BAB II
TINJAUAN TEORI

2.1 Profil Kantor Pelayanan dan Perbendaharaan Negara Kota Sorong

Kantor Pelayanan Perbendaharaan Negara (KPPN) Sorong merupakan instansi vertikal Direktorat Jenderal Perbendaharaan yang berada di bawah Kanwil Ditjen Perbendaharaan Propinsi Papua Barat. Sebagai unit organisasi di bawah payung Kementerian Keuangan maka tugas dan fungsinya berdasar pada keputusan-keputusan yang dikeluarkan oleh Menteri Keuangan.

Berdasarkan Peraturan Menteri Keuangan Nomor 169/PMK.01/2012 tanggal 6 November 2012 tentang Organisasi dan Tata Kerja Instansi Vertikal Direktorat Jenderal Perbendaharaan, Kantor Pelayanan Perbendaharaan Negara mempunyai tugas sebagai berikut:

1. Melaksanakan kewenangan perbendaharaan dan bendahara umum.
2. Penyaluran pembiayaan atas beban anggaran.
3. Penatausahaan penerimaan dan pengeluaran anggaran melalui dan dari kas negara berdasarkan peraturan perundang-undangan.

Adapun fungsi KPPN Sorong merupakan bentuk operasional yang spesifik dari tujuannya adalah sebagai berikut:

1. Pengujian terhadap surat perintah pembayaran berdasarkan peraturan perundang undangan.
2. Penerbitan surat perintah pencairan dana dari kas negara atas nama Menteri Keuangan selaku Bendahara Umum Negara.
3. Penyaluran pembiayaan atas beban APBN.
4. Penilaian dan pengesahan terhadap penggunaan uang yang telah disalurkan.
5. Penatausahaan penerimaan dan pengeluaran negara melalui dan dari kas negara
6. Pengiriman dan penerimaan kiriman uang.
7. Penyusunan laporan pelaksanaan anggaran pendapatan dan belanja negara.
8. Penyusunan laporan realisasi pembiayaan yang berasal dari pinjaman dan hibah luar negeri.
9. Penatausahaan penerimaan negara bukan pajak.

10. Penyelenggaraan verifikasi transaksi keuangan dan akuntansi.
11. Pembuatan tanggapan dan penyelesaian temuan hasil pemeriksaan.
12. Pelaksanaan kehumasan.
13. Pelaksanaan administrasi KPPN.

2.1.1 Sejarah KPPN Sorong

KPPN Sorong pada awalnya dibentuk dengan nama Kantor Bendahara Negara (KBN) setelah itu berganti nama menjadi Kantor Pembendaharaan Negara (KPN) dan Kantor Kas Negara (KKN) pada Tahun 1982. Selanjutnya pada tahun 1990 satuan kerja tersebut diintergrasikan menjadi Kantor Pembendaharaan dan Kas Negara (KPKN).

Dengan diundangkannya Undang-undang Nomor 17 tahun 2003 dan Undang-undang Nomor 1 Tahun 2004, seiring dengan reorganisasi Departemen Keuangan, maka berdasarkan Keputusan Menteri Keuangan RI Nomor : Kep-303/KMK.01/2004 KPKN berubah menjadi KPPN yang merupakan instansi vertikal Direktorat Jenderal Perbendaharaan.

Kantor Pelayanan Perbendaharaan Negara (KPPN) Sorong adalah salah satu instansi vertikal Direktorat Jenderal Perbendaharaan yang di bawah dan bertanggung jawab kepada kepala Kantor Wilayah Provinsi Papua Barat.

Terletak di Provinsi Papua Barat, KPPN Sorong menaungi 118 satuan kerja yang tersebar di satu kota dan lima kabupaten, yaitu : Kota Sorong, Kabupaten Sorong, Kabupaten Sorong Selatan, Kabupaten Raja Ampat, Kabupaten Rat, dan Kabupaten Tambraw.

2.1.2 Visi dan Misi KPPN Sorong

Visi dan misi KPPN Sorong dapat dilihat pada Tabel II-1.

Tabel II-1 Visi dan misi KPPN Sorong

Visi	Menjadi pelaksana fungsi bendahara umum negara di daerah yang profesional, transparan dan akuntabel untuk mewujudkan pelayanan prima.	
Misi	1	Menjamin kelancaran pencairan dana APBN secara tepat sasaran, tepat waktu dan tepat jumlah.
	2	Mengelola penerimaan negara secara profesional dan akuntabel.
	3	Mewujudkan pelaporan pertanggungjawaban APBN yang akurat dan tepat waktu.

2.1.3 Logo KPPN Sorong

Logo KPPN Sorong dapat dilihat pada Gambar II-1.



Gambar II-1 Logo KPPN

Adapun makna dari logo tersebut adalah:

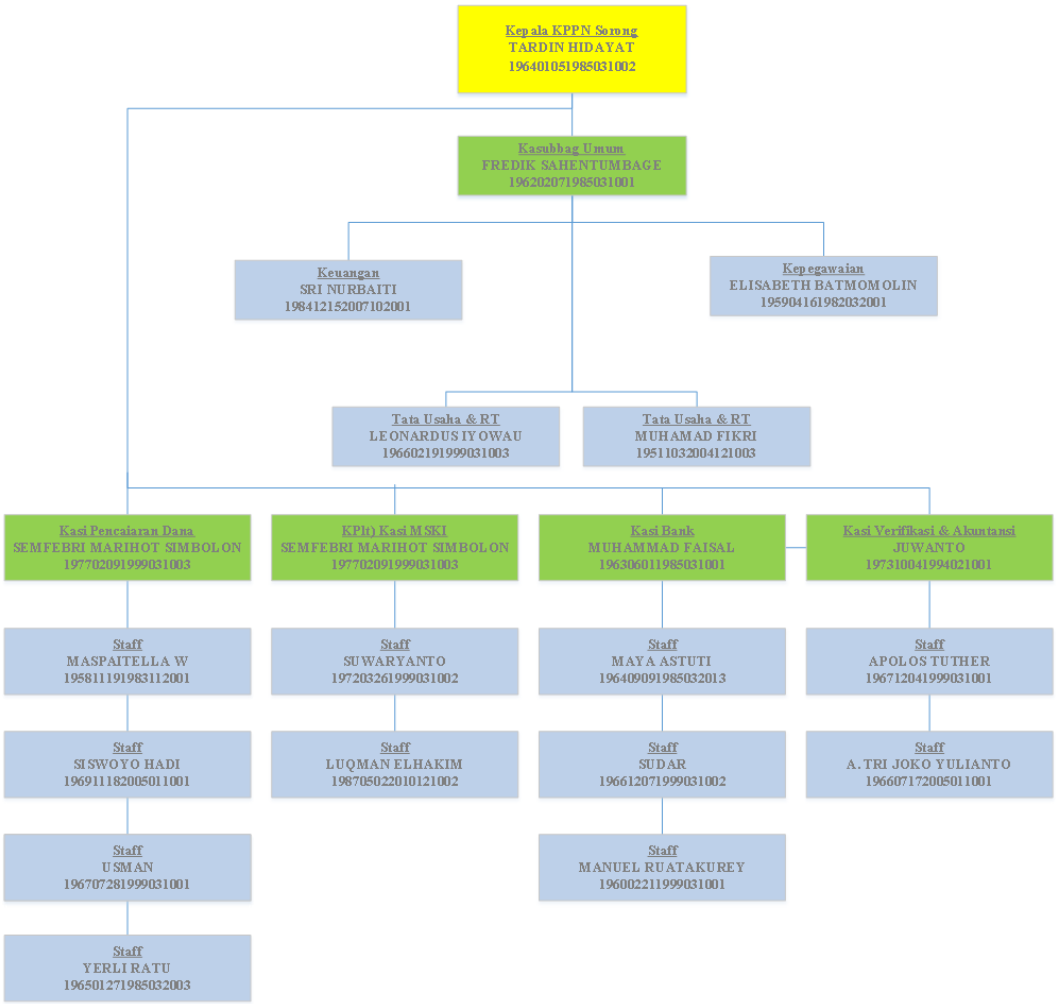
1. Empat buah huruf yang membentuk singkatan “KPPN” dengan kombinasi warna biru muda dan emas dengan outline warna putih.
   a. Huruf K, P dan N berwarna biru muda melambangkan KPPN adalah institusi yang transparan dan akuntabel.
   b. Huruf P berwarna emas melambangkan KPPN sebagai insitusi di bawah Direktorat Jenderal Perbendaharaan merupakan ujung tombak dalam pengelolaan keuangan Negara. Huruf P berwarna emas adalah singkatan dari kata “Perbendaharaan”. Penggunaan warna yang berbeda dengan huruf yang lain adalah merupakan penegasan institusi Ditjen Perbendaharaan.
   c. Outline warna putih melambangkan ketulusan para pegawai KPPN dalam memberikan pelayanan kepada pemangku kepentingan tanpa tidak mengharapkan imbalan.
   d. Outline warna putih dan biru muda bertujuan untuk memberikan karakter yang kuat dari jenis font yang digunakan.
2. Sebuah bangun elips berwarna emas melambangkan KPPN adalah institusi pengelolan keuangan Negara yang memberikan pelayanan yang cepat, sigap, tangkas dan dinamis. Makna ikon secara keseluruhan adalah suatu ungkapan kebulatan tekad KPPN sebagai ujung tombak Direktorat Jenderal Perbendahaan dalam mengelola keuangan Negara untuk memberikan pelayanan yang cepat, akurat, transparan dan akuntabel dengan mengedepankan unsur ketelitian,

kehati-hatian dan ketulusan dalam rangka mewujudkan Visi dan Misi Ditjen Perbendaharaan.

2.1.4 Struktur KPPN Organisasi

Struktur organisasi dari KPPN Sorong terdiri dari Kepala Kantor, Subbagian Umum, Seksi Pencairan Dana, Seksi Bank, Seksi Manajemen Satuan Kerja dan Kepatuhan Internal (MSKI) dan Seksi Verifikasi dan Akuntansi (VERA).

Ilustrasi struktur organisasi Kantor Pelayanan dan Perbendaharaan Negara Sorong dapat dilihat pada Gambar II-2.



Gambar II-2 Struktur organisasi KPPN Sorong

2.2 Landasan Teori

Landasan teori akan disajikan teori-teori yang menunjang dilakukannya penelitian.

2.2.1 Digital Signature

Digital signature merupakan salah satu mekanisme keamanan yang dapat dilakukan untuk mewujudkan layanan keamanan. Layanan keamanan yang dapat diwujudkan oleh *digital signature* diantaranya adalah otentikasi pesan dan keutuhan data [1].

2.2.1.1 Otentikasi Pesan

Layanan keamanan otentikasi pesan dapat diwujudkan oleh mekanisme *digital signature* dengan ilustrasi sebagai berikut:

1. A hendak mengirim pesan *M* ke B. A sebelumnya telah menyiapkan sepasang kunci private $K_{private}$ dan kunci publik K_{publik}.
2. A membuat tanda tangan digital *s* untuk pesan *M* melalui proses *sign* sebagai berikut:

$$s = sign(M, K_{private})$$

3. Kemudian A mengirim pesan *M* dan tanda tangan digital *s* pada B.
4. B dapat melakukan otentikasi pesan *M* menggunakan kunci publik K_{publik} dan tanda tangan digital *s* milik A dengan cara proses *verify* sebegai berikut:

$$verify(M, K_{publik}, s)$$

5. Jika hasil proses *verify* bernilai *true,* B mendapat kepastian bahwa tanda tangan digital *s* adalah milik A pada pesan *M* dan pesan *M* berasal dari A.

2.2.1.2 Keutuhan Data

Mekanisme *digital signature* dapat mewujudkan layanan keamanan keutuhan data dengan ilustrasi sebagai berikut:

1. A hendak mengirim pesan *M* ke B. A sebelumnya telah menyiapkan sepasang kunci private $K_{private}$ dan kunci publik K_{publik}.
2. A membuat tanda tangan digital *s* untuk sidik pesan *h(M)* melalui proses *sign* sebagai berikut:

$$s = sign(h(M), K_{private})$$

3. Kemudian A mengirim pesan *M* dan tanda tangan digital *s* pada B.
4. B dapat mengetahui keutuhan data *M* menggunakan kunci publik K_{publik} dan tanda tangan digital *s* milik A dengan cara proses *verify* sebagai berikut:

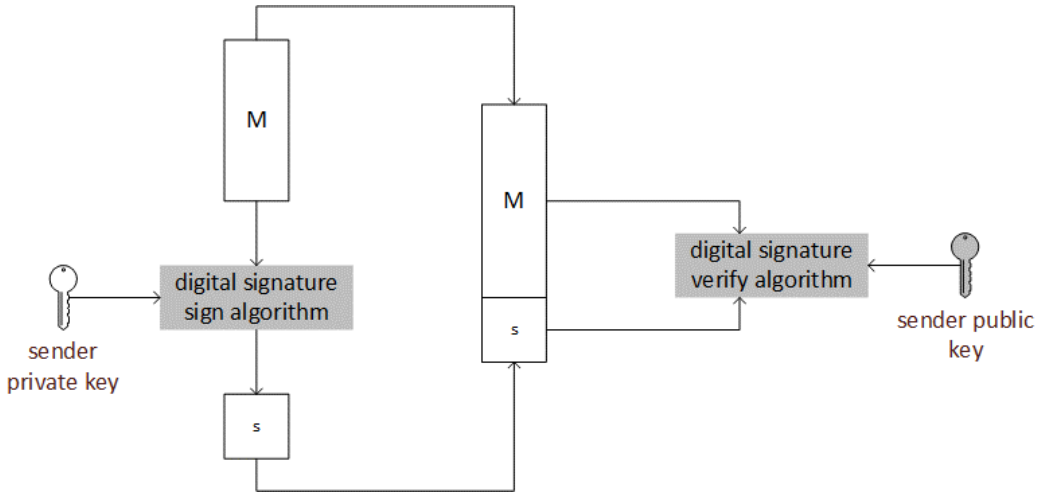
$$verify(h(M), K_{publik}, s)$$

5. Jika hasil proses *verify* bernilai *true*, B mendapat kepastian bahwa pesan *M* dari A utuh.

2.2.1.3 Mekanisme Digital Signature

Pada mekanisme keamanan *digital signature* terdapat dua proses, yaitu proses *sign* dan proses *verify*. Proses *sign* merupakan proses pembuatan tanda tangan digital *s* pada dokumen *M* menggunakan kunci private $K_{private}$ pengirim. Proses sign memiliki masukan dokumen *M*, kunci private $K_{private}$ dan memiliki keluaran *s* sebagai tanda tangan digital. Sedangkan proses *verify* merupakan proses yang menghasilkan nilai *true* atau *false* pada tanda tangan digital *s* milik pengirim pada suatu dokumen *M*. Proses *verify* memiliki masukan dokumen *M*, kunci publik K_{publik} dan tanda tangan digital *s*.

Mekanisme keamanan *digital signature* dapat dilihat pada Gambar II-3.



Gambar II-3 Mekanisme keamanan digital signature

2.2.2 Digital Signature Algorithm (DSA)

Digital signature algorithm (DSA) pertama kali dipublikasikan *National Institute of Standard and Technology* (NIST) pada tahun 1991. DSA terdiri dari algoritma *sign* dan algoritma *verify* [3].

DSA memiliki beberapa parameter yang digunakan pada algoritma *sign* maupun algoritma *verify*. Parameter tersebut dikategorikan menjadi g*lobal public key component, private key, public key, per-message secret key*. Keterangan dari masing-masing kategori parameter dapat dilihat pada Tabel II-2.

Tabel II-2 Parameter DSA

Kategori	Parameter	Keterangan
Global Public Key Component	*p*	Bilangan prima
	q	Bilangan prima yang membagi habis *p-1*
	h	$1 < h < (p-1)$
	g	$g = h^{(p-1)/q} \bmod p$
Private key	*x*	$0 < x < q$
Public key	*y*	$y = g^x \bmod q$
Per-message secret key	*k*	$0 < k < q$

2.2.2.1 Algoritma Sign DSA

Algoritma *sign* DSA menghasilkan tanda tangan digital untuk suatu dokumen *M* berupa entitas *r* dan entitas *s*. Kalkulasi entitas *r* dan entitas *s* dapat dlihat pada Tabel II-3.

Tabel II-3 Algoritma sign DSA

Algortima Sign DSA
$r = (g^k \bmod p) \bmod q$
$s = [k^{-1} (h(M) + xr)] \bmod q$

2.2.2.2 Algoritma Verify DSA

Algoritma *verify* DSA memiliki beberapa kalkulasi seperti pada Tabel II-4.

Tabel II-4 Algoritma verify DSA

Algoritma Verify DSA		
w	=	$(s)^{-1} \bmod q$
u_1	=	$[h(M)w] \bmod q$
u_2	=	$(r)w \bmod q$
v	=	$[(g^{u1} y^{u2}) \bmod p] \bmod q$

Jika $v = r$ maka tanda tangan digital r dan s pada suatu dokumen M dapat dikatakan valid.

2.2.3 Kriptografi

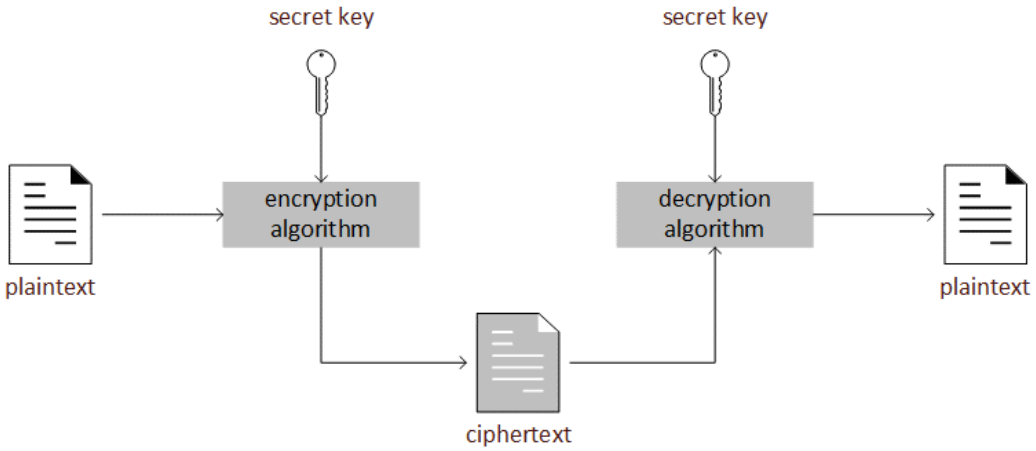
Kriptografi merupakan salah satu mekanisme keamanan untuk mewujudkan layanan keamanan kerahasian data. Kriptografi terbagi menjadi kriptografi simetri dan kriptografi asimetri.

2.2.3.1 Kriptografi Simetri

Kriptografi simetri merupakan teknik kriptografi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. Terdapat beberapa istilah pada kriptografi simetri, diantaranya:

1. *Plaintext*, merupakan pesan asli yang akan di enkripsi menggunakan suatu *encryption algorithm.*
2. *Encryption algorithm*, merupakan metode kriptografi yang digunakan untuk melakukan enkripsi.
3. *Secret key*, merupakan kunci yang digunakan saat enkripsi maupun dekripsi pada kriptografi simetri.
4. *Ciphertext*, merupakan bentuk pesan asli yang telah di enkripsi menggunakan suatu *encryption algorithm.*
5. *Decryption algorithm*, merupakan metode kriptografi yang digunakan untuk melakukan dekripsi.

Mekanisme kriptografi simetri dapat dilihat pada Gambar II-4.



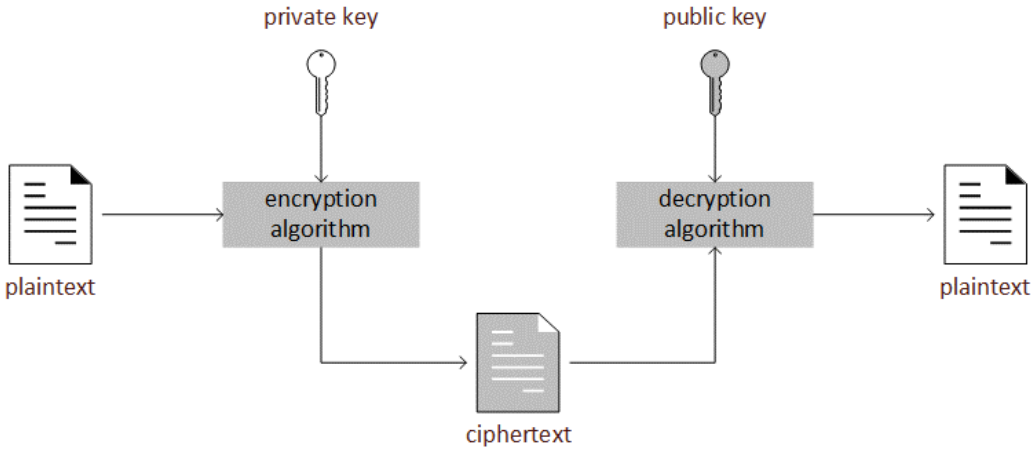
Gambar II-4 Mekanisme keamanan kriptografi simetri

2.2.3.2 Kriptografi Asimetri

Kriptografi asimetri merupakan teknik kriptografi yang menggunakan sepasang kunci, yaitu kunci *private* dan kunci publik, untuk proses enkripsi dan dekripsi. Proses enkripsi menggunakan kunci *private* sedangkan proses dekripsi menggunakan kunci publik. Terdapat beberapa istilah pada kriptografi asimetri, diantaranya:

1. *Plaintext*, merupakan pesan asli yang akan di enkripsi menggunakan suatu *encryption algorithm*.
2. *Encryption algorithm*, merupakan metode kriptografi yang digunakan untuk melakukan enkripsi.
3. *Public and Private Key*, merupakan sepasang kunci yang digunakan saat enkripsi maupun dekripsi pada kriptografi asimetri. Kunci *private* digunakan saat enkripsi sedangkan kunci publik digunakan saat dekripsi.
4. *Ciphertext*, merupakan bentuk pesan asli yang telah di enkripsi menggunakan suatu *encryption algorithm*.
5. *Decryption algorithm*, merupakan metode kriptografi yang digunakan untuk melakukan dekripsi.

Mekanisme kriptografi asimetri dapat dilihat pada Gambar II-5.



Gambar II-5 Mekanisme keamanan kriptografi asimetri

2.2.4 Advanced Encryption Standard (AES)

National Institute of Standard and Technology (NIST) membuat sayembara untuk menggantikan *Data Encryption Standard* (DES) dengan sebuah sistem penyandian yang disebut *Advanced Encryption Standard* (AES) pada tanggal 12 September 1997. Setelah beberapa seleksi, NIST memilih sistem penyandian *Rijndael* yang dikembangkan oleh *Joan Daemen* dan *Vincent Rijment* sebagai sistem penyandian AES pada tahun 2000 [1].

Algoritma AES memiliki beberapa unit ukuran data seperti pada Tabel II-5.

Tabel II-5 Unit ukuran data

Unit	Keterangan
BIT	Satuan data terkecil (biner)
BYTE	1 *byte* = 8 *bit*
WORD	1 *word* = 4 *byte* = 32 *bit*
BLOK	1 *blok* = 4 *word* = 16 *byte* = 128 *bit*
STATE	Blok yang ditata sebagai matrik beukuran 4 x 4

Kunci AES dapat memiliki panjang kunci 128, 192, 256 bit. Penyandian AES menggunakan proses berulang yang disebut dengan ronde. Jumlah ronde tergantung dengan panjang kunci yang digunakan. Relasi antara jumlah ronde dengan panjang kunci dapat dilihat pada Tabel II-6.

Tabel II-6 Relasi panjang kunci AES dengan jumlah ronde

Panjang Kunci AES	Jumlah Ronde
128 bit	10

Panjang Kunci AES	Jumlah Ronde
192 bit	12
256 bit	14

Untuk melakukan proses enkripsi dan dekripsi, algoritma AES membutuh kunci dengan panjang 44 *word*, dimana 4 *word* digunakan pada transformasi *add-round-key* sebelum proses enkripsi atau dekripsi dilakukan dan 40 *word* yang digunakan pada proses enkripsi maupun dekripsi. Sehingga untuk menghasilkan kunci dengan panjang 44 *word* diperlukan suatu proses ekspansi kunci.

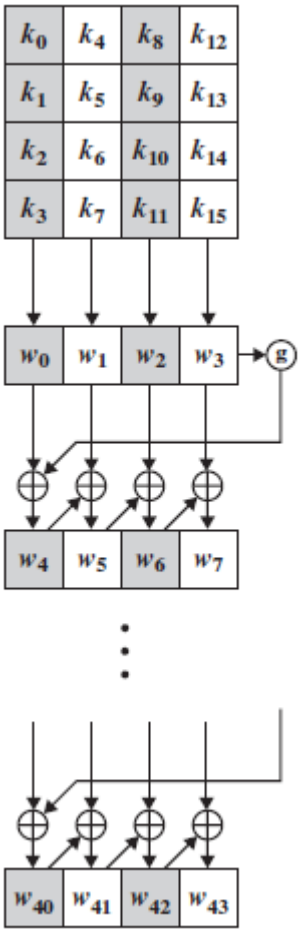
Proses ekspansi kunci memiliki masukan kunci inisial dengan panjang 4 *word*. Pada algoritma AES dengan panjang 128 bit, proses ekspansi menghasilkan 44 *word* kunci dengan melakukan beberapa operasi selama 10 iterasi. Adapun operasi yang dilakukan disetiap iterasi adalah sebagai berikut:

1. *Rot-Word*, merupakan operasi *circular left shif*t pada *word* terakhir suatu *state*.
2. *Sub-Word*, melakukan substitusi setiap elemen *word* dengan S-Box (Tabel II-8).
3. *Word* hasil dari operasi *sub-word* dilakukan operasi XOR dengan *round-constant* seperti pada Tabel II-7.

Tabel II-7 Round constant proses ekpansi kunci

i	1	2	3	4	5	6	7	8	9	10
RCon[i]	01	02	04	08	10	20	40	80	1B	36

Secara keseluruhan, proses ekspansi dapat dilihat seperti pada Gambar II-6**.**



Gambar II-6 Proses ekspansi kunci

Algortima enkripsi AES dengan jumlah ronde sebanyak 10 ronde menggunakan 4 transformasi, yaitu *Sub-Bytes, Shift-Row, Mix-Column dan Add-Round-Key* dan transformasi *Add-Round-Key* yang dilakukan sebelum ke-10 proses enkripsi. Pada ronde 1 – 9, algoritma enkripsi AES melakukan semua transformasi secara berurutan sedangkan pada ronde 10 transformasi yang dilakukan adalah *Sub-Bytes, Shift-Row* dan *Add-Round-Key*.

Algoritma dekripsi AES dengan jumlah ronde sebanyak 10 ronde menggunakan 4 transformasi, yaitu *Inverse Shift-Row, Inverse Sub-Byte, Inverse Mix-Column dan Add-Round-Key* dan transformasi *Add-Round-Key* yang dilakukan sebelum proses dekripsi. Pada ronde 1 – 9, algoritma dekripsi AES melakukan semua transformasi sedangkan pada ronde 10 transformasi yang dilakukan adalah *Inverse Shift-Row, Inverse Sub-Byte* dan *Add-Round-Key*.

2.2.4.1 Transformasi Sub-Byte dan Inverse Sub-Byte

Transformasi *Sub-Byte* dan *Inverse Sub-Bytes* merupakan transformasi yang mengganti nilai elemen *state* dengan nilai pada tabel substitusi. 4 bit paling kiri dari nilai elemen *state* digunakan untuk mengacu indeks baris pada tabel substitusi sedangkan 4 bit paling kanan dari nilai elemen *state* digunakan untuk mengacu indeks kolom tabel substitusi.

Terdapat 2 tabel substitusi yang digunakan pada algoritma AES, yaitu tabel S-Box seperti pada Tabel II-8 untuk transformasi *Sub-Byte* dan Inverse S-Box pada Tabel II-9 untuk transformasi *Inverse Sub-Byte*.

Tabel II-8 S-Box

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Tabel II-9 Inverse S-Box

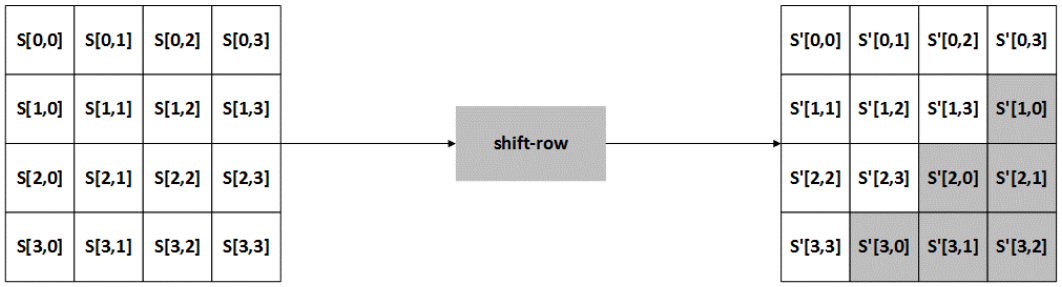
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF

E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

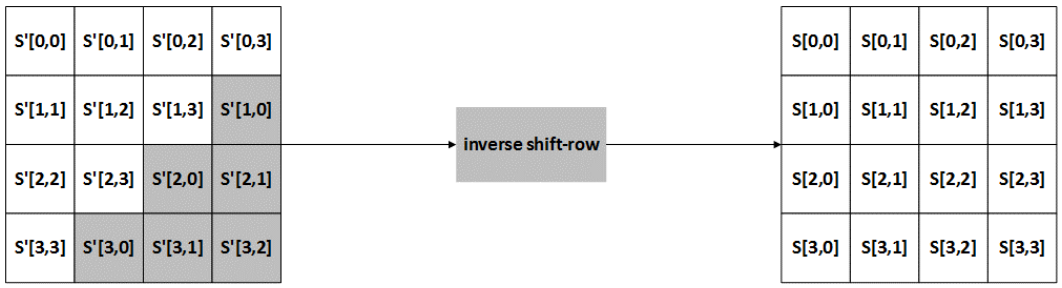
2.2.4.2 Transformasi Shift-Row dan Inverse Shift-Row

Transformasi *Shift-Row* dan *Inverse Shift-Row* merupakan jenis transformasi permutasi, yaitu pengubahan posisi elemen *state* tanpa mengubah nilanya.

Transformasi *Shift-Row* melakukan operasi *circular 1 left bit* sedangkan transformasi *Inverse Shift-Row* melakukan operasi *circular 1 right bit*. Ilustrasi transformasi *Shift-Row* dapat dilihat pada Gambar II-7 sedangkan transformasi *Inverse Shift-Row* dapat diilustrasikan seperti pada Gambar II-8.



Gambar II-7 Transformasi shift-row



Gambar II-8 Transformasi inverse shift-row

2.2.4.3 Transformasi Mix-Column dan Inverse Mix-Column

Transformasi *Mix-Column* dan *Inverse Mix-Column* melakukan perkalian antara matrik konstan dengan *state*. Operasi perkalian dilakukan antara elemen matrik konstan pada baris ke-i dengan elemen *state* pada kolom ke-j dengan menggunakan *irreducible polynominal* $x^8 + x^4 + x^3 + x + 1$.

Adapun operasi perkalian transformasi *Mix-Column* adalah sebagai berikut:

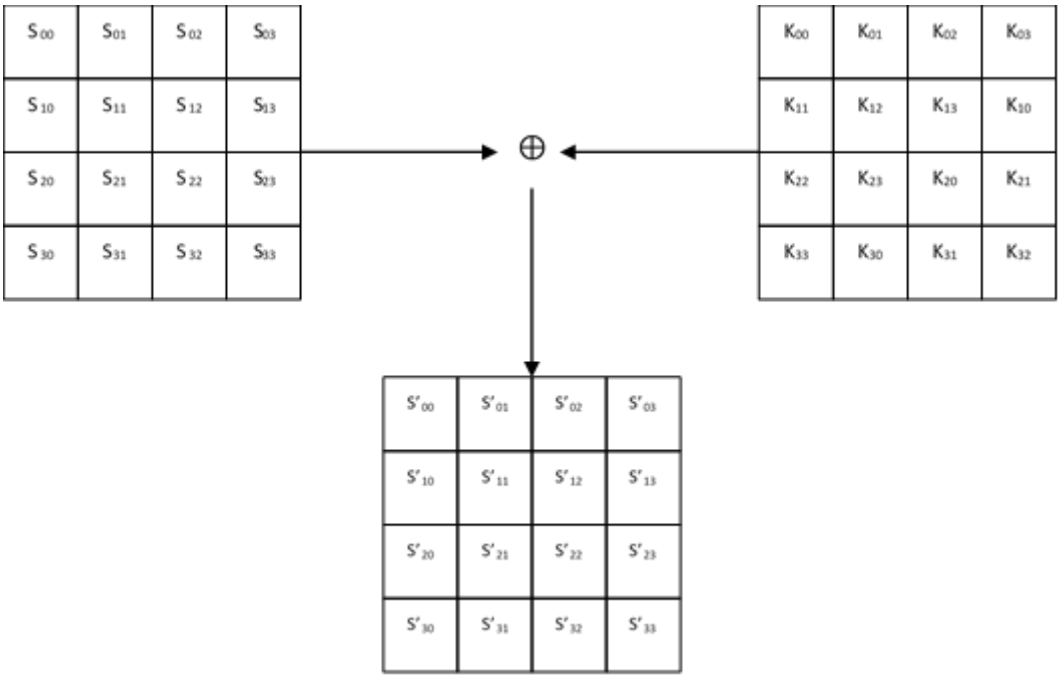
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} + \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

Sedangkan transformasi Inverse Mix-Column adalah sebagai berikut:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} + \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}$$

2.2.4.4 Transformasi Add-Round-Key

Transformasi *Add-Round-Key* merupakan operasi XOR antara *state* dengan kunci ronde. Ilustrasi dapat dilihat pada Gambar II-9**.**

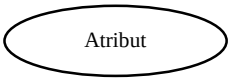


Gambar II-9 Tranformasi add-round-key

2.2.5 Entity Relationship Diagram (ERD)

ERD adalah salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database.* Adapun beberapa notasi grafik yang digunakan dalam ERD dapat dilihat pada Tabel II-10**.**

Tabel II-10 Notasi grafik ERD

Komponen	Keterangan
	Persegi panjang mewakili entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

2.2.5.1 Primary Key

Primary key adalah salah satu atribut yang dapat dijadikan sebagai kunci utama pada entitas. *Primary key* yang baik memiliki kriteria sebgai berikut [4]:

1. Bersifat *mandatory* (tidak kosong atau null)
2. Bersifat unik
3. Nilai-nilainya lebih permanen
4. Berukuran kecil

2.2.5.2 Foreign Key

Jika sebuah *primary key* terhubung ke tabel atau entitas lain, maka keberadaan *primary key* pada entitas tersebut disebut sebagai *foreign key*.

2.2.5.3 Kardinalitas

Dalam ERD relasi dapat terdiri dari sejumlah entitas yang disebut dengan derajat relasi, derajat relasi maksimum disebut dengan kardinalitas sedangkan derajat minimum disebut dengan modalitas. Kardinalitas relasi menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas relasi terjadi antara dua himpunan entitas, dapat berupa:

1. *One-to-one* yang menandakan setiap entitas pada suatu himpunan entitas dapat berelasi dengan paling banyak 1 entitas pada suatu himpunan entitas lain demikian juga sebaliknya.
2. *One-to-many* yang menandakan setiap entitas pada suatu himpunan entitas dapat berelasi dengan banyak entitas pada suatu himpunan entitas lain namun tidak sebaliknya.
3. *Many-to-one* yang menandakan setiap entitas pada suatu himpunan entitas dapat berelasi dengan banyak entitas pada suatu himpunan entitas lain namun tidak sebaliknya.
4. *Many-to-many* yang menandakan setiap entitas pada suatu himpunan entitas dapat berelasi dengan banyak entitas pada suatu himpunan entitas lain demikian pula sebaliknya.

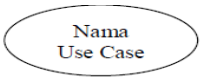
2.2.6 Unified Modeling Languange (UML)

Unified Modeling Language (UML) merupakan *tool* untuk analisis dan perancangan pengembangan perangkat lunak berorientasi objek [5]. UML menyediakan beberapa jenis diagram, diantaranya *usecase diagram, activity diagram, class diagram* dan *sequence diagram.*

2.2.5.1 Usecase Diagram

Usecase adalah gambaran grafik dari beberapa atau semua aktor *usecase* dan interaksi diantara komponem-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Tabel II-11 merupakan beberapa notasi grafik yang terdapat dalam *usecase diagram*.

Tabel II-11 Notasi grafik usecase diagram

Notasi	Keterangan	Simbol
Aktor	Aktor adalah pengguna sistem. Aktor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai aktor.	
Usecase	*Usecase* digambarkan sebagai lingkaran elips dengan nama *usecase* dituliskan didalam elips tersebut.	
Asosiasi	Asosiasi digunakan untuk menghubungkan aktor dengan *usecase*. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara aktor dengan *usecase*.	

2.2.5.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan *worlflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, tetapi aktivitas yang dapat dilakukan oleh sistem. Tabel II-12 merupakan beberapa notasi grafik dalam *activity diagram*.

Tabel II-12 Notasi grafik activity diagram

Notasi	Keterangan	Simbol
Initial	Titik awal, untuk memulai suatu aktivitas	
Final	Titik akhir, untuk mengakhiri aktivitas	
Activity	Menandakan sebuah aktifitas	
Decision	Pilihan untuk mengambil keputusan	
Fork / Join	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.	

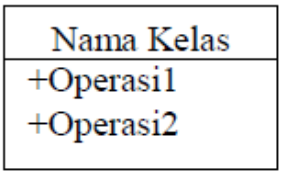
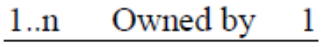
Notasi	Keterangan	Simbol
Flow Final	Untuk mengakhiri suatu aliran	

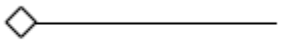
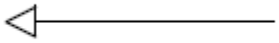
2.2.5.3 Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, kelas memiliki 3 bagian utama yaitu *attribute, operation,* dan *name.*

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Tabel II-13 merupakan notasi grafik pada class diagram.

Tabel II-13 Notasi grafik class diagram

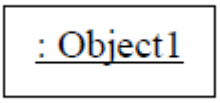
Notasi	Keterangan	Simbol
Class	*Class* adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class.* Bagian tengah mendefinisikan *property* atau atribut *class*. Bagian akhir mendefinisikan *method* dari sebuah *class*.	
Assosiation	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 *class*, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship*.	
Composition	Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi.	
Dependency	Kadangkala sebuah *class* enggunakan *class* yang lain. Hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain.	

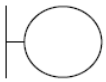
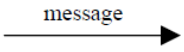
Notasi	Keterangan	Simbol
	Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik.	
Aggregation	*Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi "mempunyai sebuah" atau "bagian dari". Sebuah *aggregation* digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi.	
Generalization	Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasi objek. Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas induknya.	

2.2.5.4 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi antara objek-objek tersebut. Diagram ini juga menunjukan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan. Aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Pada Tabel II-14 disajikan notasi grafik class diagram.

Tabel II-14 Notasi grafik sequence diagram

Notasi	Keterangan	Simbol
Object	*Object* merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah *class* dengan nama objek didalamnya yang diawali dengan sebuah titik koma	
Actor	*Actor* juga dapat berkomunikasi dengan *object*, maka *actor* juga dapat diurutkan sebagai kolom.	
Lifeline	*Lifeline* mengindikasikan keberadaan sebuah objek dalam basis waktu. Notasi untuk *Lifeline* adalah garis putus-putus vertikal yang ditarik dari sebuah objek.	
Activation	*Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline*. *Activation* mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	

Notasi	Keterangan	Simbol
Boundary	*Boundary* terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori *boundary*.	
Control	*Control* berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.	
Entity	*Entity* digunakan menangani informasi yang mungkin akan disimpan secara permanen. *Entity* dapat juga merupakan sebuah tabel pada struktur basis data.	
Massage	*Message* digambarkan dengan anak panah horizontal antara *activation*. *Message* mengindikasikan komunikasi antara objek.	
Self -Message	*Self-message* mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.	
Loop	Operator *loop* adalah *fragmen* yang dapat mengeksekusi berulang kali dan *message loop* penjaga yang menunjukkan dasar iterasi.	