BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analsisis sistem adalah penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan merupakan tahap dalam analisis sebuah sistem.

Dalam proses pembangunan suatu sistem perlu dilakukan penelitian dan analisa tentang sistem yang akan dibangun. Berikut akan dijelaskan tahapan-tahapan proses yang dilakukan untuk membangun aplikasi *digital signature* menggunakan ECDSA (*elliptic curve digital signature algorithm*) dan enkripsi menggunakan ElGamal berbasis kurva eliptik untuk keamanan e-dokumen. Meliputi proses pembentukan kunci, penandatanganan, proses enkripsi, proses dekripsi, dan proses verifikasi tanda tangan.

3.1.1 Analisis Masalah

Tahap analisis masalah dilakukan sebelum tahap perancangan sistem. Hal ini dilakukan untuk mengetahui dengan jelas masalah-masalah yang dihadapi, performa dan cara kerja algoritma yang digunakan, dan solusi menyelesaikan masalah tersebut. Berdasarkan hasil penelitian, Adapun masalah sistem yang berjalan di PT Len Industri adalah sebagai berikut:

1. PT Len Industri belum memiliki sebuah sistem untuk mengetahui legalisasi dan otentikasi/ keaslian pengirim pada e-dokumen sehingga bagian unit kerja yang ingin memiliki e-dokumen tersebut bisa mengetahui e-dokumen berasal dari unit kerja terkait yang sah.
2. PT Len Industri kesulitan dalam menjaga kerahasiaan terhadap dokumen yang bersifat rahasia seperti dokumen keuangan yang tidak semua bagian unit kerja boleh mengetehauinya.

3.1.2 Analisi Sistem yang Berjalan

Sistem yang sedang berjalan merupakan sistem yang sedang dipakai, sedangkan analisis sistem yang sedang berjalan merupakan cara untuk memahami masalah yang dihadapi oleh sistem, seperti mendefinisikan kebutuhan-kebutuhan fungsional dari sistem sehingga dapat diketahui apa saja kebutuhan pemakai yang belum terpenuhi oleh sistem yang sedang berjalan tersebut.

Berdasarkan hasil wawancara yang dilakukan di PT Len Industri, terdapat beberapa prosedur penyebaran dokumen baik dokumen yang bersifat rahasia atau dokumen yang boleh diketahui oleh semua unit kerja dilingkungan internal PT Len Industri sendiri.

3.1.2.1 Penyebaran Dokumen

Prosedur penyebaran dokumen di PT Len Industri yang sedang berjalan saat ini adalah sebagai berikut:

1. Setiap unit kerja mengunduh dokumen yang dibutuhkan di web internal perusahaan.
2. Semua jenis dokumen ada di web internal tersebut, karena dokumen tersebut harus bersifat transfaran diantara semua unit kerja di lingkungan PT Len Industri.
3. Semua unit kerja boleh mengakses seluruh dokumen yang terdapat di web internal tersebut.

3.1.2.2 Analisis *Digital Signature* dan Enkripsi Terhadap Masalah

Berdasarkan hasil wawancara terdapat permasalahan yaitu masalah legalisasi yang mengeluarkan dokumen di web internal perusahaan tersebut apakah benar dari unit kerja yang bersangkutan atau bukan. Selain itu seharusnya tidak semua dokumen yang terdapat di web internal perusahaan tersebut boleh diakses oleh semua unit kerja, terutama dokumen yang bersifat rahasia seperti dokumen keuangan tertentu, dokumen hukum, desain produk, dokumen strategi marketing, dan dokumen rahasia lainnya. Sehingga unit kerja yang terkait lebih sering mengirimkan dokumen rahasia tersebut secara *point-to-point* melalui jaringan. Solusi yang akan ditawarkan adalah dengan membangun aplikasi *digital*

signature dan enkripsi untuk keamanan e-dokumen di PT Len Industri. Aplikasi *digital signature* dan enkripsi ini memiliki keunggulan yaitu:

Keunggulan dari segi *digital signature.*

1. Dapat menjamin keaslian pengirim e-dokumen, sehingga setiap unit kerja dapat mengetahui bahwa e-dokumen yang diterima berasal dari unit kerja yang terkait.
2. Dapat menjamin anti-penyangkalan, dimana unit kerja penerima e-dokumen tersebut tidak bisa menolak bahwa unit kerja pengirim e-dokumen telah mengirim e-dokumen.

Keunggulan dari segi enkripsi.

1. Dapat menjamin kerahasian data sehingga tidak bisa dibaca oleh pihak yang tidak berwenang.
2. Dapat menjamin keutuhan data, dimana dokumen yang diterima tidak mengalami perubahan ketika disebarkan melalui jaringan.

3.1.3 Gambaran Umum Sistem

Sistem yang akan dibangun pada penelitian ini adalah aplikasi *digital signature* dan enkripsi untuk keamanan e-dokumen. *Digital Signature* merupakan mekanisme dalam kriptografi dimana pengirim pesan dapat menandatangani secara elektronik sebuah e-dokumen dan penerima dapat memverifikasi tanda tangan tersebut untuk mengetahui keaslian pengirim pesan. Sebelum dikirimkan kepada pihak penerima, pesan dan tanda tangan terlebih dahulu dienkripsi menggunakan sistem kriptografi kunci publik.

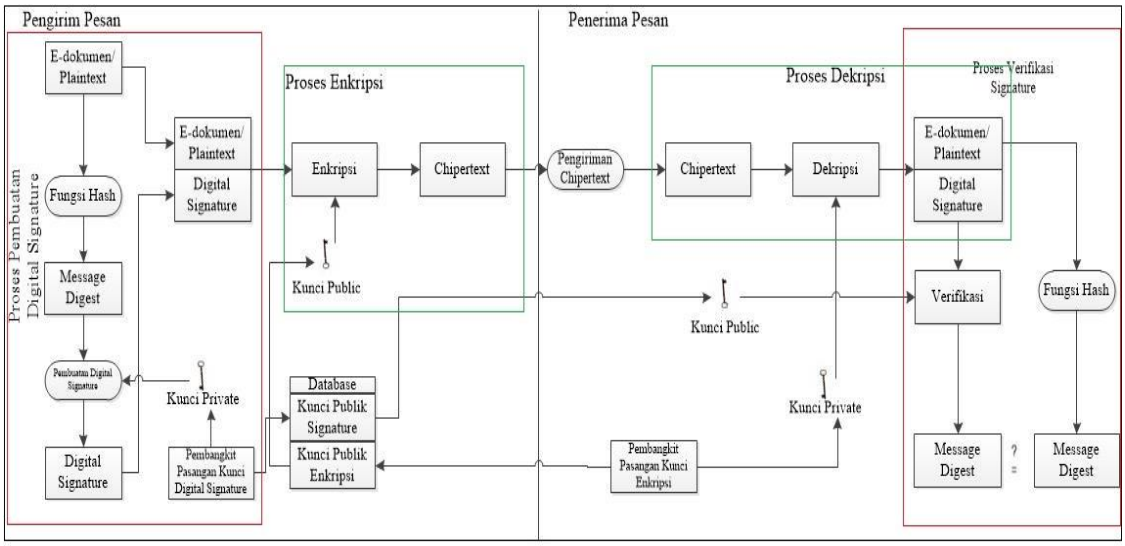
Aplikasi *Digital signature* dan enkripsi e-dokumen ini melibatkan dua pihak yaitu pembuat/pengirim (*signer*) pesan dan pihak penerima pesan sebagai *verifier*. Berikut penjelasan alur proses pada aplikasi *digital signature* dan enkripsi yang akan dibangun:

1. Pada pembuat/pengirim, pesan dimasukan ke dalam fungsi *hash* yang akan menghasilkan *message digest.*
2. Kemudian *message digest* ini ditandatangani menggunakan kunci *private* si pembuat pesan dan menghasilkan *digital signature*.
3. Kemudian pesan dan *digital signature*-nya dilekatkan dengan cara disambung.

4. Pesan dan *digital signature* yang telah dilekatkan kemudian di enkripsi menggunakan kunci *public* penerima pesan lalu kemudian pesan dikirim ke penerima.

Setelah menerima pesan, *verifier*/penerima pesan kemudian memverifikasi pesan dengan tahapan sebagai berikut:

1. Pada penerima pesan didekripsi menggunakan kunci *private* penerima pesan menghasilkan e-dokumen dan *digital signature*.
2. Kemudian *digital signature* diverifikasi, langkah pertama yaitu pesan dimasukan ke fungsi *hash* yang sama dengan fungsi *hash* pengirim pesan dan menghasilkan *message digest.*
3. *Digital signature* yang diterima bersama pesan tadi di verifikasi menggunakan kunci *public* yang merupakan pasangan kunci *private* pengirim pesan dan akan menghasilkan *message digest.*
4. Tahap selanjutnya yaitu mencocokan *message digest* dari hasil fungsi hash dengan *message digest* dari hasil dekripsi, apabila cocok maka pesan memang benar dari pihak pengirim.



Gambar 3. 1 Gambaran Umum Sistem

3.1.4 Analisis Algoritma

Analisis algoritma digunakan untuk mengetahui proses dari algoritma yang digunakan untuk dapat di implementasikan ke dalam aplikasi. Pada pembangunan aplikasi ini digunakan algoritma *Elliptic Curve Digital Signature Algorithm* (ECDSA) untuk pembuatan *digital signature* pada e-dokumen. Sementara itu untuk proses enkripsi menggunakan algortima ElGamal berbasis kurva eliptik.

3.1.4.1 Analisis Algoritma SHA-1

Fungsi Hash SHA-1 digunakan pada proses pembentukan *signature* dan verifikasi *signature*. Algoritma SHA-1 menerima masukan berupa pesan dengan ukuran maksimum 2.147.483.648 gigabyte dan menghasilkan *message digest* yang panjangnya 160 bit. Langkah-langkah pembuatan *message digest* secara garis besar adalah sebagai berikut:

1. Masukkan pesan
2. Inisialisasi *variable*
3. Penambahan bit-bit pengganjal (*padding bits*).
4. Penambahan nilai panjang pesan semula.
5. Inisialisasi penyangga (*buffer*) *message digest*.
6. Pengolahan pesan dalam blok berukuran 512 bit.



Gambar 3. 2 Analisis SHA-1

3.1.4.2 Analisis Algoritma ECDSA

Misalkan unit kerja bagian Hukum di PT Len Industri mengirimkan pesan yang berisi e-dokumen kepada bagian Keuangan, ambil contoh pesan (m) = 4564579. Untuk menjaga legalitas dan kerahasiaan pesan maka bagian Hukum menggunakan *digital signature* dan enkripsi untuk selanjutnya pesan tersebut dikirimkan kepada bagian Keuangan. Bagian hukum menggunakan fungsi *hash* yang basis pembaginya adalah 2980, yaitu h(k) = k mod 2980.

Langkah pertama adalah menemukan nilai *hash* (‘m) dari pesan (M) menggunakan fungsi *hash* berikut perhitungannya:

$$m' = h(M) = 4564579 \; mod \; 2980$$

$$m' = 2199$$

Langkah selanjutnya adalah menentukan parameter domain untuk kurva eliptik terlebih dahulu. Misalkan parameter yang diambil adalah sebagai berikut:

1 Batasan daerah (q) adalah bilangan prima 23

2 *Field Representation* (FR) :

$$y^2 mod \; 23 = (x^3 + ax + b) mod 23$$

$$4a^3 + 27b^2 \; mod \; 23 \; \neq 0$$

3 Parameter kurva: a=1, b=4

4 Ambil titik generator G: (0,2)

5 Orde n adalah bilangan prima harus lebih besar dari 4*sqrt(23) ≈19.18, maka ambil n=23.

6 Kofaktor h:

$$h = \#E(F_{23})/n \; (3.17)$$

$$h = 29/23 \approx 1.261 \approx 1$$

Maka diperoleh parameter domain yang harus disetujui oleh bagian Hukum dan bagian Keuangan yaitu D(q,FR,a,b,G,n,h)=(23,F_{23},1,4,(0,2),23,1). Langkah selanjutnya adalah proses pembangkitan kunci, berikut langkah-langkah proses pemabngkitan kunci:

1. Tentukan nilai d_A=20.
2. Maka Q_A=20 x G = (10,20)

3. Maka diperoleh kunci privat d_A=20 dan kunci publik Q_A=(10,20)

Berikut merupakan tahapan pembangkitan sepasang kunci ECDSA.

Tabel 3. 1 Pembangkitan Sepasang Kunci ECDSA

```
1   Procedure: BuatKunci (input (dA: Integer),
2   Output: K_privat, K_publik (String)
3   {Membuat kunci publik dan kunci privat
4   Pilih kurva eliptik pada GF(p) dengan parameter a dan b,
5   yaitu kurva eliptic E_p (a,b).
6   Masukan : dA
7   Keluaran : K_privat (dA), K_publik(QA)}
8   Deklarasi
9   dA : integer
10  Algoritma
11  Repeat {
12   dA ← random()} //fungsi pembangkit bilangan acak
13  Until (1,n-1)
14  Endrepeat
15   QA ← dA.(X1,Y1)
16   K_Privat ← dA
17   K_Publik ← QA
```

Setelah mendapatkan pasangan kunci tersebut , langkah selanjutnya adalah melakukan tanda tangan (*signing*), sebagai berikut:

1. Panggil kembali e = m'= 2199
2. Tentukan k = 4
3. (X_1, Y_1)=4.(0,2)=(1,12) dan r = 1 mod 23 = 1
4. Perhitungan s = 4^{-1}(2199+18.14) mod 23 = 9
5. Maka diperoleh (1,9)

Berikut merupakan tahapan pembentukan *digital signature* menggunakan algoritma ECDSA.

Tabel 3. 2 Pembentukan Digital Signature Menggunakan Algoritma ECDSA

```
1   Procedure: Pembuatan Signature input: m, K_Privat, K_Publik
2   Output: r,s {signature}
```

```
{membuat signature dari Pesan m, masukan berupa m(pesan), K_privat(dA)
K_Publik (QA)
keluaran Signature (r,s)}
Deklarasi: h(m) : string
Algoritma
m ← m
dA ← dA
H(m) ← SHA-1(m) {Ubah pesan menjadi message digest
menggunakan fungsi SHA-1}
Repeat
k ← random(){ cek nilai k [1,n-1] }
QA ← k . G (X1,Y1)
r ← X1 mod q
Until  r=0
Endrepeat
K ^ -1 mod q
e ← Hash(m)
s = k ^ -1 (e+dA • r) mod q
Signature ← (r,s)
```

Dengan demikian diperoleh *digital signature* (14,16). Andi mempunyai pesan M = 4564579, D = {23,F_{23}, 1,4, (0,2), 23,1 }, dan kunci publik $Q_A = (10,20)$, serta signature (r,s)=(1,9). Selanjutnya pesan, dan signature tersebut dienkripsi menggunakan algoritma ElGamal berbasis kurva eliptik, berikut langkah-langkahnya.

Berikut adalah proses verifikasi dari *digital signature* tersebut:

1. Cek r dan s berada pada [1,n-1], ternyata benar 14 dan 16 berada pada [1,22].
2. Hitung nilai hash e = hash(4564579.)=2199.
3. Hitung w = 9^{-1} (mod 23) =20.
4. Hitung $u_1 = 2199.20$ (mod 23) = 22 dan $u_2 = 1.20$ (mod 23) = 20
5. (X_1, Y_1) = 22G + 18Q_A=(15,17)+(11,9)=(1,11)
6. Diperoleh r ≡ X_1 (mod 23) terpenuhi.

Berikut merupakan tahapan verifikasi *digital signature* menggunakan algoritma ECDSA.

Tabel 3. 3 Tahapan Verifikasi Digital Signature

```
1   Procedure:Verifikasi Signature input:m,r,s : file;
2   K_Publik
3   {memverifikasi signature menggunakan kunci publik pengirim}
4   Output: Valid : boolean
5   Deklarasi: Hash(m) : string
6   Algoritma
7   m ← m
8   r ← r
9   s ← s
10  w ← s ^ -1 mod q
11  e(m) ← SHA-1(m) {Ubah pesan menjadi message digest
12  menggunakan fungsi SHA-1}
13  u1 ← (e(m) * w) mod q
14  u2 ← (r * w) mod q
15  v ← (((g ^ u1) * (y ^ u2)) mod p) mod q)
16  if v = r then
17  valid ← true
18  else
19  valid ← false
20  endif
```

3.1.4.3 Analisis Algoritma ElGamal Berbasis Kurva Eliptik

Pada kasus ini kunci yang digunakan untuk proses tanda-tangan dan proses enkripsi berbeda. Misalkan suatu kurva eliptik E: $y^2 = x^3 + x + 6$ atas Z_{11}.

Algoritma Pembuatan Kunci

Parameter domain kurva eliptik (p=17, E, P=(2,7), n=13)

1. Tentukan nilai d ∈(1, n-1)= 19
2. Hitung nilai Q= dP = 19 (2,7) = (19,2)
3. Hasil (Q,d) = {Q,d}={(7,2),7}

Jadi kunci publik Q = (19,2) dan kunci privat d = 19

Berikut merupakan tahapan pembentukan sepasang kunci enkripsi menggunakan algoritma Elgamal berbasis kurva eliptik.

Tabel 3. 4 Pembentukan Sepasang Kunci Enkripsi

```
1  Procedure: BuatKunci (input (V: Integer),
2  Output: Kprivat, Kpublik (String)
3  {Membuat kunci publik dan kunci privat
4  Pilih kurva eliptik pada GF(p) dengan parameter a dan b,
5  yaitu kurva eliptic E_p (a,b).
6  Masukan : V
7  Keluaran : K_privat (V), K_publik(GE)}
8  Deklarasi
9  V : integer
10 Algoritma
11 Repeat {
12  V ← random()} //fungsi pembangkit bilangan acak
13 Until (1,n-1)
14 Endrepeat
15  β ← V.(X1,Y1)
16  K_Privat ← V
17  K_Publik ← β
```

Setelah kedua pasangan kunci didapatkan maka tahap selanjutnya adalah proses enkripsi. Berikut langkah-langkah perhitungannya:

1. misalkan pesan m = (33,29) pada E(Z_p)
2. Pilih k ∈ [1, n-1] = 23
3. Hitung $C_1 = kP = 3(2,7) = (8,3)$
4. Hitung $C_2 = M + kQ = (33,29) + 3(2,7) = (8,3)$

Berikut merupakan tahapan enkripsi menggunakan algoritma Elgamal berbasis kurva eliptik.

Tabel 3. 5 Tahap Enkripsi

```
1  Procedure: Enkripsi Pesan input: Pm, KPrivat, KPublik
2  Output:
3  {enkripsi dari Pesan m, Chipertext,
4  keluaran Chipertext P_M (P1,P2)}
5  Deklarasi: P_c : Chipertext
```

6	**Algoritma**
7	$P_M \leftarrow P_M$
8	$\beta \leftarrow \beta$
9	Repeat
10	k ← random(){ cek nilai k [1,n-1] }
11	Until k ≠[1,n-1]
12	Endrepeat
13	$P_1 \leftarrow$ k. X_1, Y_1
14	$P_1 \leftarrow P_M +$ k. β
15	$P_C \leftarrow P_1$, P_2 (Chipertext)

Maka chipertext yang didapat adalah (C_1, C_2) = {(8,3), (10,2)}. Setelah itu kemudian Andi mengirim pesan kepada Budi. Budi kemudian memdekripsi pesan yang diterima dari Andi dengan langkah-langkah sebagai berikut:

Parameter Domain (p,E,P,n), kunci pribadi d, chiphertext ($C_1 C_2$)

Outuput: Pesan m dan *signature*

1. Hitung M =$C_2 - dC_1$ = (10,2) – 7 (8,3) = (10,2) - (3,5) = (10,2) + (3,6) = (10,9)
2. Hasil (m) = {(10,9)}.

Berikut merupakan tahapan dekripsi menggunakan algoritma Elgamal berbasis kurva eliptik.

Tabel 3. 6 Tahap Dekrispi

1	**Procedure**:Dekripsi Pesan input: P_C, V: file;
2	{dekripsi pesan menggunakan kunci privat penerrima }
3	**Output**: P_M : file
4	**Deklarasi:**
5	**Algoritma**
6	$P_1 \leftarrow P_1$
7	$P_2 \leftarrow P_2$
8	$V \leftarrow V$
9	$M_1 \leftarrow V$. P_1
10	$P_M \leftarrow P_2$. M_1

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan tahap analisis yang digunakan untuk memastikan bahwa aplikasi yang akan dibangun sesuai dengan kebutuhan perusahaan dan nantinya layak untuk dikembangkan. Terdapat tiga tahap analisis non funsional yaitu analisis perangkat keras, analisis perangkat lunak, analisis pemakai, dan analisis jaringan. Berikut penjelasan dari masing-masing tahapan analisis:

3.1.5.1 Analisis Kebutuhan Perangkat Keras (Hardware)

Dari hasil observasi di lapangan perangkat komputer saat ini yang terdapat di PT Len Industri adalah sebagai berikut:

Tabel 3. 7 Perangkat Keras di PT Len Industri

Perangkat Keras	**Keterangan**
Processor	Intel Dual Core 2.50 GHz
RAM	1 GHz
Harddisk	140 GHz
VGA	On Board 384 MB
Monitor	LCD 14 Inch, resolusi 1024 x 768 pixel

Spesifikasi yang dibutuhkan untuk menjalankan apliksi ini di sisi client adalah sebagai berikut:

Tabel 3. 8 Kebutuhan Perangkat Keras Untuk Menjalankan Aplikasi Pada Client

Perangkat Keras	**Keterangan**
Processor	Intel Dual Core 2.00 GHz
RAM	1 GHz
Harddisk	80 GHz
VGA	On Board
Monitor	Bebas

Sedangkan spesifikasi yang dibutuhkan untuk menjalankan aplikasi di sisi server adalah sebagai berikut:

Tabel 3. 9 Tabel Kebutuhan Server

Perangkat Keras	**Keterangan**
Processor	Intel Dual Core 2.00 GHz
RAM	2 GHz
Harddisk	350 GHz
VGA	On Board
Monitor	Bebas

Berdasarkan penjelasan dari (tabel 3.1) dan (tabel 3.2) diatas maka aplikasi *digital signature* dan enkripsi ini dapat berjalan pada perangkat komputer yang terdapat di PT Len Industri.

3.1.5.2 Analisis Kebutuhan Perangkat Lunak (Software)

Perangkat lunak yang ada saat ini di PT Len Industri adalah sebagai berikut:

Tabel 3. 10 Perangkat Lunak di PT Len Industri

Perangkat Lunak	**Keterangan**
Sistem Operasi	Windows 7

Untuk membangun aplikasi ini dibutuhkan perangkat lunak dengan spesifikasi minimal seperti pada tabel 3.11:

Tabel 3. 11 Spesifikasi Kebutuhan Perangkat Lunak

Perangkat Lunak	**Keterangan**
Sistem Operasi	Windows 7
Bahasa Pemrograman	Java
Development Tools	NetBeans IDE 7.1
Perangkat Lunak Untuk Kompilasi	JDK (*Java Development Kit*) 63.1
Perangkat Lunak Untuk Menjalankan Aplikasi yang Dibangun	JRE (*Java Runtime Environtment*)

Berdasarkan penjelasan dari (tabel 3.3) dan (tabel 3.4) maka aplikasi *digital signature* dan enkripsi ini dapat di jalankan pada spesifikasi perangkat lunak yang terdapat di PT Len Industri dengan syarat perangkat lunak JRE (*Java Runtime*

Environtment) telah ter*install,* sedankan untuk JDK (*Java Development Kit*) tidak harus ter*install.*

3.1.5.3 Analisis Pengguna

Untuk menggunakan aplikasi ini dibutuhkan pengguna dengan ciri-ciri sebagai berikut:

Tabel 3. 12 Tabel Analisis Pengguna

Pengguna	Hak Akses	Keterampilan
Pengirim Pesan	Pembuatan Kunci, Pembuatan Signature, Enkripsi Pesan, Pengiriman Pesan.	Bisa menggunakan komputer, dan menjalankan aplikasi *digital signature* dan enkripsi.
Penerima Pesan	Dekripsi Pesan, Verifikasi Pesan	Bisa menggunakan komputer, dan menjalankan aplikasi *digital signature* dan enkripsi.

3.1.5.4 Analisis Jaringan

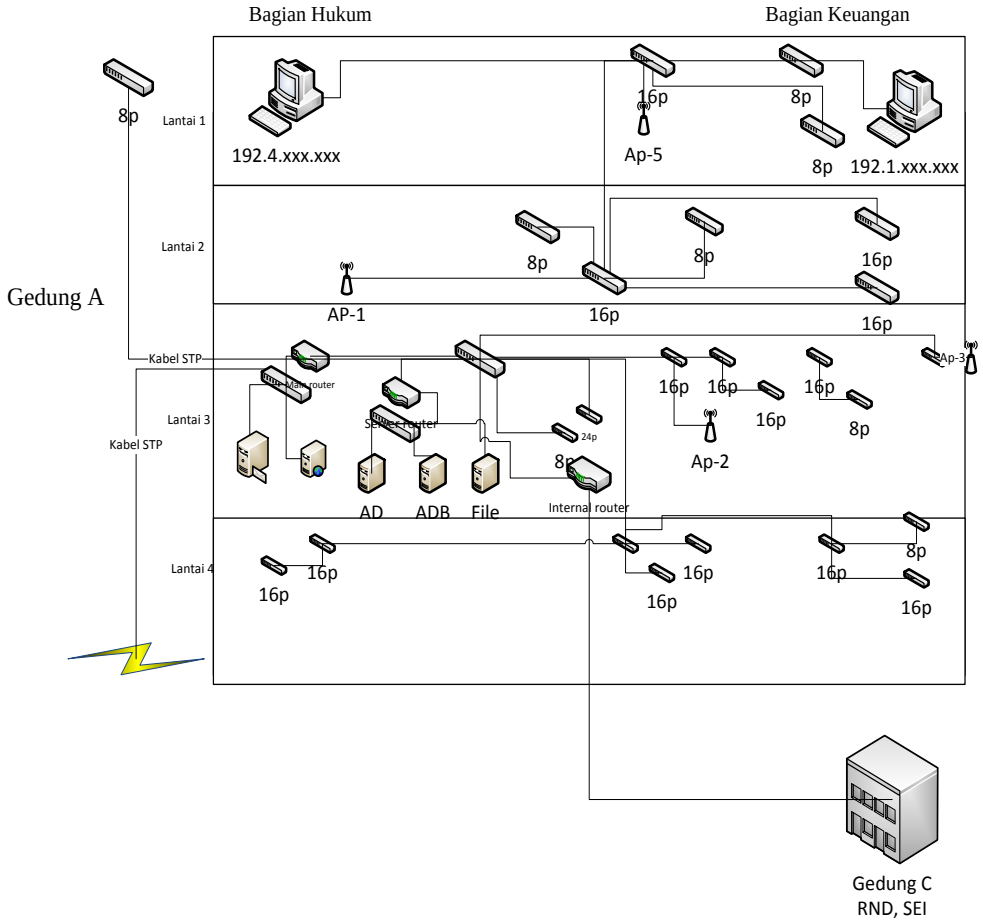
Analisis jaringan merupakan tahap analisis yang menjelaskan pemetaan jaringan yang terdapat di PT Len Industri tempat aplikasi *digital signature* dan enkripsi ini akan dijalankan. Tahap analisis ini diperlukan untuk memastikan bahwa aplikasi ini dapat berjalan pada sistem jaringan yang sudah ada. Berikut penjelasan jaringan pada PT Len Industri:

1. Terdapat sepuluh gedung pada komplek bangunan PT Len Industri, masing-masing gedung diberi nama menggunakan abjad *alphabet,* berikut penjelasannya:

a. Gedung A, merupakan gedung utama tempat semua kegiatan korporasi PT Len Industri dan memiliki empat lantai .
b. Gedung B, P dan Q merupakan tempat produksi.
c. Gedung E, koperasi.
d. Gedung C, merupakan gedung anak perusahaan yaitu SEI dan Eltran.
e. Gedung D, merupakan kantor dari bagian SDM.
f. Gedung T, merupakan gedung anak perusahaan LRS.
g. Gedung U, Kantin.
h. Masing-masing gedung diberi akses ke jaringan kecuali gedung U dan Q.

1. Masing-masing Jaringan LAN dipisahkan menjadi empat *subnet* yaitu:
   - IP 192.4.xxx.xxx digunakan di lantai 1 dan lantai 2 gedung A.
   - IP 192.1.xxx.xxx digunakan di lantai 3 dan 4 gedung A.
   - IP 192.2.xxx.xxx digunakan di gedung B,C,D.
   - IP 192.3.xxx.xxx digunakan di gedung T.
2. Karena semua kegiatan korporasi dari PT Len Industri terdapat di gedung A, Aplikasi *digital signature* dan enkripsi e-dokumen ini nantinya akan digunakan digedung A.
3. Pada gedung A ini terdapat dua model jaringan intranet yang digunakan, yaitu model jaringan yang menggunakan kabel (*wired*) dan model jaringan tanpa kabel (*wireless*).
4. Media penghubung untuk jaringan yang menggunakan kabel yaitu dengan menggunakan *switch X port*. *Ip address* yang digunakan untuk jaringan *wired* ini yaitu *ip address* statik (tetap).
5. Media penghubung untuk jaringan *wireless* menggunakan *acces point*, terdapat empat *acces point* di gedung A yaitu satu ditempatkan di lantai 1, satu di lantai 2, dan dua lagi ditempatkan di lantai 3. Pembagian *ip address* dilakukan oleh DHCP *server*, sehingga *ip address* –nya bersifat dinamis/ dapat berubah-ubah.
6. Dalam satu bagian unit kerja memungkinkan karyawan menggunakan media *wired* (kabel) dan *wireless* (tanpa kabel) untuk terhubung ke jaringan. Supaya bisa digunakan di dua model jaringan tersebut aplikasi ini menggunakan alamat *MAC address* sebagai alamat dari pengirim dan penerima pesan.

Layout Jaringan PT Len Industri



Gambar 3. 3 Layout Jaringan PT Len Industri

Karena semua kegiatan korporasi dari PT Len Industri terdapat di gedung A, maka aplikasi *digital signature* dan enkripsi e-dokumen ini nantinya hanya akan digunakan digedung A. Gambar 3.3 berikut merupakan layout jaringan dari gedung A PT Len Industri.

Di gedung A sendiri masing-masing Jaringan LAN dipisahkan menjadi dua *subnet* yaitu:

1. IP 192.4.xxx.xxx digunakan di lantai 1 dan lantai 2.
2. IP 192.1.xxx.xxx digunakan di lantai 3 dan 4.

Pengiriman pesan menggunakan aplikasi *digital signature* dan enkripsi ini dilakukan secara *point to point* yaitu antara satu bagian unit kerja dengan satu bagian unit kerja yang lain. Misalnya pengiriman pesan terjadi antara bagian

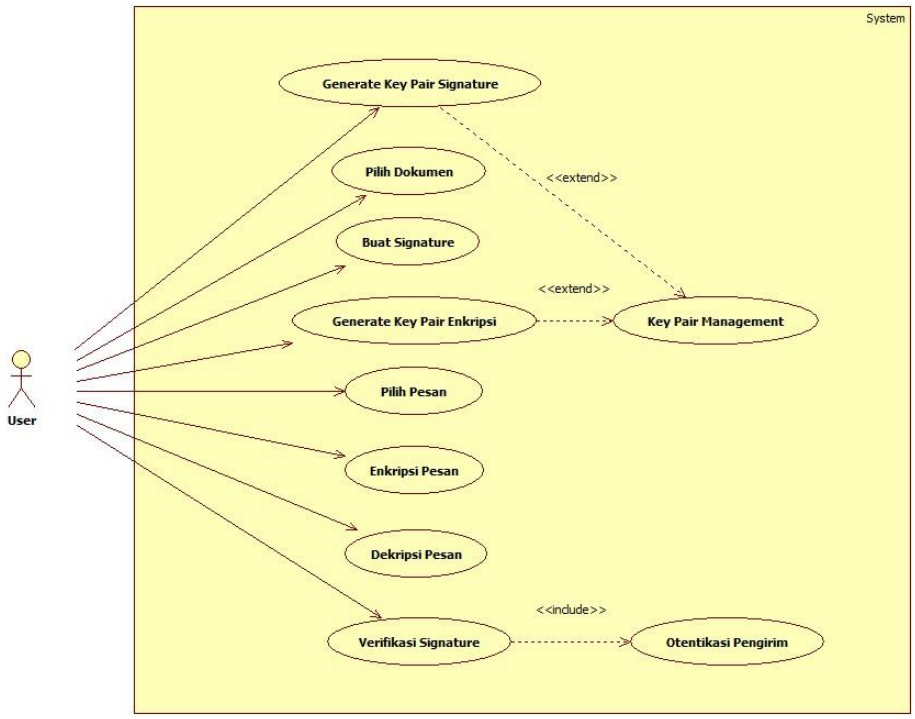
hukum dan bagian keuangan yang keduanya terdapat dilantai satu gedung A dengan subnet 192.4.xxx.xxx.

3.1.6 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan tahap analisis dengan cara memodelkan sistem, permodelan yang digunakan untuk memodelkan apliksi yang akan dibangun adalah permodelan berorientasi objek. Perangkat lunak yang digunakan StarUML, tools yang digunakan adalah *Use Case Diagram, Sequence Diagram, dan Activity Diagram*.

3.1.6.1 Use Case Diagram

Use case diagram aplikasi *digital signature* dan enkripsi ini dapat dilihat pada gambar 3.4 berikut:



Gambar 3. 4 Use Case Diagram

3.1.6.2 Skenario Use Case

Skenario *Use Case* berfungsi untuk menjelaskan modul-modul yang terdapat dalam aplikasi secara lebih detail. Tabel 3.13 akan menjelaskan skenario *use case* pada pembangunan aplikasi *digital signature* dan enkripsi.

3.1.6.2.1 Skenario *Use Case Generate Key Pair Signature*

Tabel 3. 13 Skenario Use Case Generate Key Pair Signature

Identifikasi	
Nama *Use Case*	Generate Key Pair Signature
Aktor	User/ Pengirim
Deskripsi	Membuat sepasang kunci yaitu kunci privat dan kunci publik
Skenario Utama	
Kondisi	Form *Generate Pair Key Signature*
Aksi	
Aktor	Sistem
Memilih Menu Pengaturan	Menampilkan Sub Menu Generate Key Pair Signature
Menekan Tombol *Generate*	Menampilkan Kunci

3.1.6.2.2 Skenario *Use Case* Pilih File

Tabel 3. 14 Tabel Skenario Pilih File

Identifikasi	
Nama *Use Case*	Pilih File
Aktor	User/ Pengirim
Deskripsi	Memilih File
Skenario Utama	
Kondisi	Form Buat Pesan
Aksi	
Aktor	Sistem
Memilih Menu Buat Pesan	Menampilkan Form Buat Pesan
Menekan Tombol Pilih File	Menampilkan *File Chooser*

3.1.6.2.3 Skenario *Use Case* Buat *Signature*

Tabel 3. 15 Skenario Use Case Pembuatan Signature

Identifikasi	
Nama *Use Case*	Buat *Signature*

Aktor	User/ Pengirim
Deskripsi	Membuat *signature* pada e-dokumen
Skenario Utama	
Kondisi	Form Buat Pesan
Aksi	
Aktor	Sistem
Memilih Menu Buat Pesan	Menampilkan Form Buat Pesan
Menekan Tombol Buat *Signature*	Menampilkan Hasil Signature

3.1.6.2.4 Skenario *Use Case Generate Key* Enkripsi

Tabel 3. 16 Skenario Use Case Generate Enkripsi

Identifikasi	
Nama *Use Case*	*Generate Key Pair Encryption*
Aktor	User/ Penerima
Deskripsi	Membuat sepasang kunci yaitu kunci privat dan kunci publik
Skenario Utama	
Kondisi	Form *Generate Pair Key Encryption*
Aksi	
Aktor	Sistem
Memilih Menu Pengaturan	Menampilkan Sub Menu *Generate Key Pair Encryption*
Menekan Tombol *Generate*	Menampilkan Kunci

3.1.6.2.5 Skenario *Use Case* Pilih Pesan

Tabel 3. 17 Skenario Use Case Pilih Pesan

Identifikasi	
Nama *Use Case*	Pilih Pesan
Aktor	User/ Pengirim
Deskripsi	Memilih Pesan
Skenario Utama	
Kondisi	Form Buat Pesan

Aksi	
Aktor	Sistem
Menekan Tombol Pilih Pesan	Menampilkan *File Chooser*

3.1.6.2.6 Skenario *Use Case* Enkripsi Pesan

Tabel 3. 18 Skenario Use Case Enkripsi Pesan

Identifikasi	
Nama *Use Case*	Enkripsi Pesan
Aktor	User/ Pengirim
Deskripsi	Mengenkripsi Pesan (e-dokumen dan *signature*)
Skenario Utama	
Kondisi	Form Tulis Pesan
Aksi	
Aktor	Sistem
Menekan Tombol Enkrip Pesan	Menampilkan Hasil Enkripsi Pesan

3.1.6.2.7 Skenario *Use Case* Dekripsi Pesan

Tabel 3. 19 Skenario Use Case Dekripsi Pesan

Identifikasi	
Nama *Use Case*	Dekripsi Pesan
Aktor	User/ Penerima
Deskripsi	Mendekripsi Pesan
Skenario Utama	
Kondisi	Form Pesan Masuk
Aksi	
Aktor	Sistem
Memilih Menu Pesan Masuk	Menampilkan Menu Pesan Masuk
Menekan Tombol Dekripsi Pesan	Menampilkan Hasil Dekripsi Pesan

3.1.6.2.8 *Skenario Use Case* Verifikasi *Signature*

Tabel 3. 20 Skenario Use Case Verifikasi Signature

Identifikasi	
Nama *Use Case*	Verifikasi *Signature*
Aktor	User/ Penerima
Deskripsi	Meverifikasi *digital signature*
Skenario Utama	
Kondisi	Form Pesan Masuk
Aksi	
Aktor	Sistem
Memilih Menu Pesan Masuk	Menampilkan Verifikasi Pesan

3.1.6.2.9 Skenario *Use Case Key Pair Management* Pengirim Pesan

Tabel 3. 21 Skenario *Use Case Key Pair Management* Pengirim Pesan

Identifikasi	
Nama *Use Case*	*Key Pair Management*
Aktor	User/ Pengirim
Deskripsi	Mengatur pertukaran kunci *digital signature* dan enkripsi
Skenario Utama	
Kondisi	Form Buat Pesan
Aksi	
Aktor	Sistem
1. Pengirim pesan menyimpan kunci publik *signature* di *database*	
Ketika penerima pesan akan memverifikasi *signature* tersebut penerima pesan mengambil kunci publik pengirim pesan dari *database.*	
2. Penerima pesan menyimpan kunci publik di *database*	

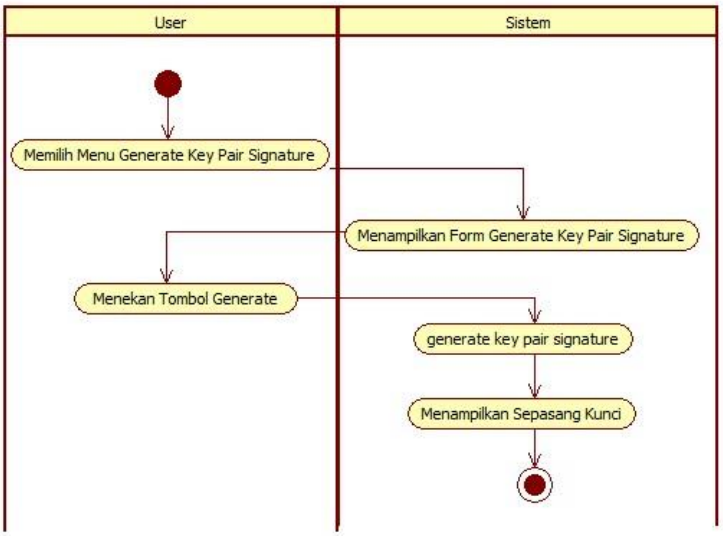
Ketika pengirim pesan akan meng-enkripsi pesan, pengirim pesan tersebut mengambil kunci publik enkripsi dari *database*	

3.1.6.3 Activity Diagram

Activity diagram merupakan teknik untuk mendeskripsikan aktifitas yang terjadi pada sistem, bagaimana alurnya, pilihan pengambilan keputusan, dan bagaimana akhirnya. Berikut *activity diagram* pada pembangunan aplikasi *digital signature* dan enkripsi ini:

3.1.6.3.1 *Generate Key Pair Signature*

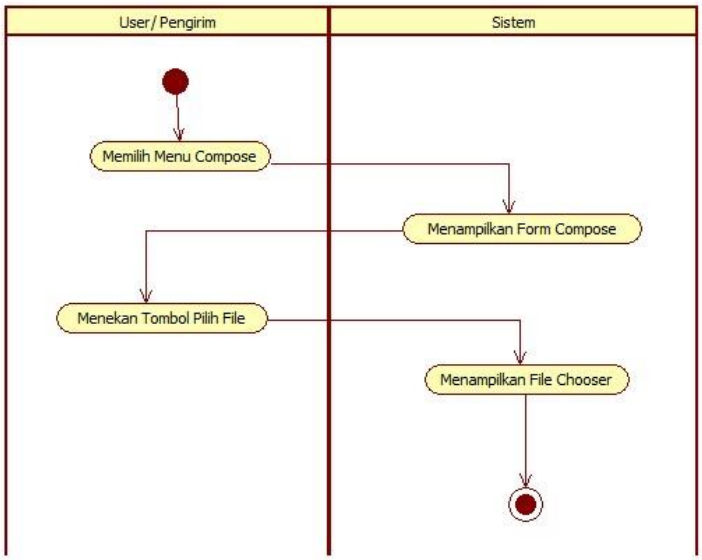
Berikut adalah *activity diagram generate key pair signature* pada pembabangunan aplikasi digital signature dan enkripsi untuk keamanan e-dokumen di PT Len Industri.



Gambar 3. 5 Activity Diagram Buat Kunci

3.1.6.3.2 Pilih *File*

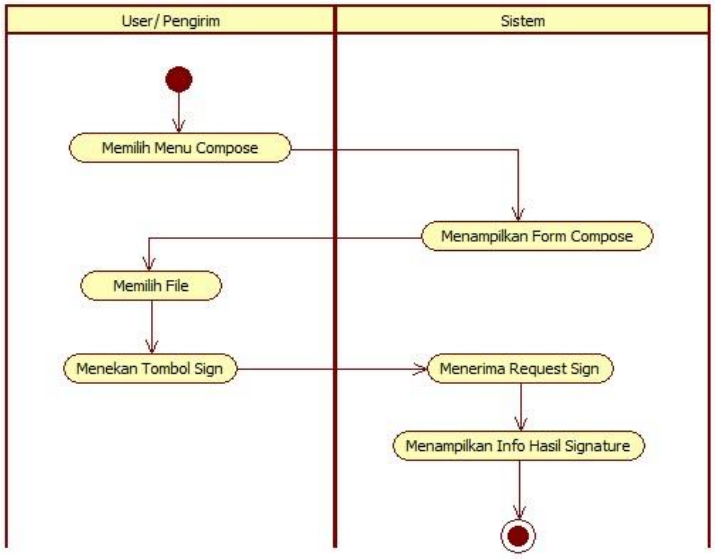
Pada gambar 3.6 menjelaskan *activity diagram* pilih *file.*



Gambar 3. 6 Activity Diagram Pilih File

3.1.6.3.3 Buat *Signature*

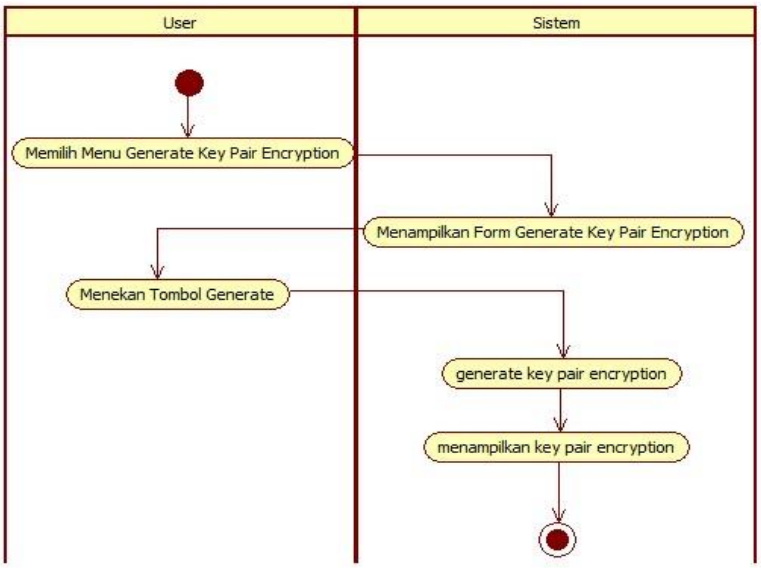
Pada gambar 3.7 menjelaskan *activity diagram* buat *signature*.



Gambar 3. 7 Activity Diagram Buat Signature

3.1.6.3.4 *Generate Key Pair Encryption*

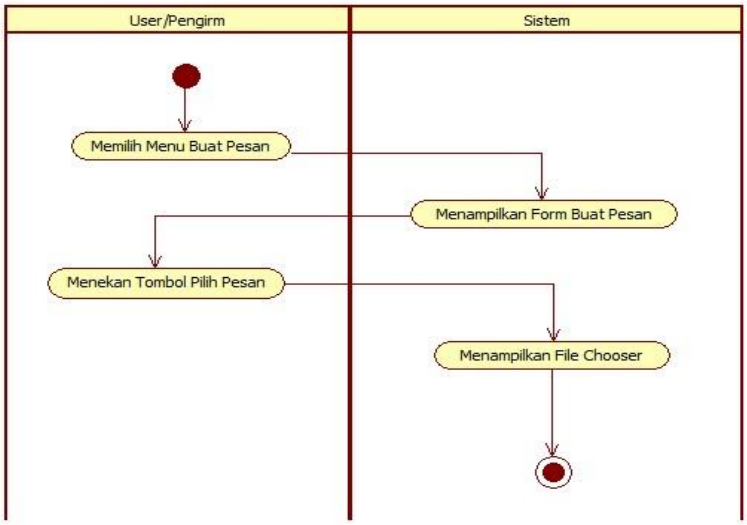
Pada gambar 3.8 menjelaskan *activity diagram generate key pair encryption*.



Gambar 3. 8 Activity Diagram Generate Key Pair Enkripsi

3.1.6.3.5 Pilih Pesan

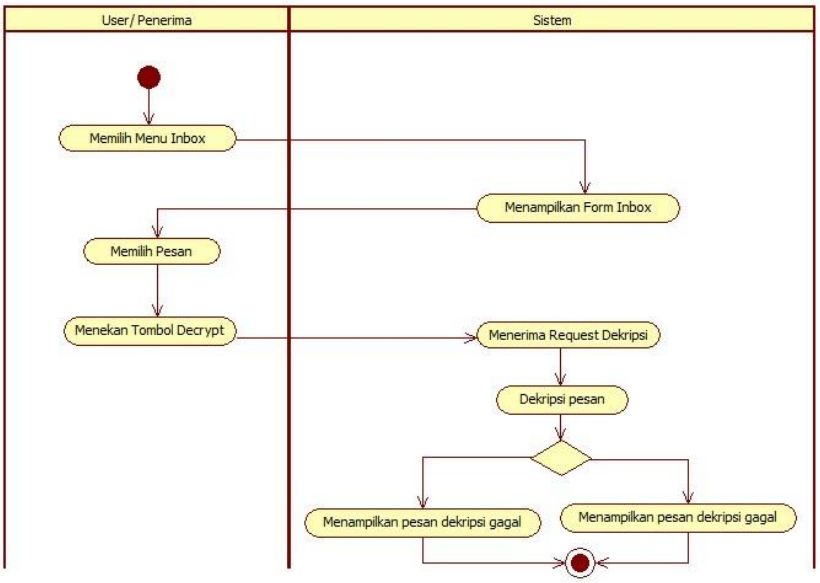
Pada gambar 3.9 menjelaskan *activity diagram* pilih pesan.



Gambar 3. 9 Activity Diagram Pilih Pesan

3.1.6.3.6 Enkripsi Pesan

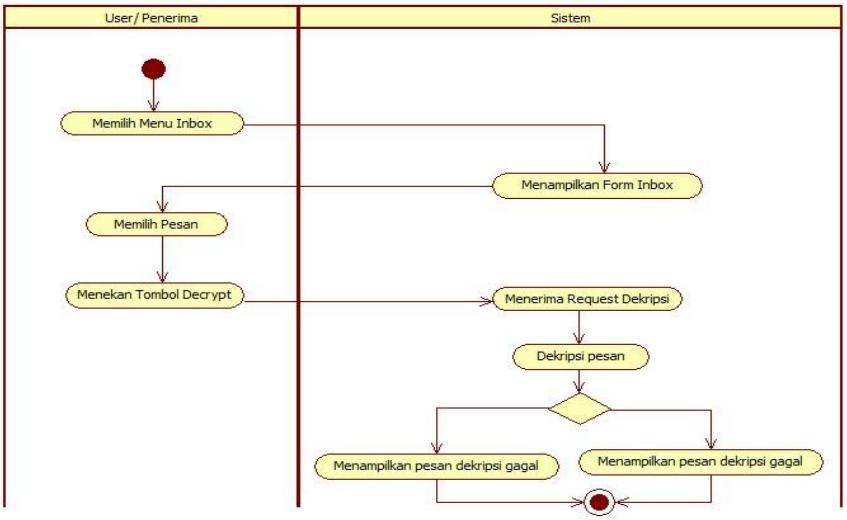
Pada gambar 3.10 menjelaskan *activity diagram* enkripsi pesan.



Gambar 3. 10 Activity Diagram Enkripsi Pesan

3.1.6.3.7 Dekripsi Pesan

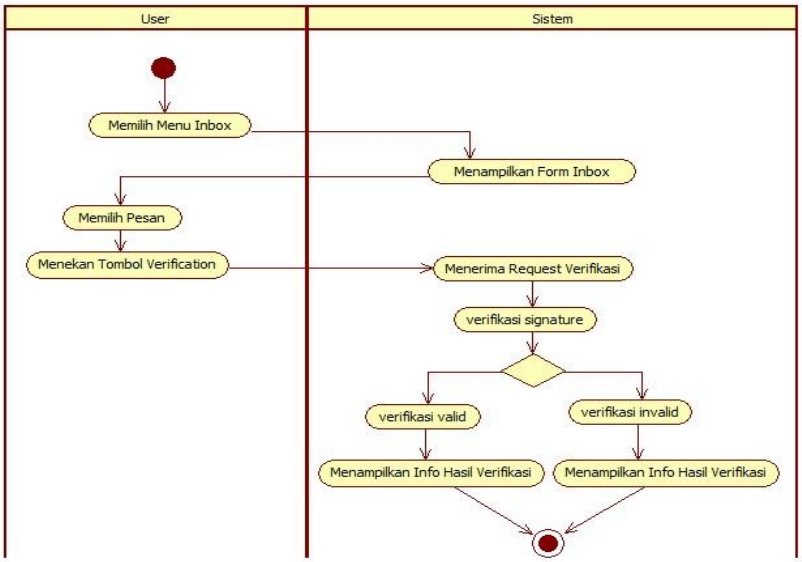
Pada gambar 3.11 menjelaskan *activity diagram* dekripsi pesan.



Gambar 3. 11 Activity Diagram Dekripsi Pesan

3.1.6.3.8 Verifikasi *Signature*

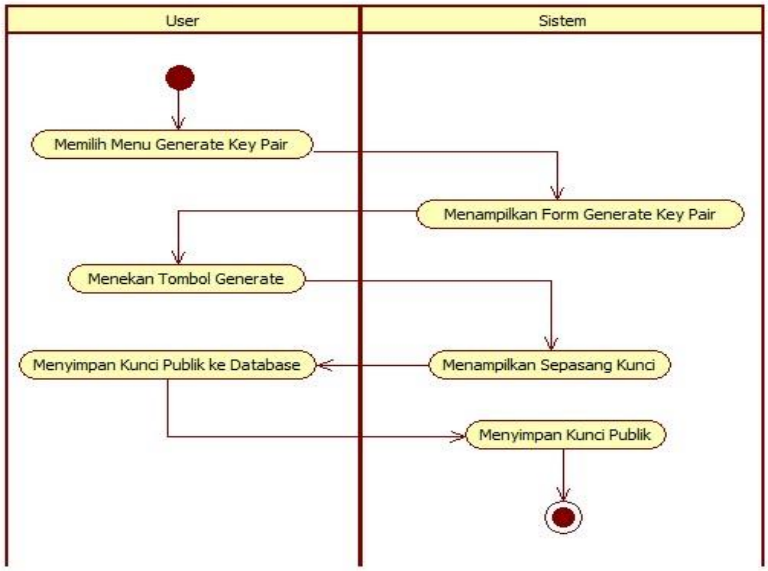
Pada gambar 3.12 menjelaskan *activity diagram* verifikasi *signature*.



Gambar 3. 12 Activity Diagram Verifikasi Pesan

3.1.6.3.9 Key Pair Management

Pada gambar 3.13 menjelaskan *activity diagram key pair management.*



Gambar 3. 13 Activity Diagram Key Pair Management

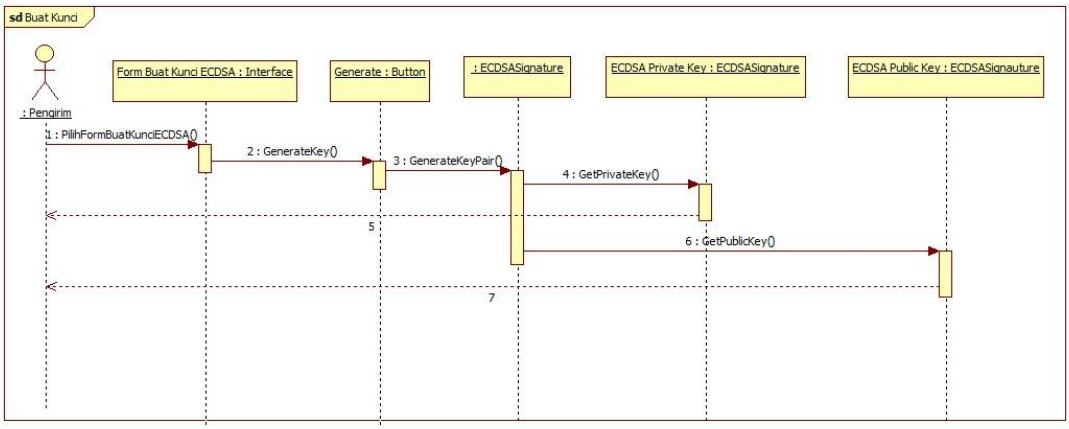
3.1.6.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan diluar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek yang terkait) yang biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* terttentu. Diawali dengan men-*triger* aktifitas tersbut, dan perubahan apa saja yang terjadi secara internal dan *output* yang dihasilkan.

Berikut *sequence diagram* pada aplikasi *digital signature* dan enkripsi untuk keamanan e-dokumen di PT Len Industri.

3.1.6.4.1 *Generate Key Pair Signature*

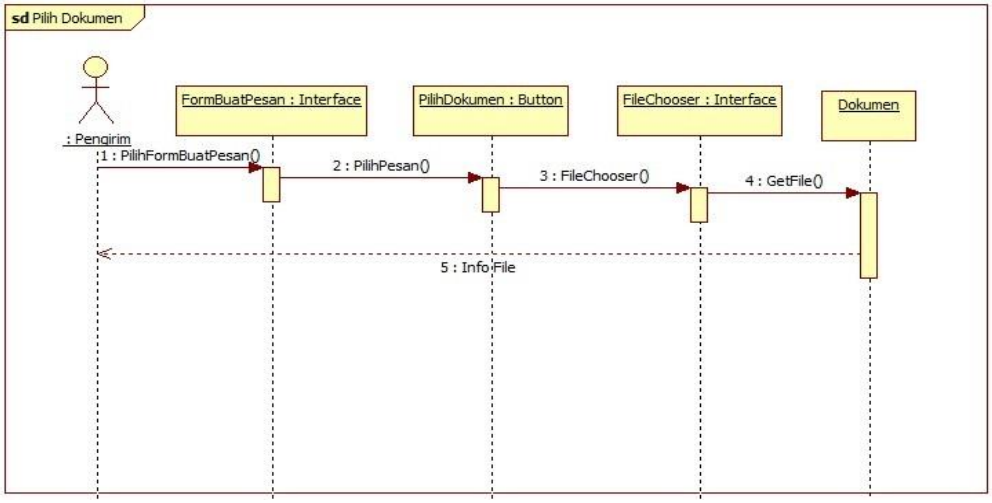
Pada gambar 3.14 merupakan *sequence diagram generate key pair signature.*



Gambar 3. 14 Sequence Diagram Generate Key Pair Signature

3.1.6.4.2 Pilih File

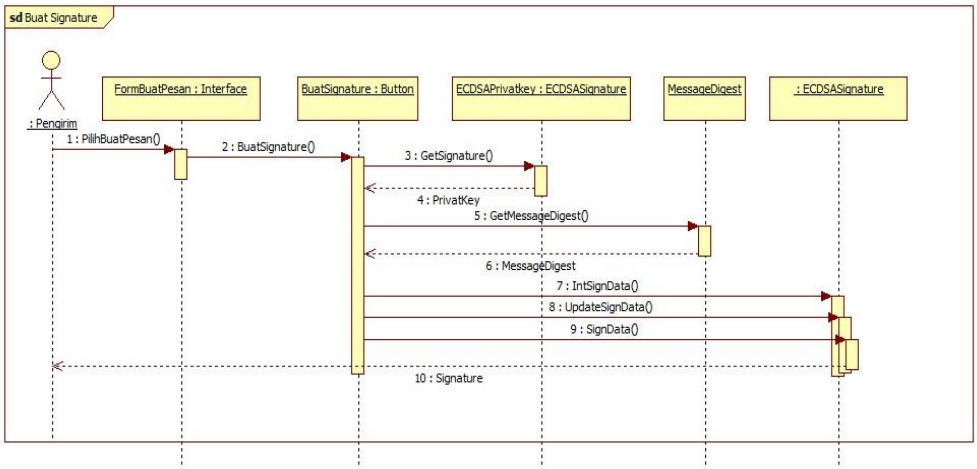
Pada gambar 3.15 merupakan *sequence diagram* pilih *file*.



Gambar 3. 15 Sequence Diagram Pilih File

3.1.6.4.3 Buat *Signature*

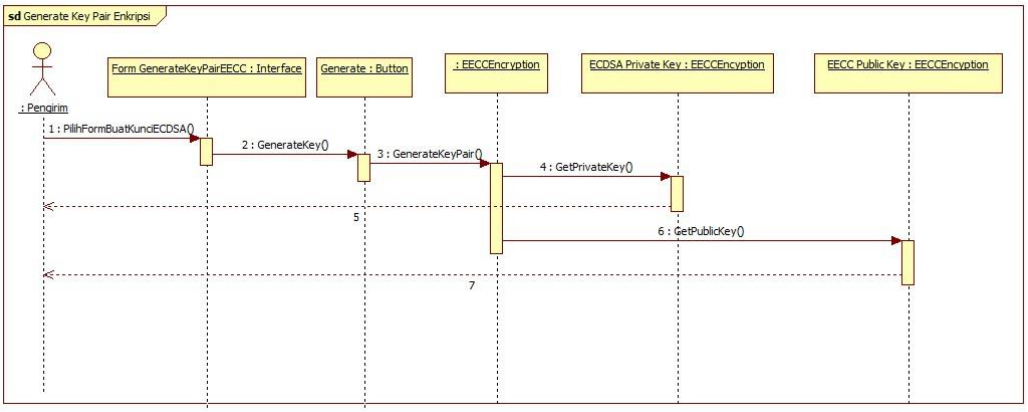
Pada gambar 3.16 merupakan *sequence diagram* buat *signature*



Gambar 3. 16 Sequence Diagram Buat Signature

3.1.6.4.4 *Generate Key Pair* Enkripsi

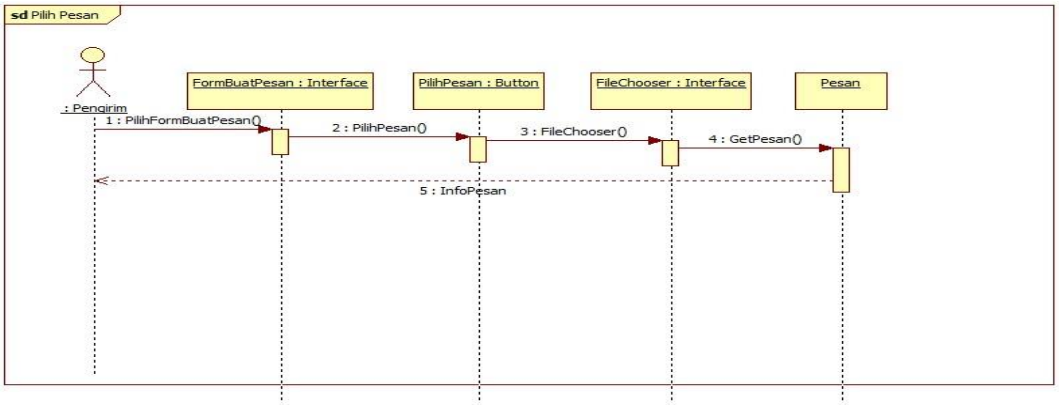
Pada gambar 3.17 merupakan *sequence diagram generate key pair signature.*



Gambar 3. 17 Generate Key Pair Enkripsi

3.1.5.4.5 Pilih Pesan

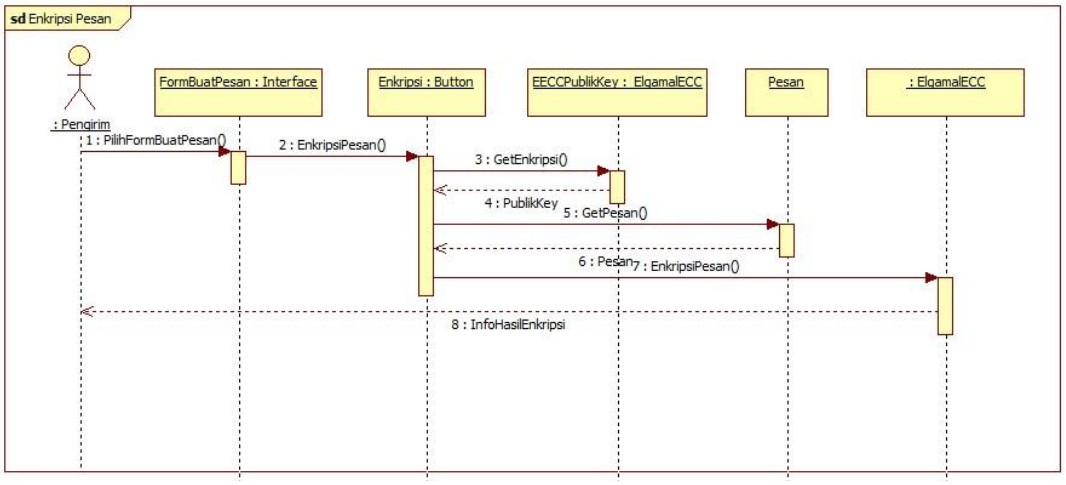
Pada gambar 3.18 merupakan *sequence diagram* pilih pesan.



Gambar 3. 18 Sequence Diagram Pilih Pesan

3.1.6.4.6 Enkripsi Pesan

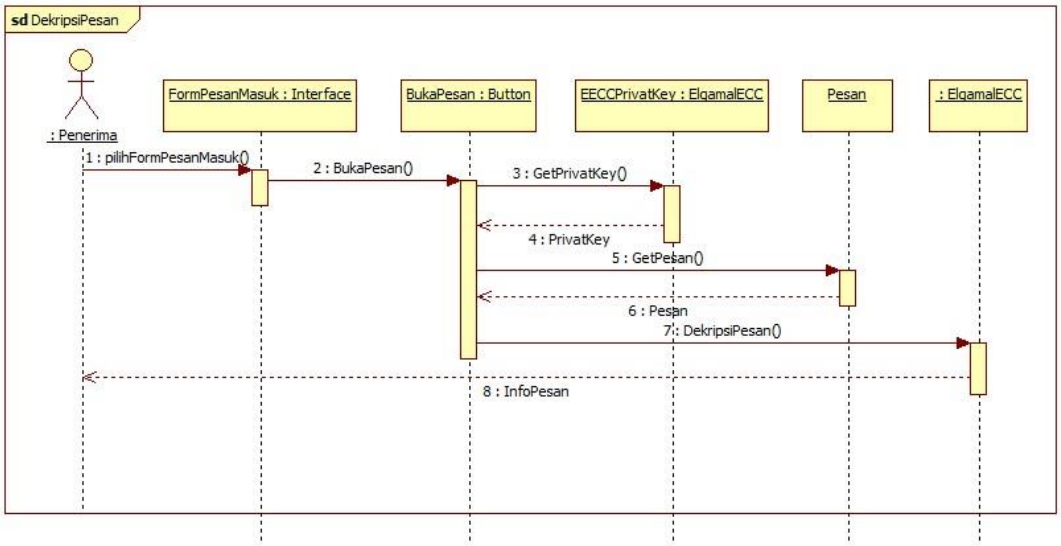
Pada gambar 3.19 merupakan *sequence diagram* enkripsi pesan.



Gambar 3. 19 Sequence Diagram Enkripsi Pesan

3.1.6.4.7 Dekripsi Pesan

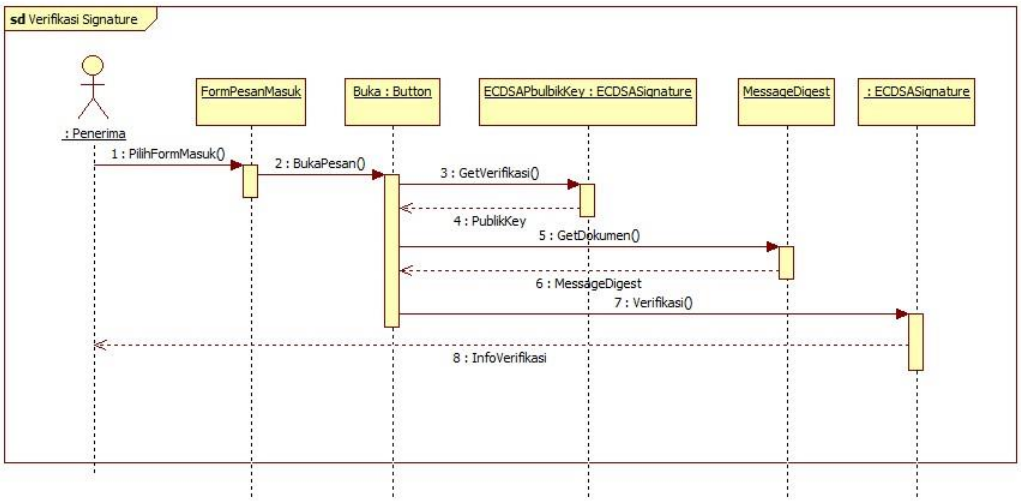
Pada gambar 3.20 merupakan *sequence diagram* dekripsi pesan.



Gambar 3. 20 Sequence Diagram Dekripsi Pesan

3.1.6.4.8 Verifikasi *Signature*

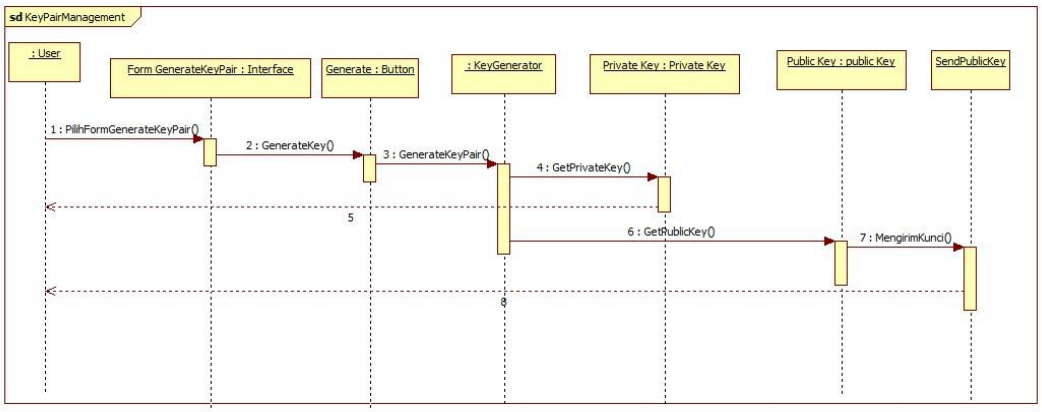
Pada gambar 3.21 merupakan *sequence diagram* verifikasi *signature*



Gambar 3. 21 Sequence Diagram Verifikasi Pesan

3.1.6.4.9 *Key Pair Management*

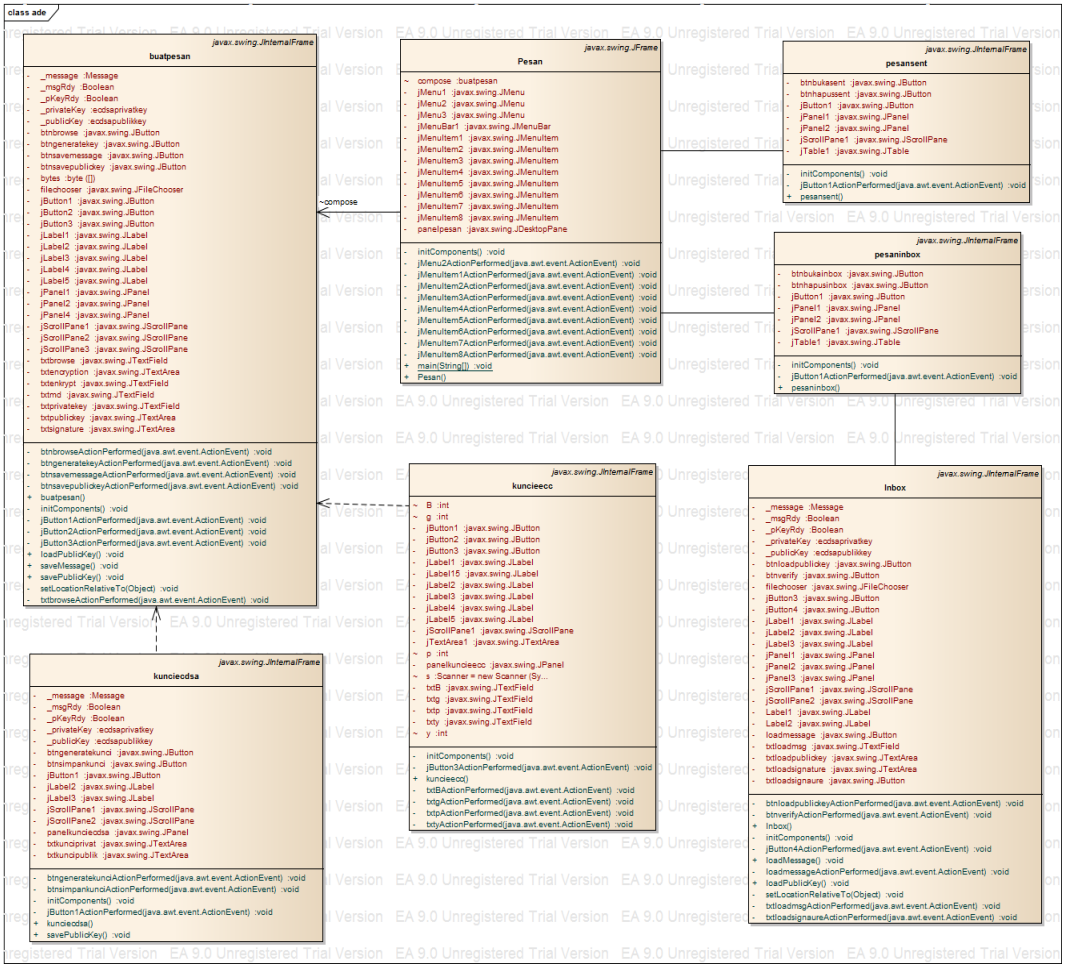
Pada gambar 3.22 merupakan *sequence diagram key pair management*



Gambar 3. 22 Sequence Diagram Key Pair Management

3.1.6.5 Class Diagram

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis. Biasanya, terdiri dari beberapa class diagram untuk sistem tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun. *Class diagram* adalah alat perancangan terbaik untuk tim pengembang. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum proses pengkodean. Berikut *class diagram* pada pembangunan aplikasi *digital signature* dan enkripsi untuk keamanan e-dokumen di PT Len Industri.



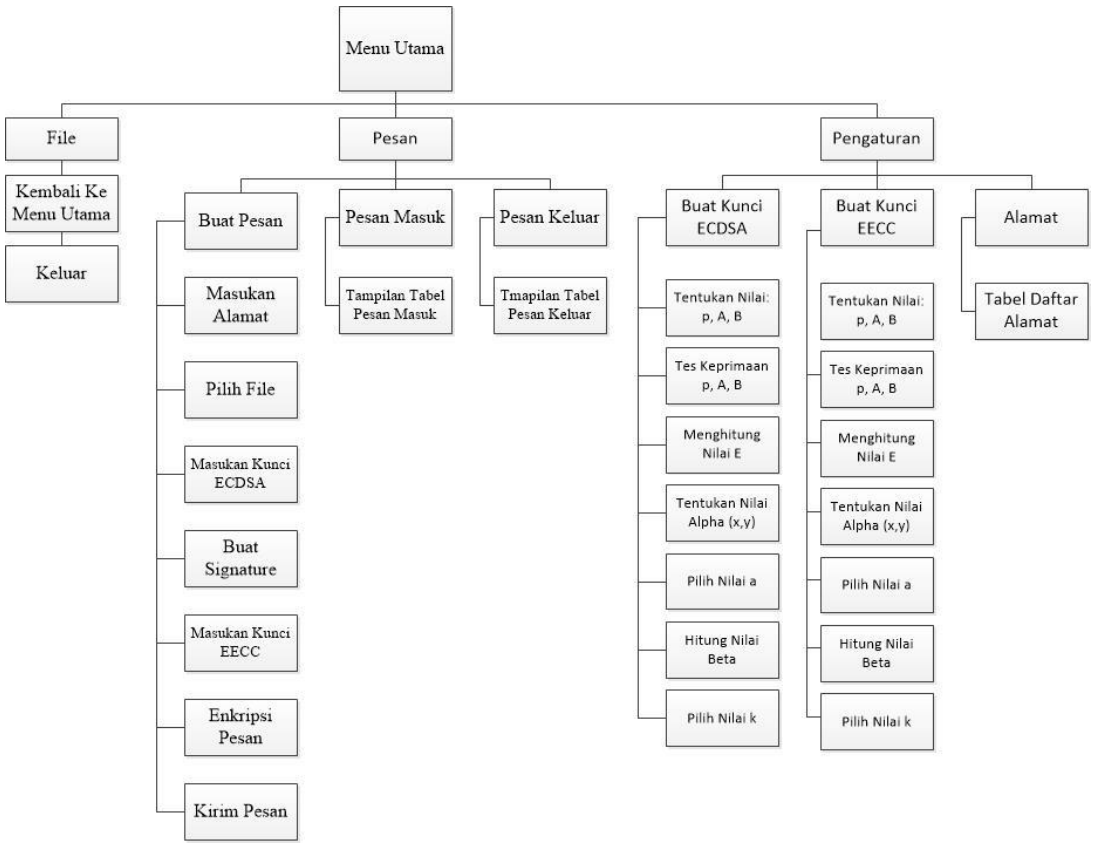
Gambar 3. 23 Class Diagram Aplikasi Digital Signature dan Enkripsi

3.2 Perancangan Sistem

Perancangan sistem merupakan tahapan berikutnya setelah analisis sistem yang menggambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam sebuah bahasa pemrograman. Dalam perancagan sistem tidak lepas dari hasil analisis, karena perancangan sistem dibuat dari hasil analisis sistem.

3.2.1 Perancangan Struktur Menu

Struktur menu pada aplikasi *digital signature* dan enkripsi yang akan dibangun adalah sebagai berikut:



Gambar 3. 24 Struktur Menu Aplikasi Digital Signature dan Enkripsi

3.2.2 Perancangan Antar Muka Perangkat Lunak

Antarmuka merupakan tampilan dari suatu perangkat lunak sebagai sarana antara pengguna dengan perangkat lunak. Perancangan antarmuka ini meliputi penggambaran, perencanaan. Aplikasi yang dibangun diharapkan menyediakan suatu tampilan antarmuka yang mudah difahami oleh pengguna. Berikut tampilan antarmuka dari aplikasi *digital signature* dan enkripsi.

3.2.2.1 Perancangan Antarmuka Halaman Utama

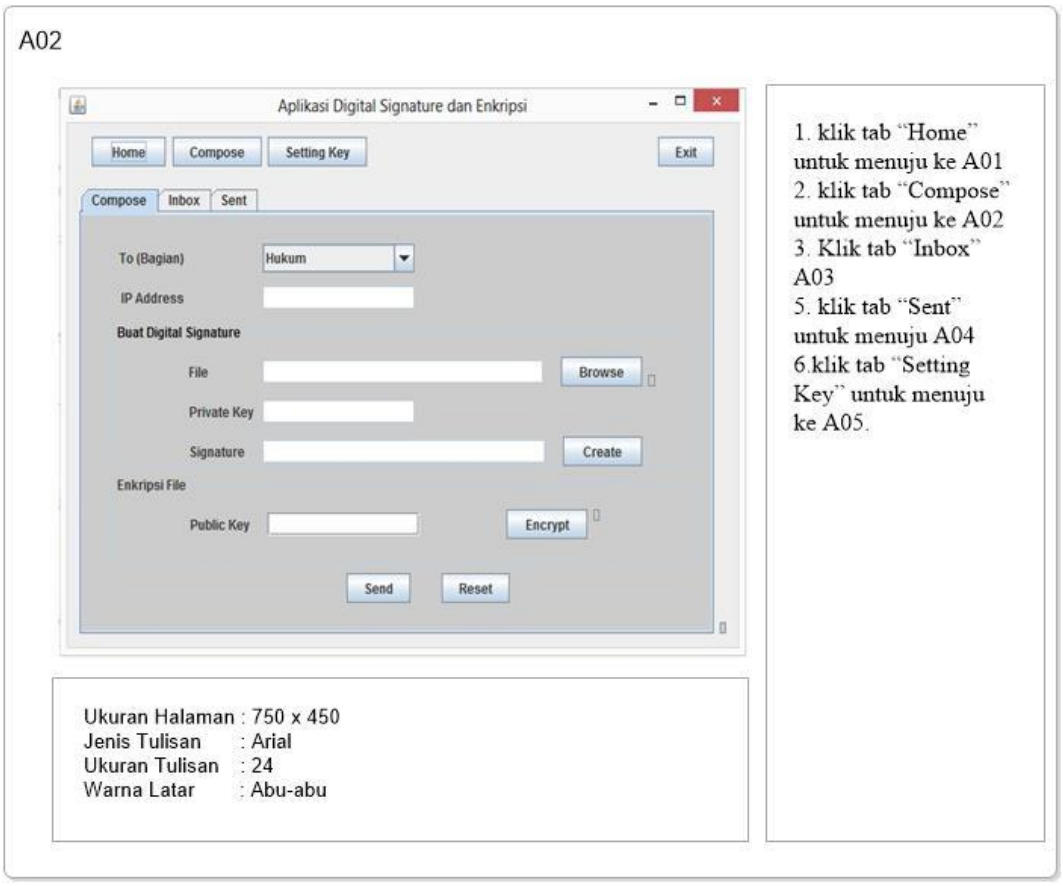
Halaman utama ini merupakan halaman pertama dari aplikasi *digital signature* dan enkripsi ketika aplikasi pertama kali di eksekusi. Berikut tampilan halaman utama:



Gambar 3. 25 Perancangan Antarmuka Halaman Utama

3.2.2.2 Perancangan Antarmuka Buat Pesan

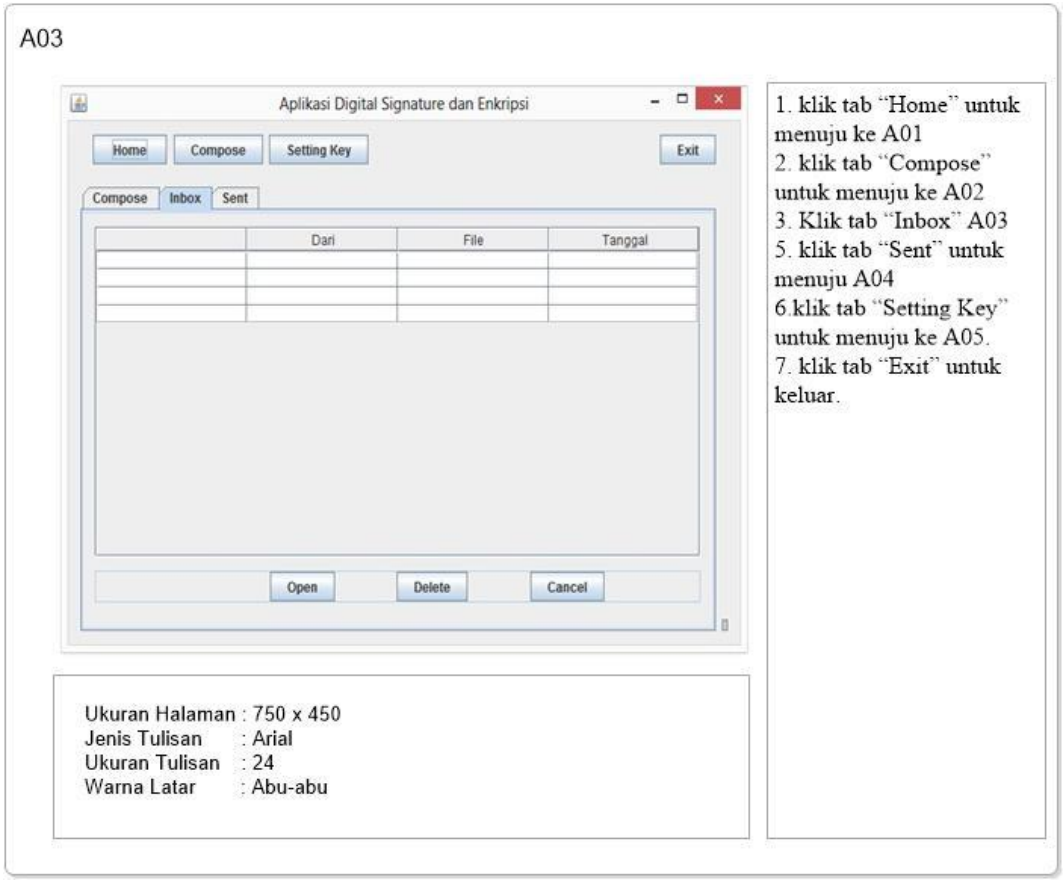
Berikut adalah perancangan antarmuka buat pesan, halaman buat pesan ini akan muncul ketika pengguna memilih tombol ok pada halaman utama dan memilih menu buat pesan pada *form* pesan. Tampilan antarmuka buat pesan terlihat pada gambar 3.26 seperti berikut.



Gambar 3. 26 Perancangan Antarmuka Buat Pesan

3.2.2.3 Perancangan Antarmuka Pesan Masuk

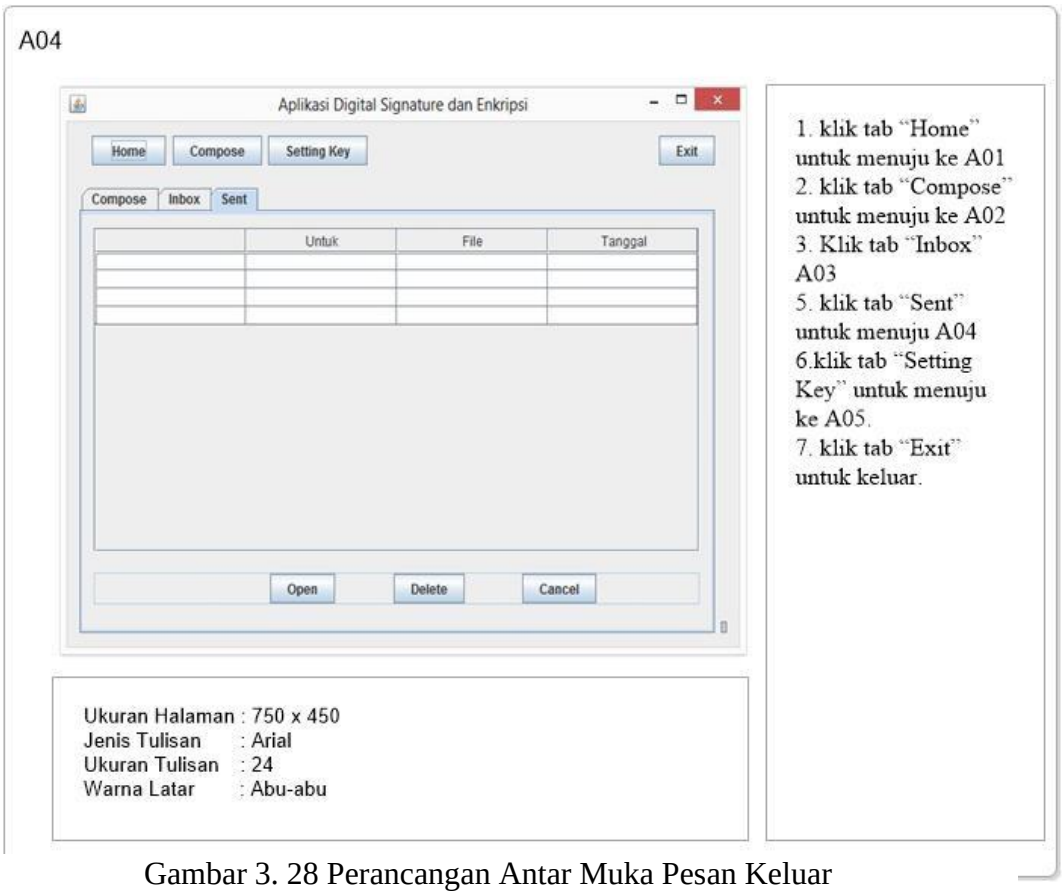
Perancangan antarmuka pesan masuk seperti tampak pada gambar 3.27, *form* pesan masuk akan keluar apabila memilih menu pesan masuk pada menu pesan. Berikut tampilan gambar perancangan pesan masuk:



Gambar 3. 27 Perancagan Antarmuka Pesan Masuk

3.2.2.4 Perancangan Antarmuka Pesan Keluar

Perancangan antarmuka pesan keluar seperti tampak pada gambar 3.29, ketika memilih menu pesan keluar pada menu pesan maka tampilan *form* pesan keluar aadalah sebagai berikut:



Gambar 3. 28 Perancangan Antar Muka Pesan Keluar

3.2.2.5 Perancangan Antarmuka Buat Kunci ECDSA

Buat kunci ECDSA adalah menu untuk membuat kunci yang akan digunakan pada proses *digital signature.* Ketika memilih menu buat kunci ECDSA pada *parent* menu pengaturan, maka tampilannya adalah seperti pada gambar berikut:

3.2.3 Perancangan Pesan Peringatan Kesalahan

Pesan peringatan kesalahan merupakan tampilan yang berfungsi untuk menyampaikan pemberitahuan kepada pengguna apabila terjadi kesalahan penggunaan aplikasi. Berikut perancangan pesan kesalahan pada aplikasi *digital signature* dan enkripsi :

3.2.3.1 Perancangan Pesan Kesalahan Belum Memasukan Kunci

Peringatan ini akan muncul apabila pengguna menekan tombol buat signature dan tombol enkripsi tapi tidak memasukan kunci terlebih dahulu.



Gambar 3. 29 Perancangan Pesan Kesalahan Belum Memasukan Kunci

3.2.3.2 Perancagan Pesan Kesalahan Belum Memilih File

Peringatan ini akan muncul apabila pengguna menekan tombol buat signature dan tombol enkripsi tapi belum memilih *file* terlebih dahulu.



Gambar 3. 30 Perancagan Pesan Kesalahan Belum Memiliha File

3.2.3.3 Perancangan Pesan Kesalahan Gagal Membuat Signature

Peringatan ini akan muncul apabila pengguna gagal membuat signature karena kesalahan memasukan kunci.



Gambar 3. 31 Perancangan Pesan Kesalahan Gagal Membuat Signature

3.2.3.4 Perancangan Pesan Kesalahan Gagal Enkripsi Pesan

Peringatan ini akan muncul apabila pengguna gagal melakukan enkripsi pesan karena pengguna tidak melakukan proses buat *signature* atau kesalahan memasukan kunci.



Gambar 3. 32 Perancagan Pesan Kesalahan Gagal Enkripsi Pesan

3.2.3.5 Perancangan Kesalahan Gagal Kirim Pesan

Peringatan ini akan muncul apabila pengguna menekan tombol kirim pesan tetapi tidak melakukan buat *signature* dan enkripsi terlebih dahulu.



Gambar 3. 33 Perancagan Kesalahan Gagal Kirim Pesan

3.2.3.6 Perancagan Kesalahan Verifikasi Pesan

Pesan ini akan muncul apabila pengguna salah memasukan kunci publik pengirim pesan.



Gambar 3. 34 Perancangan Kesalahan Verifikasi Pesan