BAB III

ANALISIS DAN PERANCANGAN SISTEM

3. 1 Analisis Sistem

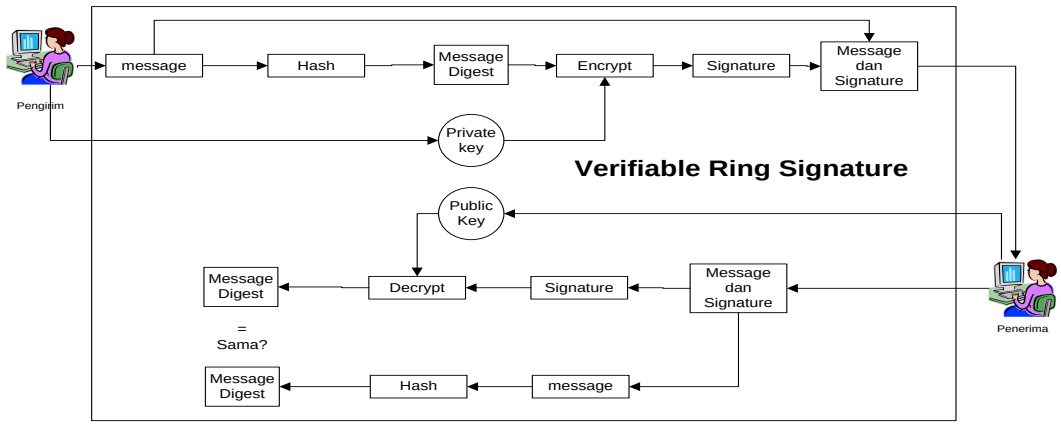
Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennnya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Dalam proses pembuatan suatu sistem mutlak dilakukan penelitian dan penganalisaan tentang sistem yang akan dibangun, berikut adalah beberapa analisis yang dilakukan untuk membangun Aplikasi pembuatan tanda tangan digital.

3. 1. 1. Deskripsi Umum Sistem

Proses sistem dimana pada perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean kedalam suatu bahasa pemrograman. Dalam perancangan sistem tidak lepas dari hasil analisis, karena dari hasil analisis sistem baru dapat dibuat suatu perancangan sistem.

Desain umum yang akan diaplikasikan bertujuan untuk memberikan gambaran secara umum kepada penggunaan tentang sistem yang akan dibangun.



Gambar 3. 1. Gambaran Umum Sistem

Gambar 3.1 diatas menunjukan tentang gambaran umum sistem.

Pada pengirim :

1. pengirim harus mempunyai dua pasang kunci, terutama kunci private untuk memproses signature.
2. Pesan akan di hash yang akan menghasilkan message digest (value)
3. Hasil message digest tersebut akan dienkrip menggunakan private key dengan menggunakan metode verifiable ring signature. Hasil enkrip inilah yang dinamakan dengan *digital signature*
4. Kemudian *digital signature* dilekatkan dengan pesan (dengan cara menyambung),lalu keduanya dikirmkan melalui saluran komunikasi. Dalam hal ini kita katakana bahwa pesan sudah ditandatangani oleh pengirim.

Pada penerima :

1. *Signature* di otentifikasi untuk dibuktikan keaslian

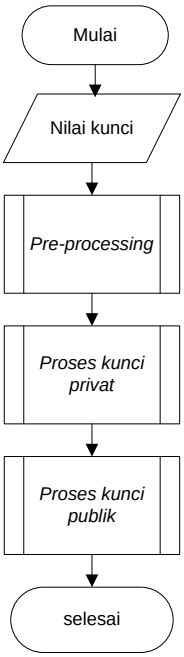
2. signature di dekrip menggunakan kunci public pengirim, menghasilkan message digest semula
3. Pesan di hashing kembali untuk menghasilkan message digest
4. Selanjutnya message digest akan dibandingkan Jika MD'=MD, berarti otentifikasi valid dan signature yang diterima berasal dari pengirim yang benar

3. 1. 1. 1 Analisis Verifiable Ring Signature

Verifiable Ring Signature ini adalah metode yang akan digunakan dalam pembuatan perangkat lunak. *Verifiable Ring Signature* dapat digunakan oleh salah satu anggota kelompok pengguna yang berada dalam sebuah lingkar kerahasiaan. Untuk menandatagani sebuah dokumen, dengan menggunakan metode ini diperlukan kunci privat dari individu yang mengirim dan semua kunci publik dari anggota lingkar kerahasiaan tersebut. Salah satu karakteristik unik dari metode ini adalah pihak penerima mengetahui bahwa pesan tersebut dikirim oleh sekumpulan orang yang tergabung dalam lingkar kerahasiaan tersebut, dan bisa mengetahui siapakah individu yang mengirimkan pesan tersebut.

3. 1. 1. 2 Analisis Pembentukan Kunci

Dalam pembentukan kunci ini, menjadi 3 tahap, tahap pertama adalah *pre-pre*-processing , tahap kedua adalah tahap pemrosesan kunci privat, tahap ketiga adalah pemrosesan kunci public. Pembentukan kunci pada aplikasi ini mengikuti alur sesuai yang di tunjukan pada gambar 3.2



Gambar 3. 2. Alur Pembentukan Kunci

3.1.1.2.1. *Pre-Processing*

Pembentukan kunci yang akan dibuat adalah kunci private (untuk pengirim) dan kunci public (untuk penerima). Pada *pre-processing* ini akan menjelaskan 3 parameter yang dibutuhkan pada saat pembuatan kunci privat dan kunci publik, parameter tersebut adalah *p*, *q*, dan *a*. parameter yang pertama adalah *p, yaitu* suatu nilai angka yang harus prima dimana nilai *p* ini sudah ditentukan, alasan sudah ditentukan karena nilai p ini tidak ada perhitungan khusus. Parameter yang kedua adalah *q*, yaitu suatu nilai angka yang harus prima dimana syarat nilai *q* adalah *(p-1)mod q=0.* Parameter yang ketiga adala *a*, yaitu suatu nilai dimana syarat nilai *a* adalah $a^q(mod\ q) = 1$. Alasan nilai *p*, *q*, dan *a* ini harus dihitung terlebih dahulu karena pada proses perhitungan kunci private

dan kunci public akan memakai nilai-nilai tersebut. Prosedur perhitungannya adalah sebagai berikut :

1. Penentuan nilai *p*

   Proses nilai *p* tidak ada perhitungan khusus tetapi memiliki syarat bahwa nilai *p* harus bilangan prima. Contoh Inisialisasi nilai *p* adalah *23.* Mengapa 23, karena 23 adalah bilangan prima atau bisa juga di inisialisasi dengan nilai bilangan prima lainnya.

2. Penentuan nilai *q*

   Proses nilai *q* tidak ada perhitungan khusus tetapi memiliki beberapa syarat yang harus terpenuhi. Syarat pertama adalah *q* harus bilangan prima dan yang kedua jika nilai *q* memakai rumus :

   $$(p\text{-}1) \bmod q = 0 \qquad (3.1)$$

   Contoh perhitungan :

   Diinisialisasikan

   p = 23

   q = 11

   = (p – 1) mod q

   = (23 – 1) mod 11

   = 22 mod 11

   = 0

Berdasarkan contoh perhitungan di atas nilai q yang di dapat adalah 11. Karena nilai tersebut telah memenuhi syarat yaitu bilangan prima dan

berdasarkan rumus (3.1) hasilnya bernilai 0. Nilai q bisa saja menggunakan bilangan lainnya asalkan memenuhi dua syarat di atas.

3. Penentuan nilai *a*

   Proses nilai *a* tidak ada perhitungan khusus tetapi memiliki syarat yang harus terpenuhi. Yaitu jika memakai rumus :

$$a^q \ mod \ p = 1 \qquad (3.2)$$

   Contoh perhitungan :

   Diinisialisasikan

   p = 23

   q = 11

   a = 8

   = $a^q \ mod \ p$

   = $8^{11} mod \ 23$

   = 8589934592 mod 23

   = 1

Berdasarkan contoh perhitungan di atas nilai *a* yang di dapat adalah 8. Karena nilai tersebut telah memenuhi syarat berdasarkan rumus (3.2). Nilai *a* bisa menggunakan bilangan lainnya asalkan memenuhi syarat tersebut.

3.1.1.2.2. Proses Pembentukan Kunci Privat

Pada proses pembentukan kunci privat ini menggunakan rumus :

$$s < q \qquad (3.3)$$

Algoritma prosedur nilai s

```
Procedure fungsi nilai s
Deklarasi
  s, q : integer
algoritma
q ← q
s ← random
while (true)
{
  If(s<q) then
     Return true
   Else
     S← random
}
```

Berdasarkan algoritma di atas pembentukan kunci privat ini adalah menggunakan bilangan secara acak (random) dimana bilangan acak ini memiliki syarat harus lebih kecil dari nilai q. Bila nilai q = 11 maka nilai s berada di antara nilai 1 – 10. Contoh s adalah 8.

3.1.1.2.3. Proses Pembentukan Kunci Publik

Proses pembentukan kunci publik ini adalah mencari nilai λ dengan menggunakan rumus :

$$\lambda = (a^{-1} \bmod p)^s \bmod p \qquad (3.4)$$

Algoritma procedure kunci public λ

```
Procedure fungsi hitung lamda
Deklarasi
```

```
  Lamdha, a, p, s : integer
Algoritma
  a ← a
  p ← p
  s ← s
lamdha ← (a^-1 mod p)^s mod p
```

Contoh perhitungan :

a = 8

p = 23

s = 8

λ =$(a^{-1} \bmod p)^s \bmod p$

$=(8^{-1} mod\ 23)^8 mod\ 23$

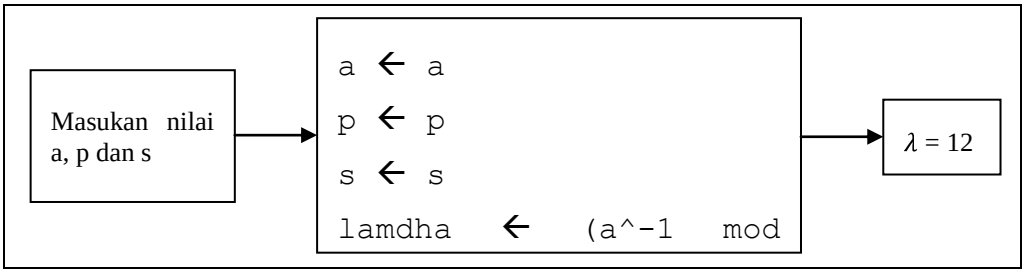
$= (7 \bmod 23)^8 \bmod 23$

$= 7^8 \bmod 23$

$= 5764801 \bmod 23$

$= 12$

Alur masukan nilai a, p, dan s, diproses dengan rumus kunci public, lalu menghasilkan nilai λ.



3. 1. 1. 3 **Pembentukan *Message Digest***

Message digest adalah suatu besaran (value) yang berasal dari suatu data atau pesan yang memiliki sifat yang unik yang menandai bahwa pesan tersebut mempunyai suatu besaran tertentu.

Perhitungan *message digest* menggunakan hash SHA-256 pada pesan atau file data yang diberikan sebagai input.

Beberapa hal yang dilakukan dalam pengisian pesan adalah sebagai berikut :

a. Panjang dari pesan,M adalah k bits dimana panjang $k < 2^{64}$. Tambahkan bit "1" pada akhir pesan. Misalkan pesan yang asli adalah "01010000" maka setelah diisi menjadi "010100001".
b. Tambahkan bit "0", angka bit "0" tergantung dari panjang pesan. Misalnya :Pesan asli yang merupakan bit string : **abcde**

   01100001 01100010 01100011 01100100 01100101.

Setelah langkah (a) dilakukan

01100001 01100010 01100011 01100100 0110010 1.

Panjang k = 40 dan angka bit di atas adalah 41dan 407 ditambah bit "0" (448 – (40+1) = 407). Kemudian diubah dalam hex:

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

c. Untuk memperoleh 2 kata dari k,angka bit dalam pesan asli yaitu jika k < 2^{32} maka kata pertama adalah semua bit ”0”. Maka gambaran dari 2 kata dari k = 40 dalam hex adalah 00000000 00000028.

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000028

SHA-256 menggunakan urutan fungsi logika yang dilambangkan dengan f_0, f_1,…, f_{79}. Untuk masing-masing f_t, dimana $0 \leq t < 79$ akan menghasilkan output sebanyak 32 bit.

Fungsinya adalah sebagai berikut:

$$f_t(B, C, D) = \begin{cases} (B \wedge C) \vee (\neg B \wedge D) & 0 \leq t \leq 19 \\ B \oplus C \oplus D & 20 \leq t \leq 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & 40 \leq t \leq 59 \\ B \oplus C \oplus D & 60 \leq t \leq 79 \end{cases}$$

⊕ = fungsi XOR

Konstanta kata yang digunakan pada SHA-1 yang disimbolkan secara berurutan dari K(0), K(1),…, K(79) dalam bentuk *hex* adalah sebagai berikut :

$$K_t = \begin{cases} 5A827999 & 0 \le t \le 19 \\ 6ED9EBA1 & 20 \le t \le 39 \\ 8F1BBCDC & 40 \le t \le 59 \\ CA62C1D6 & 60 \le t \le 79 \end{cases}$$

Algoritma SHA-256 dapat diringkas sebagai berikut:

a. Penghitungan menggunakan dua *buffer* dimana masing-masing *buffer* terdiri dari lima sebesar 32 bit kata dan urutan 80 juga sebesar 32 bit kata. Lima kata pertama pada *buffer* kata diberi nama A, B, C, D, E sedangkan lima kata kedua diberi nama H_0, H_1, H_2, H_3, dan H_4. Kemudian pada 80 kata yang berurutan diberi nama W_0, W_1, …, W_{79} dan pada penghitungan ini juga memakai sebuah variabel sementara, TEMP.

b. Lakukan pengisian pesan, M dan kemudian parsingkan pesan tersebut ke dalam N 512 bit blok pesan, $M^{(1)}$, $M^{(2)}$, …, $M^{(n)}$. Caranya : 32 bit pertama dari blok pesan ditunjukkan ke $M_0^{(i)}$, lalu 32 bit berikutnya adalah $M_1^{(i)}$ dan selanjutnya berlaku hingga $M_{15}^{(i)}$.

c. Inisialisasi Nilai *Hash* (dalam bentuk hex) :

   H_0 = 67452301 H_3 = 10325476

   H_1 = EFCDAB89 H_4 = C3D2E1F0

   H_2 = 98BADCFE

d. Lakukan proses M_1, M_2, …, M_n dengan cara membagi M_i ke dalam 16 kata W_0, W_1, …, W_{15} dimana W_0 merupakan *left most*.

e. Hitung : For t = 16 to 79

$$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$$

f. Inisialisasi 5 variabel A, B, C, D, dan E dengan nilai *Hash* :

$$A = H_0 ; B = H_1; C = H_2; D = H_3; E = H_4.$$

g. Hitung : For t = 0 to 79

$$TEMP = S^5(A) + f_t(B,C,D) + E + Wt + Kt$$

$$E = D; D = C; C = S^{30}(B); B = A; A = TEMP.$$

h. Hitung Nilai *Hash* :

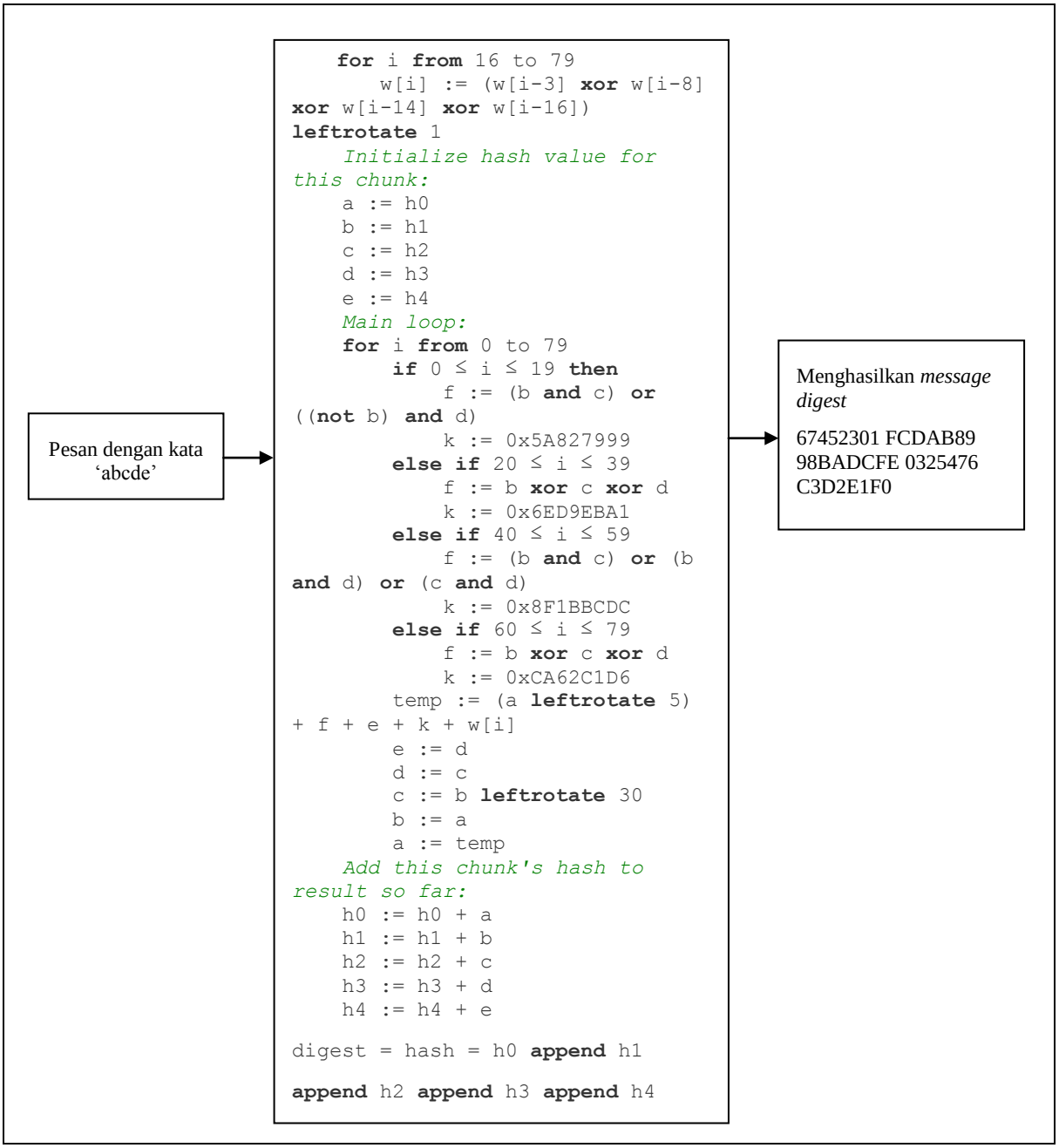
$H_0 = H_0 + A ; H_1 = H_1 + B ;$

$H_2 = H_2 + C ; H_3 = H_3 + D ;$

$H_4 = H_4 + E.$

Hasil dari *message digest*, M adalah : H_0 H_1 H_2 H_3 H_4**,** yaitu 67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0.

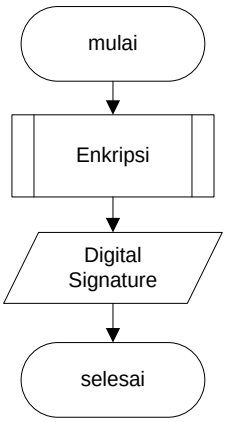
Alur masukan pesan lalu di proses dengan fungsi hash SHA-256 untuk menghasilkan *message digest.*



3. 1. 1. 4 Analisis Pembentukan Digital Signature

Tanda tangan digital (digital signature) merupakan protocol yang dijalankan untuk dapat memverifikasi keaslian dan keutuhan dari pesan yang akan dikirimkan dalam suatu saluran komunikasi.

Analisis pembentukan digital signature ini dibagi menjadi 2 tahap, tahap pertama adalah tahap enkrip pesan dan tahap kedua adalah tahap hasil *digital signature*. Pembentukan *digital signature* pada aplikasi ini mengikuti alur sesuai yang ditunjukan pada gambar 3.8.



Gambar 3. 3. Alur Pembentukan *Digital Signature*

3. 1. 1. 4. 1 Proses Enkripsi Pesan

Proses ini di bagi menjadi

Proses ini adalah proses dimana hasil nilai hash yaitu 67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0. Akan di enkripsi untuk menghasilkan sebuah *digital signature.* Prosedurnya adalah sebagai berikut :

1. Lakukan perhitungan nilai x dengan menggunakan rumus :

$$x = a^r mod\ p \text{ dimana r < q} \qquad (3.5)$$

Algoritma prosedur nilai x

```
Procedure fungsi nilai x
Deklarasi
  x, p, a, r, q : integer
algoritma
q ← q
a ← a
p ← p
r ← random
while (true)
{
  If ( r < q ) then
      x ← (a ^ r)  mod p
   Else
      S← random
}
```

Contoh perhitungan :

x = suatu nilai angka

a = 8

p = 23

r = 5

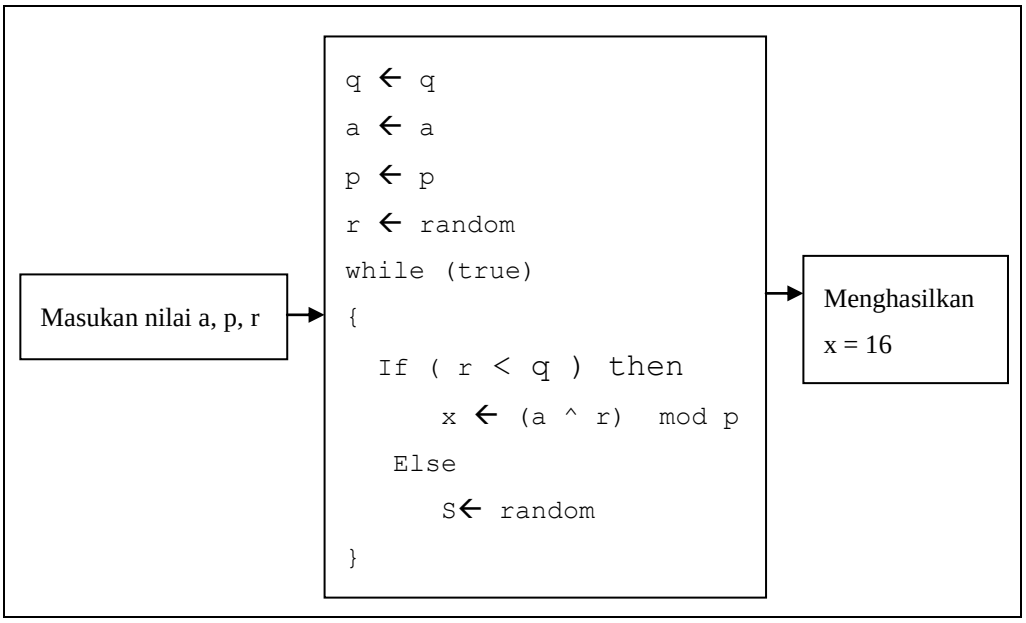
$x = a^r mod\ p$

$= 8^5$ mod 23

= 32768 mod 23

= 16

Alur masukan mulai dari nilai a, p, dan r lalu akan diproses dengan rumus (3.5) agar menghasilkan nilai x



Berdasarkan rumus (3.5) dijelaskan bahwa nilai r < q, nilai r adalah nilai bilangan secara acak dimana memiliki syarat nilai r lebih kecil dari q. Karena nilai q = 11 maka nilai r diantara 1-10. contoh diinisialisasikan r adalah 5.

2. Lakukan Perhitungan nilai h dengan menggunakan rumus :

$$h||x \qquad (3.6)$$

Contoh perhitungan :

h = 67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0 (hexadecimal) di convert ke decimal menjadi 415047364035088646

x = 16

= h||x

= 415047364035088646 || 16

= 415047364035088661

3. Lakukan Perhitungan nilai y dengan menggunakan rumus :

$$y = (r + (s \times h)) \pmod p \qquad (3.7)$$

Algoritma *procedure* nilai y

```
Procedure fungsi y
Deklarasi
   y,r,s,p : integter
   h       : longint
   sig,h'  : bit
Algoritma
  r ← r
  s ← s
  p ← p
     y = (r + (s x h)) (mod p)
  sig = h' append y
```

Contoh perhitungan :

y = suatu nilai angka

r = 5

s = 8

h = 415047364035088661

p = 23

y = (r + (s x h)) (mod p)

= (5 + (8 x 415047364035088661)) (mod p)

= (5+3320378912280709288) (mod 23)

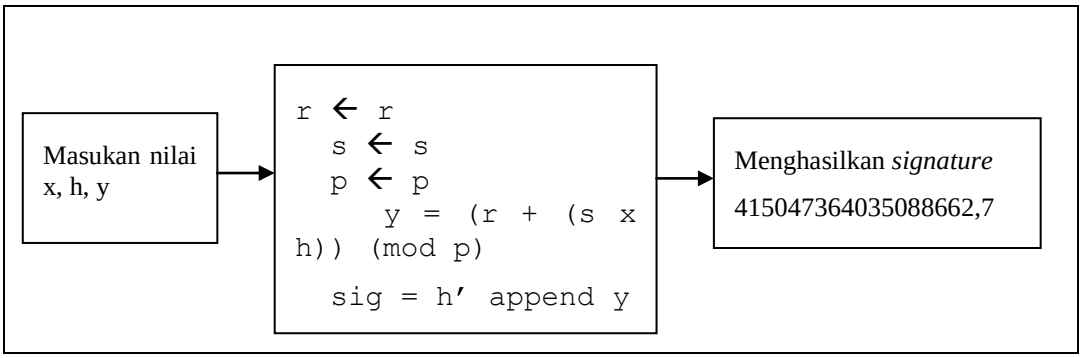
= 3320378912280709293 mod 23

= 7

Hasil perhitungan menghasilkan *signature* h' adalah h ; y adalah :

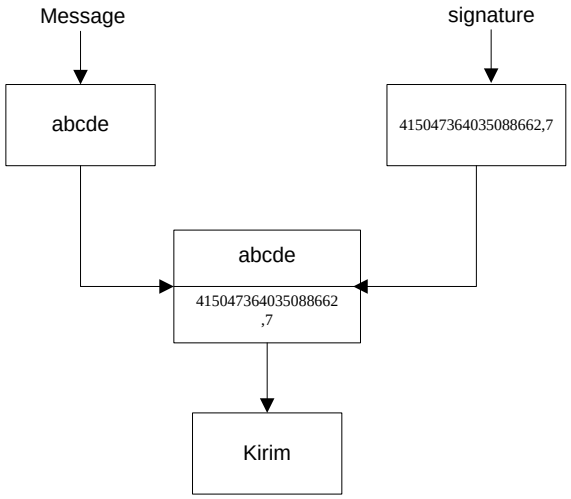
415047364035088662,7

Alur masukan nilai x, h, y lalu akan di enkipsi menggunakan *verifiable ring* agar menghasilkan *signature*.



3. 1. 1. 5 Proses Melekatkan Digital Signature dan Message

Proses ini dimana signature yang sudah dibuat akan dilekatkan pada pesan kemudian akan dikirim. Alur prosesnya adalah sebagai berikut



Gambar 3. 4. Pelekatan Message dan Signature

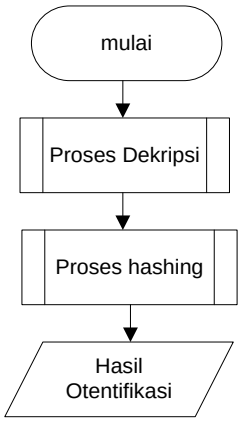
Dari perhitungan yang sudah dilakukan, didapat pesan asli yaitu abcde dan hasil signaturenya adalah 415047364035088662,7. Message dan signature

dilekatkan lalu dikirimkan dan di penerima akan di autentufikasi untuk menegetahui pesan dan signature itu valid atau tidak.

3. 1. 1. 6 Analisis Proses Otentifikasi

Proses otentifikasi adalah proses memastikan keaslian si pengirim. Artinya apakah yang mengirimkan itu asli dari orang yang bersangkutan. Berikut adalah alur proses otentifikasi.

Dalam analisi otentifikasi ini dibagi menjadi 3 tahap, tahap pertama adalah proses dekripisi signature, tahap kedua adalah proses hashing pesan, lalu hasil dekrip dan hashing tersebut dibandingkan apakah nilainya sama. Jika sama maka otentifikasi valid. Proses otentifikasi tersebut mengikuti alur yang ditunjukan pada gambar 3.5.



Gambar 3. 5. Alur Proses Otentifikasi

3. 1. 1. 6. 1 Proses Dekripsi

Proses dekripsi ini adalah pengembalian signature menjadi file text kembali. Pada proses ini signature akan didekrip hasil prosesnya akan menghasilkan nilai message digest. Beberapa prosesnya adalah sebagai berikut :

a. Lakukan perhitungan z menggunakan rumus :

$$z = (a \,.\, \lambda) mod\ p \qquad (3.8)$$

Algoritma *procedure* nilai z dan h'

```
Procedure fungsi nilai z dan h'
Deklarasi
  z,a, λ,p    : integer
algoritma
  a ← a
  lamda ← lamda
  p ← p
    z = (a x lamda) mod p
   hash = h' or z
```

Contoh perhitungan :

a = 8

λ = 12

p = 23

z = suatu nilai

z = $a \,.\, \lambda (mod\ p)$

= 8 .12(mod 23)

= 96 mod 23

= 4

b. Lakukan perhitungan h’ dengan rumus

$$h' = h||z \quad (3.9)$$

Contoh perhitungan :

h = 415047364035088662

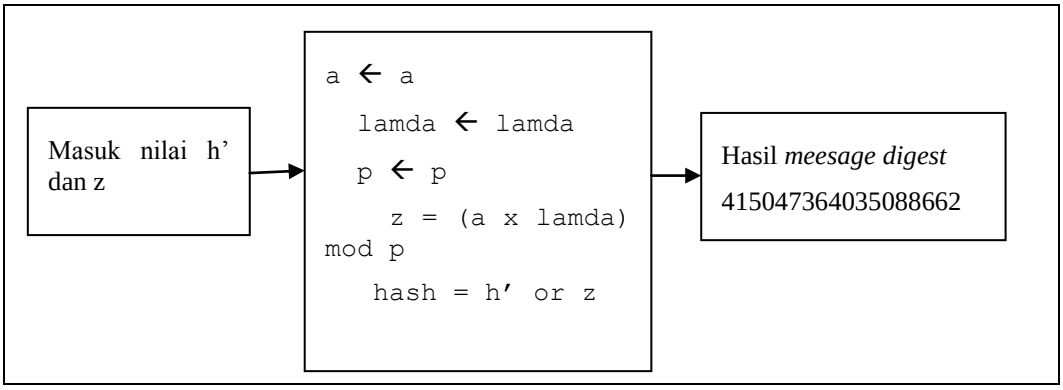
z = 4

h’ = h||z

= 415047364035088646||4

= 415047364035088662

Alur masukan h’ dan z lalu akan diproses pendekripsian dengan metode verifiable ring agar menghasilkan message digest.



3. 1. 1. 6. 2 Proses Otentifikasi

Proses ini hanya membandingkan hasil message digest dari pesan yang diterima dan hasil dekrip dari signature yang telah diterima apakah memiliki nilai yang sama atau tidak. Proses hashing pada pesan yang diterima sama dengan

proses hashing yang sudah di jelaskan pada sub bab sebelumnya. Proses pembandingannya adalah sebagai berikut :

- jika h mod q = h’ mod q (valid)

- Jika h mod q ≠ h’ mod q (invalid)

Algoritma *procedure* otentifikasi

```
Procedure otentifikasi
Deklarasi
   Valid    : Boolean
   h', h      : longint
Algoritma
  if h' = h then
     valid ← true
  else
     valid ← false
```

Contoh :

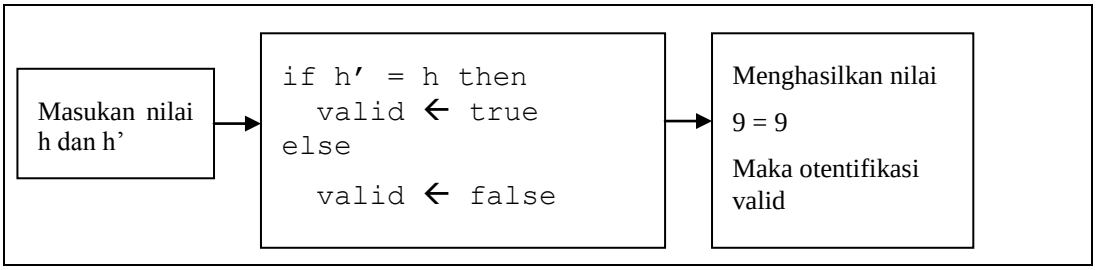
H mod q = h’ mod q

415047364035088662 mod 11 = 415047364035088662 mod 11

9 = 9

Maka autentifikasi valid

Alur masukan h dan h’ diproses dengan rumus pembanding verifiable ring, lalu akan diproses agar menghasilkan nilai untuk mengetahui otentifikasi valid atau tidak.



3. 2 Analisi kebutuhan Non Fungsional

Berikut ini adalah kebutuhan non fungsional yang dibutuhkan oleh sistem :

1. komputer berspesifikasi minimal Pentium 3 atau yang setara dengan jumlah Ram 512 MB
2. Maksimal jumlah memori yang digunakan oleh perngkat lunak tidak melebihi 50 MB

3. 2. 1. Atribut Kualitas Perangkat Lunak

Berikut ini menjelaskan tentang kualitas perangkat lunak yang dibangun :

1. Perangkat lunak dapat dijalankan pada komputer dengan spesifikasi rendah
2. Perangkat lunak dapat digunakan untuk membuat tanda tanggal digital
3. Bahasa pemrograman yang digunakan adalah bahasa yang kecil dan mudah memungkinkan untuk dikembangkan
4. Perangkat lunak dibuat dengan bahasa pemrograman yang dapat bekerja di berbagai arsitektur komputer.

3. 2. 2. Perancangan Lingkungan Implementasi

Spesifikasi ini meliputi Sistem Operasi, Development Tools dan Bahasa pemrograman yang dipakai. Berikut ini adalah spesifikasi yang digunakan :

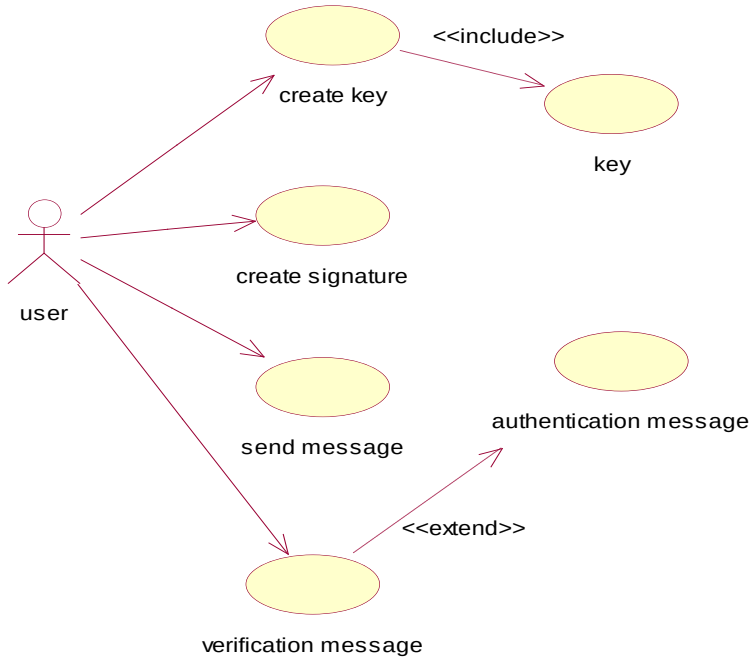
Tabel 3. 1. Perancangan Sistem

Digital Signature	Spesifikasi
Sistem Operasi	Windows XP
Development Tools	NetBeans IDE 6.8
Bahasa Pemrograman	Java

3. 3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional dianalisis dengan memodelkan sistem. Pemodelan yang digunakan untuk memodelkan perangkat lunak ini adalah pemodelan berorientasi objek. Perangkat lunak ini dimodelkan menggunakan UML (*Unified Modeling Language*). *Tools* yang digunakan adalah Use Case Diagram, Sequence Diagram dan Activity Diagram yang dibuat menggunakan *Rational Rose* sebagai perangkat lunak yang digunakan.

Berikut ini adalah usecase diagram dari perangkat lunak tanda tangan digital:



Gambar 3. 6. Usecase Diagram

1. **Skenario**

Berikut ini adalah scenario *create key* :

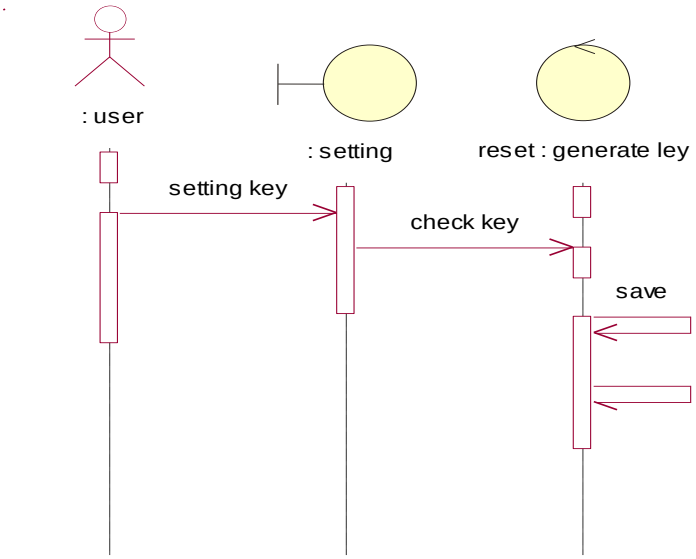
Tabel 3. 2 Skenario Proses *Create Key*

Identifikasi	
Nomor	1
Nama	Create key

Tujuan	Buat kunci
Deskripsi	Pengaturan kunci
Aktor	User, sistem
Skenario Utama	
Kondisi awal	*Pada form setting*

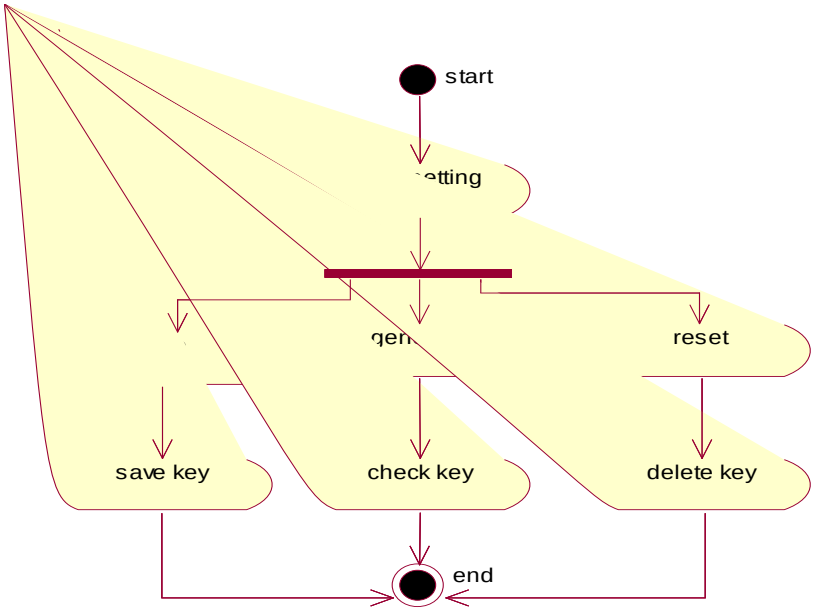
Aksi Aktor	**Reaksi Sistem**
1. Pilih setting	2. muncul form setting
3. pilih generate	4. menampilkan kunci

2. Dinamika Objek (*sequence Diagram*)



Gambar 3. 7. *Sequence Diagram Craete Key*

3. *Activity Diagram*



Gambar 3. 8. *Activity Diagram Create Key*

3. 3. 1. Proses Create Signature

1. Skenario

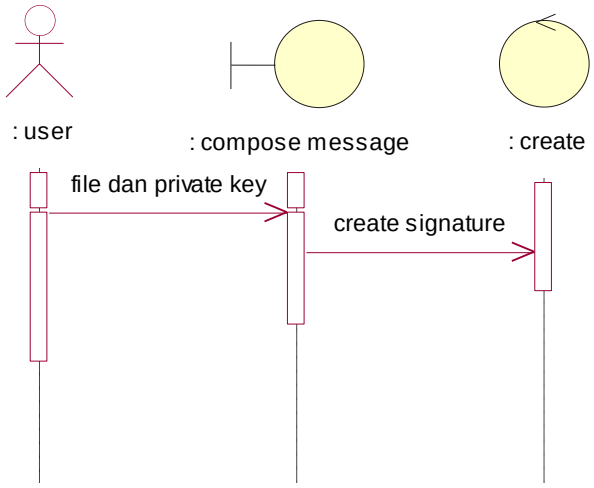
Berikut ini adalah skenario proses *create signature*

Tabel 3. 3 Skenario *Create Signature*

Identifikasi	
Nomor	2
Nama	*Create Signature*
Tujuan	Membuat signature
Deskripsi	File akan dibuat signature

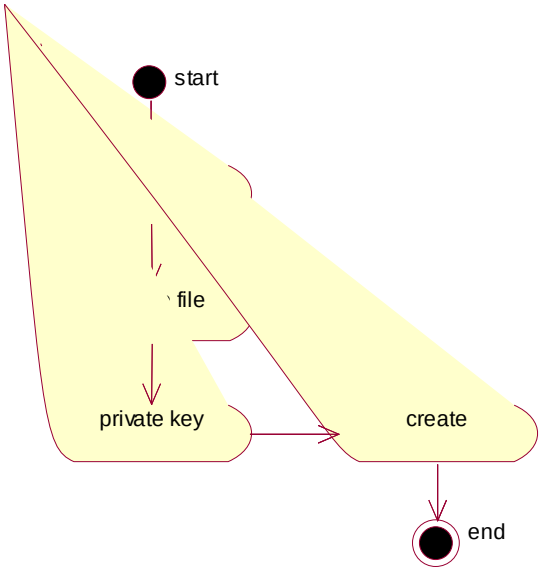
Aktor	User, sistem
Skenario Utama	
Kondisi awal	Masuk ke form compose message
Aksi Aktor	**Reaksi Sistem**
1. Input file dan private key	2. Buat signature
3. Pilih send	4. Kirim pesan
Kondisi akhir	File terkirim

2. Dinamika Objek (*Sequence Diagram*)



Gambar 3. 9. Sequence Diagram *Create Signature*

3. *Activity Diagram*



Gambar 3. 7. *Activity Diagram Create Siganture*

3. 3. 2. Proses Send Message

1. Skenario

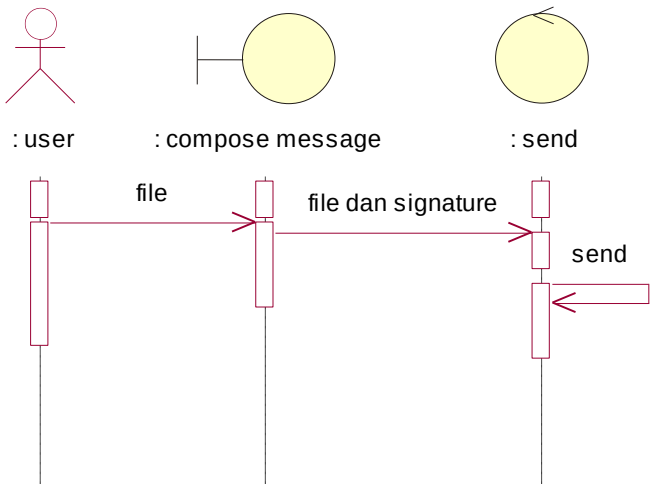
Berikut ini adalah skenario proses *send message* :

Tabel 3. 4 Skenario Send Message

Identifikasi	
Nomor	3
Nama	Send message
Tujuan	Mengirim file
Deskripsi	Pengiriman fie

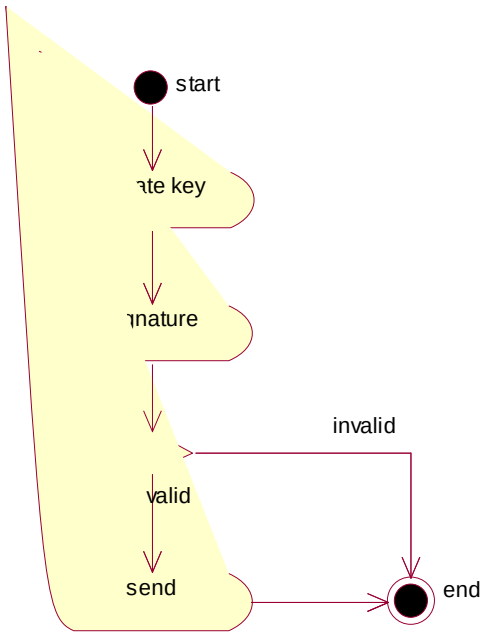
Aktor	User, sistem
Skenario Utama	
Kondisi awal	Masuk pada form compose message
Aksi Aktor	**Reaksi Sistem**
1. Pilih compose message	2. Menampilkan form compose message
3. Pilih file dan create signature	4. Kirim file dan signature
Kondisi akhir	File dan signature telah terkirim

2. Dinamika Objek (*Sequence Diagram*)



Gambar 3. 81. Sequence Diagram Send Message

3. Activity Diagram



Gambar 3.12 activity diagram proses send message

3. 3. 3. Proses Verification Message

1. Skenario

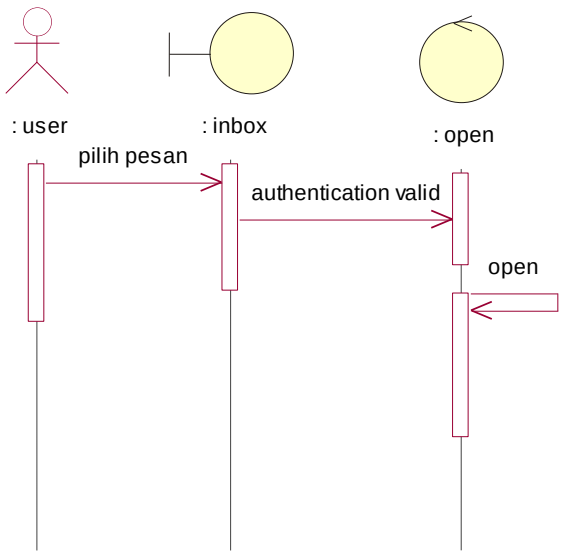
Berikut ini adalah skenario proses *Verification Message* :

Tabel 3. 5 Skenario Proses *Verifiation Message*

Identifikasi	
Nomor	4
Nama	Verification message
Tujuan	Check keaslian pesan
Deskripsi	File akan diverifikasi dan dicheck keasliannya

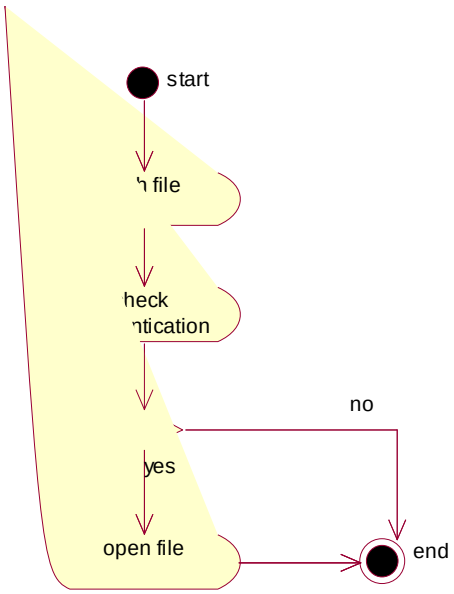
Aktor	User, sistem
Skenario Utama	
Kondisi awal	Masuk ke form inbox
Aksi Aktor	**Reaksi Sistem**
1. pilih file pesan masuk	2. akan memverifikasi pesan
2. jika berhasil	3. membuka file
Kondisi akhir	Pesan berhasil dibuka

2. Dinamika Objek (*Sequence Diagram*)



Gambar 3. 9. Sequence Diagram Proses *Verification Message*

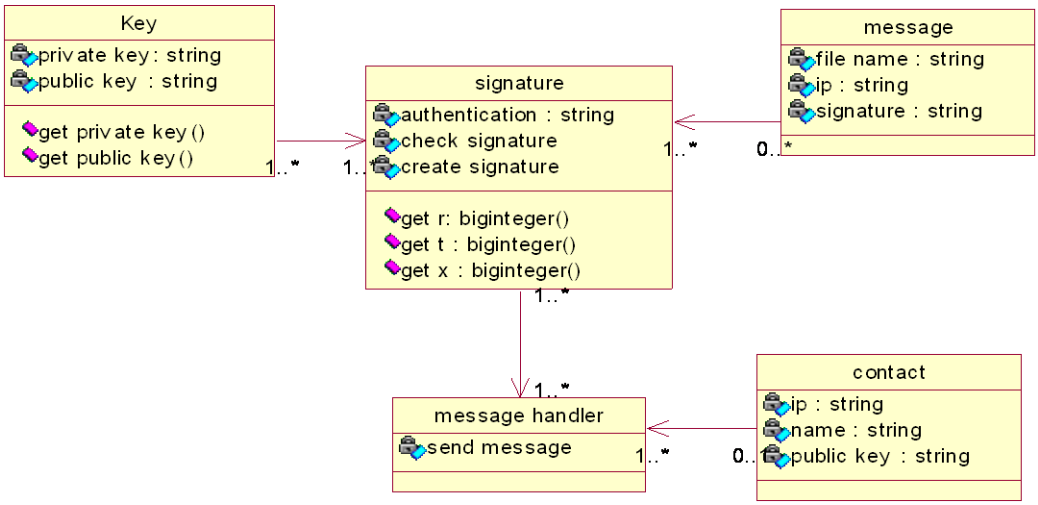
3. Activity Diagram



Gambar 3. 10. Activity Diagram Proses *Verification Message*

3. 3. 4. Deskripsi Kelas-kelas

3.3.4.1. Diagram Class



Gambar 3. 11. Diagram Class

Jenis kelas pada sistem dapat dilihat pada tabel berikut ini :

Tabel 3. 6. Tabel Keterangan Class Diagram

No	Kelas	Jenis	Keterangan
1	Key	Boundary	Kelas membuat kunci
2	Signature	Boundary	Kelas untuk membuat signature
3	Message	Control	Kelas pesan
4	Message hendler	Control	Kelas untuk mengirim dan menerima pesan
5	Contact	Interface	Kelas untuk menyimpan kontak

3. 4 Perancangan Sistem

Perancangan merupakan tahap lanjutan dari analisis sistem dimana pada perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean kedalam suatu bahasa pemograman. Dalam perancangan sistem tidak lepas dari hasil analisis, karena dari hasil analisis sistem baru dapat dibuat suatu perancangan sistem.

Desain umum yang akan diaplikasikan bertujuan untuk memberikan gambaran secara umum kepada penggunaan tentang sistem yang akan dibangun.

3. 4. 1. Batasan Perancangan

Batasan-batasan pada tahap perancangan perangkat lunak tanda tangan digital ini adalah :

1. Aplikasi ini hanya dapat membaca file teks tidak berbentuk gambar.

2. Perancangan yang dikembangkan meliputi perancangan procedural, arsitektur dan interface.

3. 4. 2. Perancangan Lingkungan Implementasi

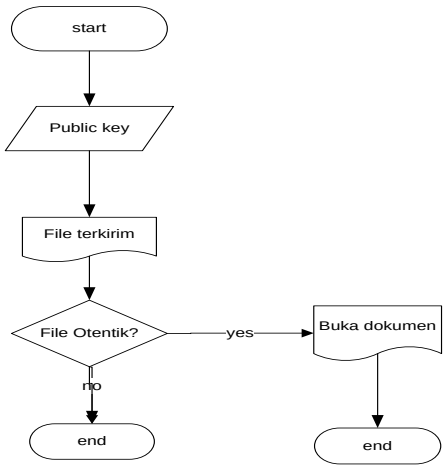
Spesifikasi ini meliputi Sistem Operasi, Development Tools dan Bahasa pemrograman yang dipakai. Berikut ini adalah spesifikasi yang digunakan :

Tabel 3. 7. Perancangan Sistem

Digital Signature	Spesifikasi
Sistem Operasi	Windows XP
Development Tools	NetBeans IDE 6.8
Bahasa Pemrograman	Java

3. 4. 3. Perancangan Proses Otentifikasi

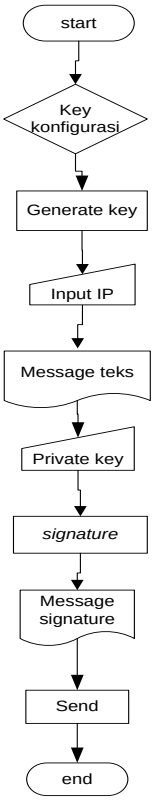
Perancangan Proses otentifikasi. Mulai dari memasukan kunci publik kemudian membuka file yang sudah terkirim kemudian proses verifikasi. Bila berhasil pesan dapat dibuka dan bila tidak pesan tidak dapat dibuka dan ada pemberitahuan bahwa pesan tersebut tidak otentik.



Gambar 3. 12. Perancangan Otentifikasi

3. 4. 4. Perancangan Struktur Aplikasi

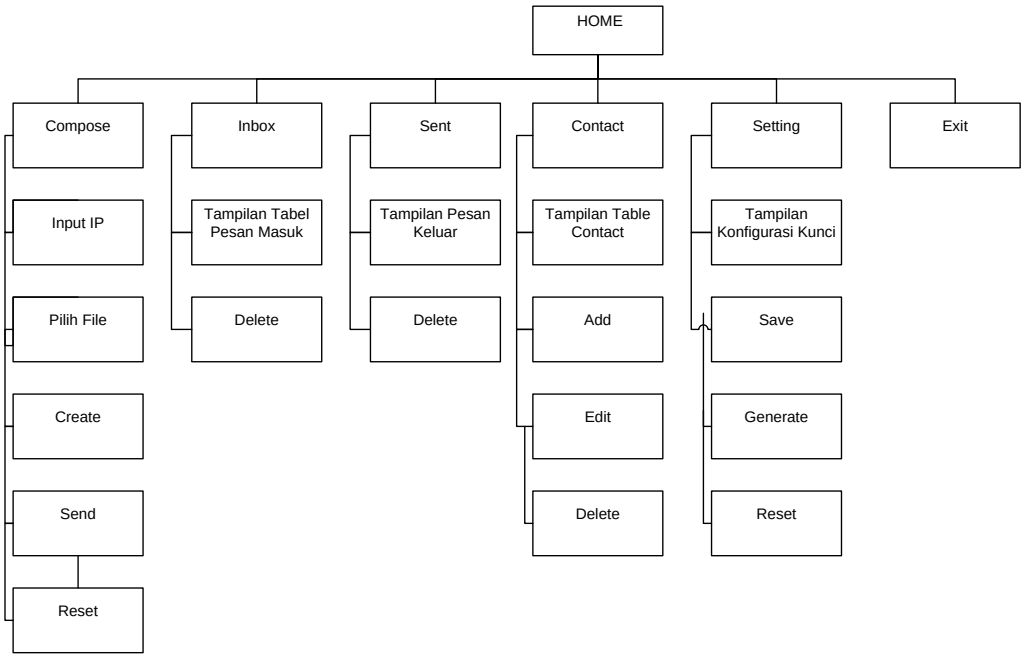
Perancangan struktur program . Mulai dari *generate key* kemudian masuk pada tahap penginputan IP, kemudian input file (.doc) dan masuk pada tahap pembuatan *signature*, tahap akhir file *signature* siap dikirim.



Gambar 3. 13. Struktur Kerja Aplikasi *Digital Signature*

3. 4. 5. Perancangan Struktur Menu

Struktur program merepresentasikan organisasi komponen program atau modul secara hirarki. Notasi yang digunakan merepresentasikan hirarki tersebut menggunakan diagram pohon. Fungsi pada struktur direpresentasikan dengan simbol persegi, input dan output digambarkan dengan anak panah. Adapun struktur program untuk perangkat lunak pembuka aplikasi tanda tangan digital adalah sebagai berikut :



Gambar 3. 14. Struktur Menu Perangkat Lunak

3. 4. 6. Perancangan Antarmuka

Perancangan antarmuka merupakan sebuah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Adapun perancangan antarmuka perangkat lunak tanda tangan digital adalah sebagai berikut :

3. 4. 6. 1. Perancangan Antarmuka *Form* Aplikasi

1. Desain *Form* Menu Utama



Gambar 3. 15. Form Menu Utama

2. Deskripsi Objek *form* Menu Utama

Berikut ini adalah deskripsi objek *form* Menu Utama :

Tabel 3. 8 Deskripsi Objek Menu Utama

Objek	Jenis	Keterangan
Home	Tab	Form utama
Compose	Tab	Form untuk membuat dokumen baru untuk dikirimkan
Inbox	Tab	Form untuk melihat pesan dokumen masuk

Sent	Tab	Form untuk melihat pesan keluar dokumen
Contact	Tab	Form untuk melihat dan menambah kontak baru
Setting key	Tab	Form untuk membuat kunci privat dan kunci publik
Exit	Tab	Form untuk keluar dari aplikasi

3. 4. 6. 2. Perancangan Antarmuka *Compose Message*

1. Desain form *compose message*



Gambar 3. 16. *Form Compose Message*

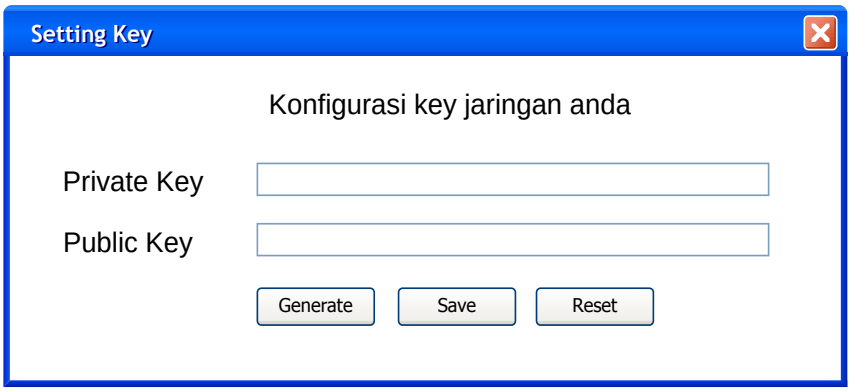
2. Deskripsi Objek

Tabel 3. 9 Deskripsi Objek *Form Compose Message*

Objek	Jenis	Keterangan
To(name)	Drop down	Input nama ip
To(IP)	Text	Input ip
File	Text	Input file dokumen
Browse	Button	Pilih file dokumen
Private key	Text	Input private key
Signature	Text	Input signature
Create	Button	Membuat signature
Send	Button	Kirim file dokumen
Cancel	Button	Keluar

3. 4. 6. 3. Perancangan Antarmuka form *setting key*

1. Desain form setting key



Gambar 3. 17. Form Setting Key

2. Deskripsi Objek

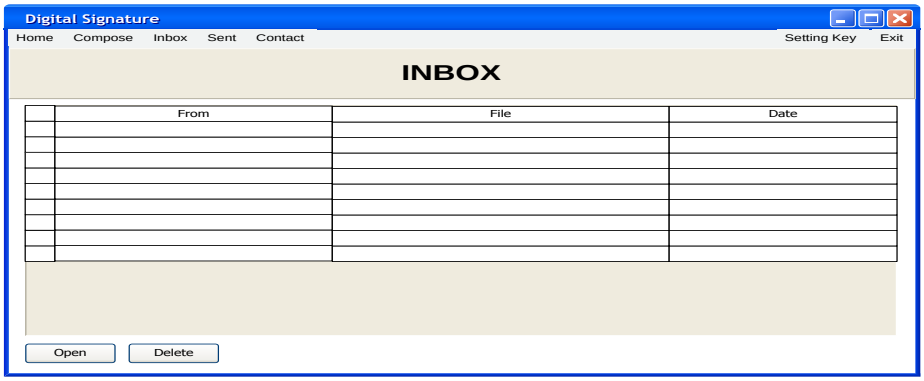
Berikut ini adalah deskripsi objek dari form *setting key* :

Tabel 3. 10 Deskripsi Objek Form *Setting Key*

Objek	Jenis	Keterangan
Private key	Text	Input private key
Publik key	Text	Input publik key
Generate	Button	Membuat kunci publik dan kunci privat
Save	Button	Simpan kunci
Reset	Button	Ulang buat kunci

3. 4. 6. 4. Perancangan Antarmuka Form *inbox*

1. Desain form inbox



Gambar 3. 18. Form Inbox

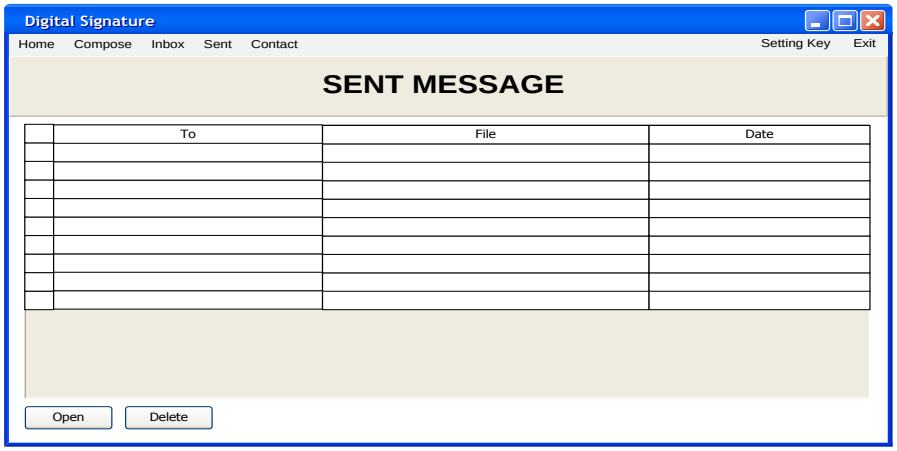
2. Deskripsi Objek

Tabel 3. 11 Deskripsi Objek Form Inbox

Objek	Jenis	Keterangan
Open	Button	Membuka pesan masuk
Delete	Button	Hapus pesan

3. 4. 6. 5. Perancangan Antarmuka *Form Sent*

1. Desain *form sent*



Gambar 3. 19 Form Sent

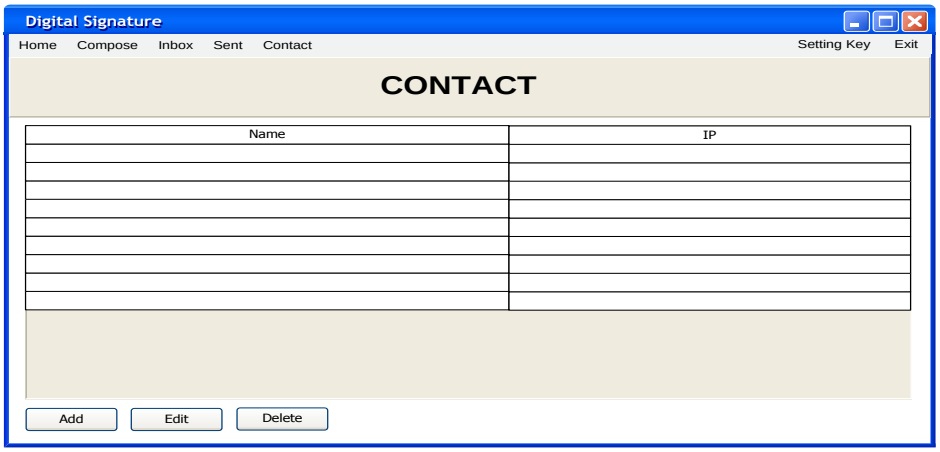
2. Deskripsi Objek

Tabel 3. 12 Deskripsi Objek Form Sent

Objek	Jenis	Keterangan
Open	button	Membuka file pesan keluar
Delete	Button	Hapus file pada tabel

3. 4. 6. 6. Perancangan Antarmuka *Form Contact*

1. Desain Form Contact



Gambar 3. 20. Form Contact

2. deskripsi objek

Tabel 3. 13. Deskripsi Objek *Form Contact*

Objek	Jenis	Keterangan
Add	Button	Menambah contact IP
Edit	Button	Mengubah contact
Delete	Button	Hapus kontak

3. 4. 6. 7. Perancangan Antarmuka *Form New Contact*

1. Desain *Form New Contact*



Gambar 3. 21. *Form New Contact*

2. Deskripsi Objek Form New Contact

Tabel 3. 14. Deskripsi Objek Form New Contact

Objek	Jenis	Keterangan
Name	Text	Input nama
IP	Text	Input IP
Save	Button	Simpan IP
Cancel	Button	Batal tambah kontak

3. 4. 6. 8. Perancangan Antarmuka form view message

1. Desain form view message



Gambar 3. 22. *Form View Message*

2. Deskripsi Objek view message

Tabel 3. 15. Deskripsi Objek *Form View Message*

Objek	Jenis	Keterangan
Form	Text	Input nama
Save contact	Button	Simpan ke kontak
File	Text	File dokumen yang masuk
Open	Button	Buka pesan dokumen
Signature	Text	View signature
Reply	Button	Balas pesan dokumen