type	Width	Keterangan
1	nim	Nchar	nim	Idmahasiswa
2	Nama	Nchar	10	Nama
3	Alamat	Nchar	25	Tempat
4	Tanggal	Date	date	Tanggal
5	jeniskelamin	Nchar	10	Alamat
6	Nama ortu	Nchar	25	Nama ortu
7	Tamatan	Nchar	25	Tamatan
8	Tempat lahir	Ncar	100	Tempat lahir

III.5.2.2. Struktur Tabel admin

Tabel admin digunakan untuk menyimpan *record* data user dengan properti atribut *id admin, 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 karyawan

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

Tabel III.3. Tabel Karyawan

No	Field Name	Type	Width	Keterangan
1	nik	Nchar	10	Nik
2	Nama	Nchar	10	Nama
3	Alamat	Nchar	25	Tempat
4	Tanggal	Date	date	Tanggal
5	jeniskelamin	Nchar	10	Alamat
6	Nama ortu	Nchar	25	Nama ortu
7	Tamatan	Nchar	25	Tamatan
8	Tempat lahir	Nchar	100	Tempat lahir
9	Gaji	Int	-	Gaji
10	jabatan	Nchar	25	jabatan
11	Devisi	Nchar	25	Devisi