BAB II
TINJAUAN PUSTAKA

2.1. Landasan Teori

Dasar Negara Negara Republik Indonesia dituangkan dalam UUD Tahun 1945 dijadikan landasan dalam menjalankan kehidupan berbangsa dan bernegara. Salah satu tujuan utama dibentuknya Negara Indonesia yaitu melindungi segenap bangsa Indonesia dan seluruh tumpah darah Indonesia. Negara harus melindungi seluruh tumpah darah Indonesia dari berbagai ancaman yang datang. Salah satu ancaman yang dihadapi saat ini yaitu ancaman stabilitas dan integritas sistem perekonomian dan sistem keuangan dari tindak pidana pencucian uang dan pendanaan terorisme. TPPU dan TPPT merupakan ancaman yang cukup serius karena dapat merusak sendi-sendi kehidupan bermasyarakat, berbangsa, dan bernegara berdasarkan Pancasila dan Undang-Undang Dasar Negara Republik Indonesia Tahun 1945, dengan adanya penempatan harta kekayaan hasil tindak pidana ke dalam sistem perekonomian.

Di era perkembangan teknologi informasi yang sedemikian pesat disatu sisi terdapat peluang namun disisi yang lain terdapat ancaman. Ancaman siber saat ini masuk dalam kelompok ancaman multidimensi (fisik dan nonfisik). Ancaman siber termasuk dalam ancaman nonfisik karena ancaman siber terkadang tidak dapat terlihat jelas. Disebut ancaman fisik karena dari siber dapat memberikan dampak yang dapat dirasakan secara fisik. Ancaman Siber tidak hanya mengancam instansi pertahanan tetapi juga non-pertahanan. Oleh karena itu dalam menghadapi bentuk dan sifat ancaman nonmiliter di luar wewenang instansi pertahanan, penanggulangannya dikoordinasikan oleh pimpinan instansi sesuai bidangnya (UU Nomor 3 Tahun 2002 tentang Pertahanan Negara, 2002). PPATK yang merupakan instansi pemerintah yang berfungsi sebagai *focal point* rezim anti pencucian uang perlu bersiap untuk mengantisipasi ancaman siber yang bisa datang setiap saat.

Pemerintah perlu melindungi kepentingan umum dari segala jenis gangguan terhadap Infrastruktur Informasi Vital sebagai akibat penyalahgunaan informasi elektronik dan transaksi elektronik yang mengganggu ketertiban umum. Infrastruktur Informasi Vital yang selanjutnya disingkat IIV adalah Sistem Elektronik yang memanfaatkan teknologi informasi dan/atau teknologi operasional, baik berdiri senciiri maupun saling bergantung dengan Sistem Elektronik lainnya dalam menunjang sektor strategis, yang jika terjadi gangguan, kerusakan, dan/ atau kehancuran pada infrastruktur dimaksud berdampak serius terhadap kepentingan umum, pelayanan publik, pertahanan dan keamanan, atau perekonomian nasional. Sektor IIV meliputi (Perpres No. 82, 2002) :

a. Administrasi Pemerintahan;
b. energi dan sumber daya mineral;
c. transportasi;
d. keuangan;
e. kesehatan;
f. teknologi informasi dan komunikasi;
g. pangan;
h. pertahanan; dan
i. sektor lain yang ditetapkan Presiden.

PPATK merupakan salah satu lembaga yang menyelenggarakan administrasi pemerintahan di bidang intelijen keuangan yang dapat dikelompokkkan dalam kelompok infrastruktur informasi vital perlu melindungi penyelenggaraan sistem elektroniknya. PPATK merupakan lembaga Pemerintah di bawah Presiden bertugas untuk mencegah dan memberantas tindak pindana pencucian uang dan pendanaan terorisme (UU No. 8, 2010). PPATK mengelola data-data yang sifatnya rahasia dari Pihak Pelapor yang meliputi penyedia jasa keuangan, penyedia barang dan/ jasa lain, lembaga pembiayaan dan pihak pelapor profesi. Data atau laporan yang dikelola oleh PPATK diantaranya adalah transaksi keuangan mencurigakan, transaksi keuangan tunai, transaksi keuangan transfer dana

dari dan ke luar negeri, pembelian dan penjualan properti, pembelian dan penjualan badan hukum.

Sering terjadinya kebocoran data di Indonesia, yang kemudian datanya diperjual belikan di suatu forum yang ada di internet oleh *hacker* yang bernama Bjorka dapat menurunkan kewibawaan pemerintah atas ketidakmampuan mengamankan data sebagai asset yang berharga. Kebocoran data ini meliputi kartu SIM, Data Polri, dokumen rahasia negara, dan provider internet (Desi Arisandi, Tri Sutrisno, Iwan Kurniawan, 2023). Contoh lainnya meskipun aplikasi Peduli Lindungi sudah dienkripsi kebocoran data terjadi bisa terjadi kelengahan pengguna dan kelemahan aplikasi yang tidak otomatis *logout*. Aplikasi Peduli Lindungi sudah dilakukan enkripsi pada level jaringan dan tetapi tidak disebutkan metode enkripsinya (Hendro Wijayanto, Daryono, Siti Nasiroh , 2021). Persepsi publik terhadap masalah keamanan data dan *data breaches* cenderung negatif berdasarkan hasil analisis sentimen pada diskusi Twitter mengenai topik tersebut (Ahmad Turmudi Zy, Wahyu Hadikristanto, 2023). Maka dari itu sangat penting untuk membuat mekanisme pertahanan untuk memperkuat pertahanan siber secara keseluruhan.

Standar teknis keamanan data dan informasi salah satu aspek nya adalah kerahasiaan. Terpenuhinya aspek kerahasiaan dilakukan dengan prosedur menerapkan enkripsi dengan sistem kriptografi (BSSN, 2021). Proses enkripsi, misalnya pada aplikasi persuratan ditandai dengan perubahan nomor surat, tanggal pengiriman, dan tujuan pengiriman surat yang berupa kata sandi yang tidak dapat dimengerti. Sebaliknya proses dekripsi misalnya pada aplikasi persuratan data berupa surat yang telah dirahasiakan akan diubah kembali menjadi surat yang dapat dibaca dan dipahami (Essay Puspita Sitopu, Nurul Khairina, Rizki Muliono, Muhathir, 2022).

Salah satu penerapan kriptografi adalah pada level aplikasi. Dalam enkripsi tingkat aplikasi, proses enkripsi data diselesaikan oleh aplikasi yang telah digunakan untuk menghasilkan atau mengubah data yang akan

dienkripsi. Pada dasarnya berarti data dienkripsi sebelum ditulis ke database. Pendekatan enkripsi yang unik ini memungkinkan proses enkripsi disesuaikan untuk setiap pengguna berdasarkan informasi (seperti hak atau peran) yang diketahui aplikasi tentang penggunanya. Keuntungan terpenting enkripsi tingkat aplikasi adalah berpotensi menyederhanakan proses enkripsi yang digunakan oleh suatu instansi. Jika suatu aplikasi mengenkripsi data yang ditulis atau dimodifikasi dari database maka alat enkripsi sekunder tidak perlu diintegrasikan ke dalam sistem. Keuntungan utama kedua berkaitan dengan pencurian yang menyeluruh, karenaa data dienkripsi sebelum ditulis ke server maka peretas perlu memiliki akses ke konten database serta aplikasi yang digunakan untuk mengenkripsi dan mendekripsi konten database untuk mendekripsi data sensitif.

Algoritma AES 256 bisa dirancang dan diimplementasikan untuk keamanan data MySQL pada *e-Commerce* dengan tujuan menjadi salah satu cara mengatasi masalah ancaman *SQL Injection* dan pencurian data. (Kartika Imam Santoso, Wahyu Priyoatmoko, 2016). Algoritma beroperasi pada medan galois GF (2^8), dimana semua operasi aritmetika dilakukan pada byte berukuran 8 bit. Rijndael mendukung panjang kunci 128 bit sampai 256 bit dengan *step* 32 bit. Algorithma Rijndael punya 3 parameter yaitu input yang berisi larik 16 bit, output yang berukuran 16 bit hasil enkripsi dan kunci 16 bit kunci *cipher* (Munir, 2019).

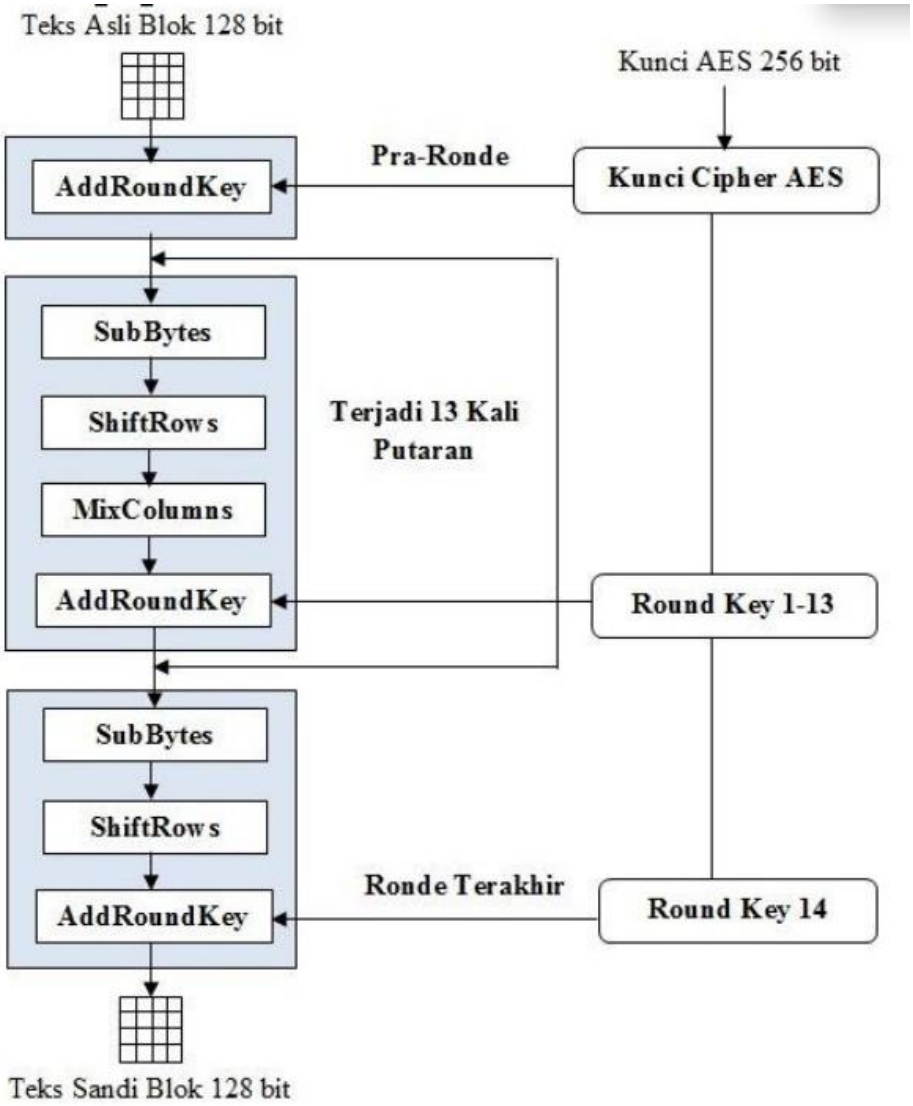
Algoritma enkripsi Rijndael oleh NIST diharapkan menjadi standar kriptografi yang dominan paling sedikit selama 10 tahun kedepan sejak ditetapkan menjadi standar pada tahun 2001. Algoritma AES merupakan kriptografi yang memproses *plainteks* atau *chipperteks* dalam bentuk blok-blok bit dengan panjang sudah ditentukan sebelumnya. Algoritma AES setiap blok panjangnya 128 bit yang berarti setara dengan 16 karakter. Adapun variasi AES yang lain walaupun panjang kuncinya ada yang 128, 192, dan 256 tetapi ukuran bloknya tetap sama yaitu 128 bit seperti pada tabel di bawah ini:

Tabel 2.1 Kunci-Blok-Putaran Kombinasi Algoritma AES

	Key Length		*Block Size*		*Number Of Rounds*
	Nk	*In Bits*	Nb	*In Bits*	Nr
AES-128	4	128	4	128	10
AES-192	6	192	4	128	12
AES-256	8	256	4	128	14

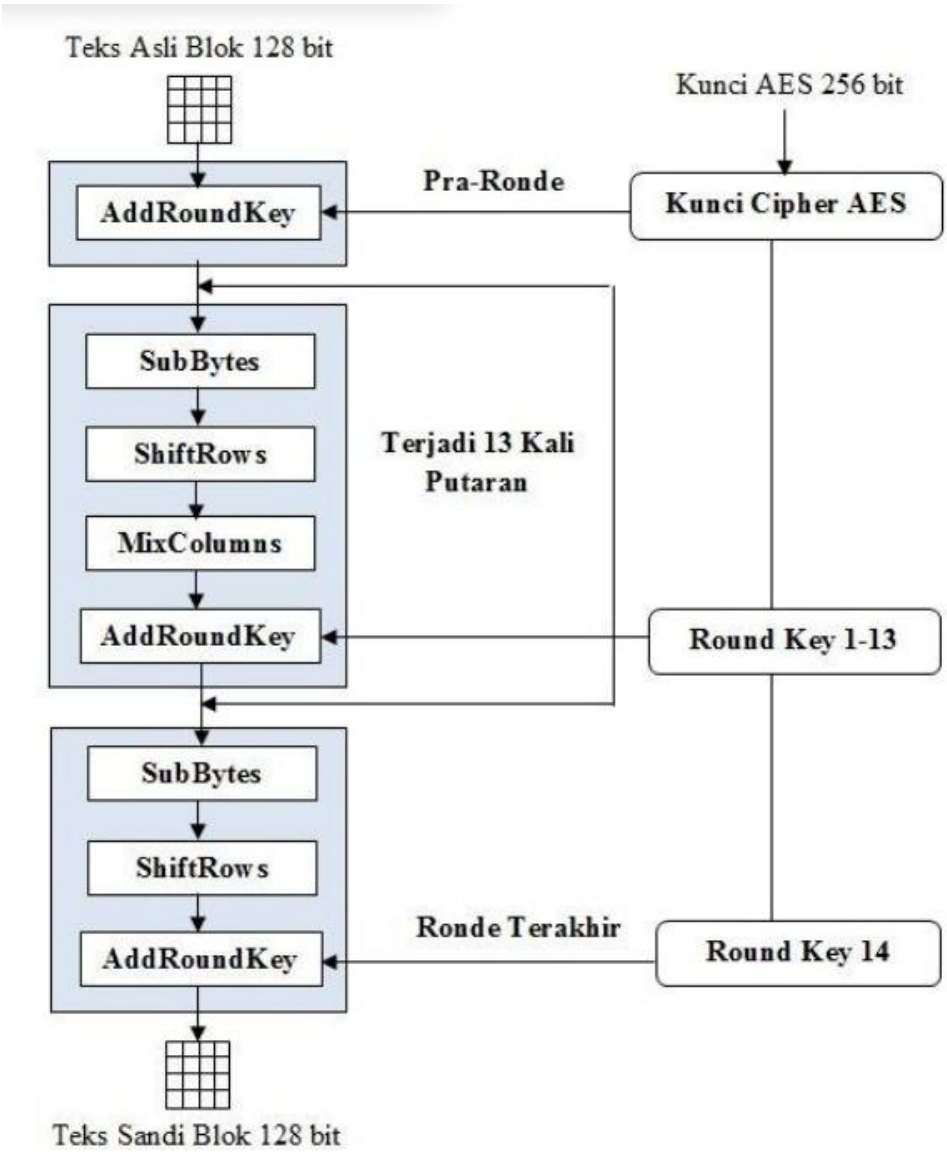
Sumber: (*Information Technology Laboratory National Institute of Standards and Technology U.S. Department of Commerce*, 2023)

Blok-blok data masukan dan kunci dioperasikan dalam bentuk *array*. Setiap anggota *array* sebelum menghasilkan keluaran *ciphertext* dinamakan dengan *state*. Setiap *state* akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu *AddRoundKey*, *SubBytes*, *ShiftRows*, dan *MixColumns*. Kecuali tahap *MixColumns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *MixColumns* tidak akan dilakukan pada tahap terakhir. Proses enkripsi adalah kebalikan dari dekripsi (Kartika Imam Santoso, Wahyu Priyoatmoko, 2016). Proses enkripsi AES 256 seperti gambar di bawah ini:



Gambar 2.1 Alur Proses Enkripsi AES 256

Sedangkan flow proses untuk membaca kembali pesan aslinya dari file yang sudah dienkripsi seperti pada gambar di bawah ini:



Gambar 2.2 Alur Proses Dekripsi AES 256

Pada proses enkripsi dan dekripsi AES proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada state dengan operasi XOR. Setiap *round key* terdiri dari *Nb word* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga:

$[S'_{0,c}, S'_{1,c}, S'_{2,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c}] \oplus [w_{round*Nb+c}]\ untuk\ 0 \leq c \leq Nb$ [wi] adalah *word* dari *key* yang bersesuaian dimana $i = round \times Nb + c$. Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada $round = 0$ untuk *round* selanjutnya $round = round + 1$, pada proses

dekripsi pertama kali pada $round = 14$ untuk *round* selanjutnya $round = round - 1$ berarti untuk AES 256 putaran sebanyak 14 - 1= 13 kali putaran.

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box).

Tabel 2.2 Proses transformasi substitusi bit menjadi nilai tabel 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	53	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	95	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

Sumber : Buku Kriptografi Munir, 2019

Seperti yang telah diketahui sebelumnya, AES merupakan algoritma simetri, yang berarti tabel subtitusi yang dibutuhkan untuk mengenkripsi berbeda dengan untuk mendekripsi. Untuk acuan tersebut, digunakanlah tabel S-box inversi seperti berikut ini:

Tabel 2.3 S-Box Inversi

	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	3d	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

Sumber : Buku Kriptografi Rinaldi Munir, 2019

Algoritma enkripsi Rijndael tersebut secara garis besar sebagai berikut:

Algorithm 1 *Pseudocode for CIPHER()*

```
1: Procedure CIPHER(in, Nr, w)
2:     state ← in
3:     state ← ADDROUNDKEY(state,w[0..3])
4:     for round from 1 to Nr − 1 do
```

```
5:          state ← SUBBYTES(state)
6:          state ← SHIFTROWS(state)
7:          state ← MIXCOLUMNS(state)
8:          state ← ADDROUNDKEY(state,w[4 ∗ round..4 ∗ round
+3])
9:          end for
10:    state ← SUBBYTES(state)
11:    state ← SHIFTROWS(state)
12:    state ← ADDROUNDKEY(state,w[4 ∗Nr..4 ∗Nr +3])
13:    return state
14: end procedure
```

Algorithm AES tersebut tersebut merupakan standard yang dikeluarkan oleh kementrian perdagangan amerika serikat (Information Technology Laboratory National Institute of Standards and Technology U.S. Department of Commerce, 2023). Algoritma AES tersebut detailnya akan dituliskan pada bagian selanjutnya, bagian dibawah ini adalah algoritma untuk membangkitkan kunci AES:

Algorithm 2 *Pseudocode for KEYEXPANSION()*

```
1: Procedure KEYEXPANSION(key)
2:      i ← 0
3:      while i ≤ Nk −1 do
4:          w[i] ← key[4 ∗ i..4 ∗ i+3]
5:          i ← i+1
6:      end while
```

```
7:      while i ≤ 4 ∗Nr +3 do
8:          temp ← w[i − 1]
9:          if i mod Nk = 0 then
10:             temp ← SUBWORD(ROTWORD(temp))⊕Rcon[i/Nk]
11:         else if Nk > 6 and i mod Nk = 4 then
12:         temp ← SUBWORD(temp)
13:         end if
14:         w[i] ← w[i − Nk]⊕temp
15:         i ← i+1
16:     end while
17:     return w
18: end procedure
```

Kemudian algoritma untuk *invers chipper* adalah sebagai berikut.

Algorithm 3 *Pseudocode for INVCIPHER()*

```
1: Procedure INVCIPHER(in, Nr, w)
2:      state ← in
3:      state ← ADDROUNDKEY(state,w[4 ∗Nr..4 ∗Nr +3])
4:      for round from Nr − 1 downto 1 do
5:          state ← INVSHIFTROWS(state)
6:          state ← INVSUBBYTES(state)
7:          state ← ADDROUNDKEY(state,w[4 ∗ round..4 ∗ round
+3])
```

```
8:              state ← INVMIXCOLUMNS(state)
9:      end for
10:     state ← INVSHIFTROWS(state)
11:     state ← INVSUBBYTES(state)
12:     state ← ADDROUNDKEY(state,w[0..3])
13:     return state
14: end procedure
```

Kemudian algoritma untuk *equivalent inverse cipher* adalah sebagai berikut.

Algorithm 4 *Pseudocode for EQINVCIPHER()*

```
1: Procedure EQINVCIPHER(in, Nr, dw)
2:      state ← in
3:      state ← ADDROUNDKEY(state,dw[4 ∗Nr..4 ∗Nr +3])
4:       for round from Nr −1 downto 1 do
5:              state ← INVSUBBYTES(state)
6:              state ← INVSHIFTROWS(state)
7:              state ← INVMIXCOLUMNS(state)
8:              state ← ADDROUNDKEY(state,dw[4 ∗ round..4 ∗ round
+3])
9:      end for
10:     state ← INVSUBBYTES(state)
11:     state ← INVSHIFTROWS(state)
12:     state ← ADDROUNDKEY(state,dw[0..3])
```

```
13: return state
14: end procedure
```

Algorithm 5 *Pseudocode for KEYEXPANSIONEIC()*

```
1: procedure KEYEXPANSIONEIC(key)
2:      i ← 0
3:      while i ≤ Nk − 1 do
4:              w[i] ← key[4i..4i+3]
5:              dw[i] ← w[i] 6: i ← i+1
7:      end while . When the loop concludes, i = Nk.
8:      while i ≤ 4 ∗Nr +3 do
9:              temp ← w[i − 1]
10:             if i mod Nk = 0 then
11:     temp ← SUBWORD(ROTWORD(temp))⊕Rcon[i/Nk]
12:             else if Nk > 6 and i mod Nk = 4 then
13:                     temp ← SUBWORD(temp)
14:             end if 15: w[i] ← w[i − Nk]⊕temp
16:             dw[i] ← w[i] 17: i ← i+1
18:     end while
19:     for round from 1 to Nr − 1 do
20:             i ← 4 ∗ round
21:             dw[i..i+3] ← INVMIXCOLUMNS(dw[i..i+3]) . Note change
of type.
```

```
22:     end for
23:     return dw
24: end procedure
```

MySQL adalah salah satu jenis *database server* yang sangat terkenal, karena menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis *Relational Database Management Sistem* (RDBMS) (Aldi Dwi Febriyanto, Sofia Naning Hertiana, Yudha Purwanto, 2022). Algoritma Elgamal dapat digunakan untuk merubah teks yang terdapat di dalam basis data MySQL menjadi teks rahasia (*ciphertext*) sehingga tidak dapat dibaca oleh pencuri data (Niko Surya Atmaja, Yuhandri Yunus, Sumijan, 2019). Ancaman yang sering terjadi pada aplikasi web adalah *SQL Injection* dan pencurian data, salah satu cara untuk mengatasinya adalah dengan merancang dan mengimplementasikan keamanan data MySQL pada *e-Commerce* dengan Algoritma AES 256 (Kartika Imam Santoso, Wahyu Priyoatmoko, 2016). Metode Merkle Hellman diterapkan pada penyandian melakukan enkripsi dan deskripsi database MySQL dengan pembentukan kunci yang berasal dari bilangan *super increasing* (Ahmad Rifai, Hery Sunandar, 2016). Sistem informasi layanan *online* warga yang dikembangkan dengan menggunakan metode Waterfall dan enkripsi MD5 pada *database* MySQL dapat meningkatkan efisiensi layanan masyarakat (Yohanes Murfi, Sugiyatno, Mugiarso, 2020).

Ada dua kelompok metode kriptografi berdasarkan kuncinya adalah menggunakan kunci simetris dan kunci asimetris. Pengamanan dokomen dengan menggunakan teknik kriptografi algoritma *Blowfish* yang merupakan kriptografi modern dengan kunci simetris berbentuk *cipher block* (Annas Rifa'i, Lilis Cucu Sumartini, 2019). Pada suatu sistem aplikasi yang dibangun menggunakan algoritma Blowfish dinyatakan mampu melakukan enkripsi dan dekripsi dengan baik pada perangkat *mobile*, yaitu membuat *file* tidak dapat diketahui lagi maksud atau teks aslinya (Siswo

Wardoyo, 2014). AES-256 merupakan metode enkripsi dengan menggunakan kunci yang juga sama dengan *Blowfish* yaitu kunci simetris dan *block chipper*. Algoritma AES-128 saja sudah sangat baik dalam mengamankan file ujian untuk menjaga kerahasiaanya dan mendekripsi filenya sama sekali tidak mengalami perubahan dari file asli (Diana Permatasari, Safitri Juanita, 2016). Namun pada penelitian kali ini menggunakan metode AES 256 supaya lebih kuat menggunakan kunci yang lebih panjang. Algoritma AES merupakan algoritma yang direkomendasikan sebagai algoritma untuk mengamankan penggunaan file (Rohbi Visdya Harris Chandra, Ari Kusyanti, Mahendra Data, 2019).

Pada penelitian ini algoritma Rivest Shamir Adleman atau RSA akan digunakan untuk mengamankan kunci simetri AES 256 yang digunakan untuk mengenkripsi database dan file lampiran aplikasi statistik penanganan kejahatan TPPU dan TPPT. Algoritma RSA sebelumnya terbukti dapat digunakan untuk mengenkripsi database pada sistem informasi akademik Sismik (Sudirman Sudirman *et al*., n.d.), dengan demikian diharapkan RSA juga dapat digunakan untuk mengenkripsi kunci enkripsi AES 256 yang disimpan di database. RSA dapat digunakan untuk mengenkripsi seluruh data pada *database SQL server* pada aplikasi program keluarga harapan menggunakan kunci publik dan dapat didekripsi kembali menggunakan kunci privat (Cahya Putra et al., 2021). Metode Rivest Shamir Adleman (RSA) digunakan untuk merahasiakan bagian isi dari database MySQL Perum Bulog Kanwil SUMUT (Farhan & Leman, n.d.). pengamanan database dengan enkripsi RSA dan base64 sangat bagus untuk mengenkripsi data hasil pemilihan e-voting menjadi teks yang tidak dapat dibaca (Pratama Putra et al., 2021). Untuk membuat kriptografi yang lebih aman bisa dengan menggunakan kombinasi algoritma RSA dan AES karena pesan yang dikirim harus melalui dua kali proses enkripsi dan dekripsi (Hermawan et al., 2021).

Algoritma RSA merupakan algoritma kriptografi kunci publik yang paling popular dibuat oleh 3 orang peneliti dari MIT yaitu Ron (R)ivest, Adi

(S)hamir, dan Leonard (A)dleman. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan besar menjadi faktor prima yang akan digunakan menjadi kunci privat. Algoritma RSA terdiri dari tiga tahapan yaitu ekspansi kunci, enkripsi, dan dekripsi. Tahapan ekspansi kunci merupakan tahapan untuk membangkitkan kunci publik dan kunci privat. Pada penelitian ini algoritma RSA digunakan untuk mengenkripsi kunci yang dihasilkan pada saat proses enkripsi sebelumnya yaitu AES 256 selesai dilakukan.

Tahap pertama dari algoritma RSA adalah tahap ekpansi kunci dimana tahapannya adalah memilih dua buah bilangan prima berukuran besar p dan q. Kedua bilangan ini tidak boleh sama. Untuk memperoleh tingkat keamanan yang tinggi pilih p dan q yang berukuran hingga misalnya 1024 bit. Kedua, hitung $n = p\,x\,q$, ketiga, hitung $m = (p-1)\,x\,(q-1)$, keempat memilih *e* yang relatif prima terhadap *m*. Untuk menghasilkan *e* kita perlu mencari $gcd(e,m) = 1$, artinya faktor pembagi terbesar *e* dan *m* adalah 1, mencarinya dengan algoritma Euclidean. Kelima, mencari *d* dengan rumus $e\,x\,d\,mod\,m = 1$, disini kita menebak nilai d yang jika dikalikan dengan e dan di mod kan dengan m maka hasilnya adalah 1. Dan terakhir didapatlah sepasang kunci *public* = (*e*,*n*) dan kunci *private* = (*d,n*).

Langkah-langkah untuk melakukan enkripsi dan dekripsi algoritma RSA secara garis besar adalah pertama dengan ambil kunci publik penerima pesan, e, dan modulus n. Kedua, nyatakan plainteks m_i menjadi blok-blok m_1, m_2, …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n – 1]. Ketiga setiap blok mi dienkripsi menjadi blok ci dengan rumus $c_i = m_i\,e\,mod\,n$. Kemudian untuk mendekripsinya cukup dengan satu langkah yaitu setiap blok *cipherteks* c_i didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i\,d\,mod\,n$.

Aplikasi statistik TPPU dan TPPT saat ini sudah jadi pada versi awal, tetapi aplikasi ini masih menggunakan enkripsi database sebagai fungsi autentikasi. Pada kesempatan penelitian ini pengembangan dilakukan

dalam upaya memperluas fungsi enkripsi untuk melindungi data dan file yang sifatnya rahasia. Metode pembangunan sistem aplikasi dengan *Rapid Aplication Development* atau RAD yang terdiri dari tahap perencanaan, workshop, rancangan basisdata dan antarmuka, dan terakhir pengujian cocok dipilih karena waktu yang diperlukan sangat singkat (Andriani Anik, 2018). Metode RAD digunakan untuk membangun sistem berbasis web untuk perpustakaan yang berisi 12 fungsional yang semuanya dapat berjalan dengan baik (Ardhana *et al*., 2022). Pada penelitian ini kurang lebih terdapat 8 fungsional yang utama seperti pada tabel beritkut ini :

Tabel 2.4 Fungsionalitas Hybrid Kriptografi pada Aplikasi Statistik Penanganan TPPU dan TPPT

No.	Fungsional	Algoritma yang digunakan
1.	Enkripsi level aplikasi dari *plain text* menjadi *chipper text* di database sesuai dengan hak atau peran pengguna	AES 256
2.	Enkripsi file dokumen	AES 256
3.	Enkripsi kunci AES 256	RSA
4.	Dekripsi kunci AES 256	RSA
5.	Dekripsi file dokumen	AES 256
6.	Dekripsi level aplikasi dari *chipper text* pada database menjadi *plain text* pada aplikasi sesuai dengan hak atau peran pengguna	AES 256

Pada penelitian ini akan dilakukan pengujian *white box testing* untuk menguji penambahan fitur enkripsi dan dekripsi pada aplikasi stasistik TPPU dan TPPT yang digunakan oleh PPATK. Pengujian *white box* terdapat beberapa tahapan seperti pemetaan source code, membuat flow *graph*, penghitungan *cyclomatic complexity*, menentukan independent path, pembuatan *graph matrix* bisa diterapkan pada skenario uji pada halaman *login user* (Kusuma & Setiawan, 2018). Pengujian *white box* dengan *basis path* bisa digunakan untuk menguji verifikasi *login* aplikasi (Farhan Londjo,

2021). Pengujian *white box* dengan teknik *basis path* dapat digunakan untuk menemukan cacat atau *error website room* dengan 68 skenario pengujian dengan hasil tingkat resiko rendah terhadap cacat sebesar 94% dan tingkat resiko menengah sebesar 6% (Sie Judith Bryan L et al., 2022). Pengujian *white box testing* sistem informasi penjualan menghasilkan sebuah rekomendasi dalam skala 5 (lima) untuk perusahaan dengan hasil kriteria pertama tampilan mendapatkan rata-rata 4.5, kriteria kedua *user* (pengguna) mendapatkan rata-rata 4.33, kriteria ketiga kemudahan penggunaan mendapatkan rata-rata 4.42, dan kriteria ke empat isi (*content*) mendapatkan rata-rata 4.47 (Suprapti et al., 2017).

2.1. Hasil Penelitian Terdahulu

Untuk mendapatkan informasi penunjang diperlukan untuk mereview penelitian terdahulu yang berkaitan dengan topik penelitian terkait dengan enkripsi pada file dan database. Beberapa penelitian 5 (lima) tahun sebelumnya menjadi rujukan dalan penelitian ini. Penelitian sebelumnya dipetakan dalam bentuk matrik persamaan dan perbedaan untuk menunjukkan originalitas penelitian dan juga mengetahui tujuan penelitian untuk dibandingkan dengan hasil penelitian ini pada bab IV.

Tabel 2. 5 Pemetaan Penelitian 5 Tahun Sebelumnya

No.	Peneliti, Tahun, Judul	Tujuan Penelitian	Ringkasan Penelitian	
			Persamaan	Perbedaan
1.	Rohbi Visdya Harris Chandra, Ari Kusyanti, Mahendra Data. (2019). Analisis Performa Proses Enkripsi dan Dekripsi Menggunakan Algoritme Algoritme AES-128 Pada Berbagai Format File . Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X Vol. 3, No. 1, Januari 2019, 481-486.	- mengkaji tentang algoritma AES sebagai algoritma pengamanan file yang aman dan kecepatan enkripsi dan dekripsi ketika algoritma AES melakukan pengamanan pada tiap file. - Kedua mengimplementasikan algoritme AES untuk enkripsi dan dekripsi terhadap beberapa format file. - Ketiga melakukan analisis perbandingan hasil terhadap keamanan proses enkripsi dekripsi dan kecepatan waktu komputasi algoritme AES saat proses enkripsi dan dekripsi file.	Metode enkripsi AES dapat diterapkan pada tipe data teks, gambar, audio, dan video.	AES diimplementasikan ada file saja, sedangkan penelitian ini AES diimplementasikan pada file sekaligus juga database

2.	*Information Technology Laboratory - National Institute of Standards and Technology U.S. Department of Commerce. (2023). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication.*	*algorithms for block ciphers as the foundation for many cryptographic services provide assurance of the confidentiality of data.*	*Using algorithm AES 256 to assurance of confidentiality of data.*	*Implementation algorithm AES 256 specifically in database mysql and file attachment.*
3.	Wisnu Handi Prabowo, Satriya Wibawa, Fuad Azmi. (2020). Perlindungan Data Personal Siber di Indonesia. Padjadjaran Journal of International Relations, (218-239).	Perlindungan terhadap data yang telah dilakukan oleh Pemerintah Indonesia dan dampaknya terhadap vital core dari human security penduduk Indonesia.	Pentingnya pemerintah melindungi data yang disimpan pada database pemerintah sebagai wujud kedaulatan data.	Implementasi metode teknis untuk untuk melindungi data yang disimpan.
4.	Anjur S Manullang, Ratih Puspasari, Wiwi Verina. (2020). Penyandian Database Menggunakan Metode Base64	Membuat pencuri data harus bekerja lebih sulit untuk mendeskripsikan suatu data dan informasi yang sifatnya	Penyandian database untuk mempersulit	Metode yang digunakan bukan Base64 dan Rot13 tetapi AES 256

	Dan Rot13. Jurnal Mahasiswa Fakultas Teknik dan Ilmu Komputer.	kerahasiaan dengan cara penyandian database menggunakan metode Base64 dan Rot13.	mendeskripsi database.	
5.	Hermawan Aditya, Heri Ujianto Erik Iman. Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA. Jurnal Nasional Informatika dan Teknologi Jaringan VOL. 5 NO.2 (2021) EDISI MARET.	- Algoritma RSA digunakan oleh pengirim untuk menghasilkan kunci publik dan pribadi. - Algoritma RSA juga digunakan untuk mengenkripsi kunci rahasia, dan - algoritma AES digunakan untuk mengenkripsi plaintext. - Pengirim akan mendapatkan kunci pribadi, kunci rahasia terenkripsi, dan ciphertext. Penerima mendekripsi kunci rahasia terlebih dahulu menggunakan algoritma RSA, kemudian mendekripsi ciphertext	Kombinasi AES dan RSA untuk mengenkripsi pesan rahasia.	AES digunakan untuk mengenkripsi database dan file lampiran, sedangkan RSA digunakan untuk mengenkripsi kunci dari database dan file menggunakan Bahasa pemrograman php.

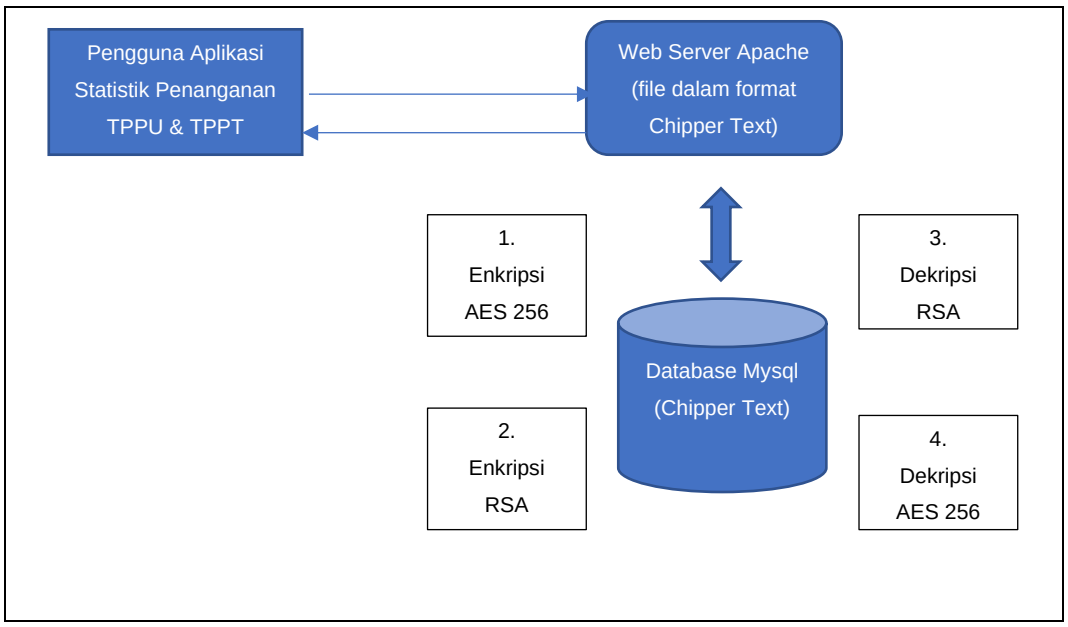
		menggunakan algoritma AES dan kunci rahasia.		
6.	Londjo Muammar Farhan. Implementasi *White box testing* Dengan Teknik Basis Path Pada Pengujian Form Login. Jurnal Siliwangi Seri Sains dan Teknologi Vol.7. No.2, 2021	- Penguji coba fungsi form login dengan membangdingkan hasil uji coba dengan hasil yang diharapkan dalam tabel test case. Jika setelah dilakukan uji pada Aplikasi Form Login dan tidak ditemukan error dalam pengujian, maka semua tes berhasil.	Menguji coba menggunakan tabel test case	Menguji coba enkripsi dan dekripsi pada database dan file lampiran.
7.	Yatong Jiang, Tao Shang, Jianwei Liu, 2021, *SM algorithms-based encryption scheme for large genomic data files*	- *Hybrid kryptografi SM4 symmetric encryption system is used to encrypt the genomic data.* - *SM2 asymmetric encryption algorithm is used to encrypt the SM4 key*	*Hybrid kriptografi between simetric and asymmetric implemented for database encryption*	*Not User SM2 & SM4 that ECC elliptic curve cryptographic based but using AES 256 and RSA*

8.	Mouna Bedoui a , Hassen Mestiri b,d,a , Belgacem Bouallegue c,a , Belgacem Hamdi d,a , Mohsen Machhout, 2021*, An improvement of both security and reliability for AES implementations*	*to obtain a reliable and robust AES implementation against faults injection attacks.*	*can strike a balance between high safety and inexpensive implementation costs.*	*Implement in database using AES 256*
9.	Heba El-Rahman Hassan, Mohamed Tahoun, Gh.S. ElTaweel. 2020. *A robust computational DRM framework for protecting multimedia contents using AES and ECC.*	- *protecting multimedia contents using AES and ECC* - *AES-256 is used to encrypt and decrypt the data and ECC-256 which is used to encrypt and decrypt the shared keys*	*AES 256 used to encypt data multimedia*	*RSA used to encrypt and decrypt shared keys*

2.2. Kerangka Pemikiran

Kerangka pemikiran dalam penelitian ini telah di buat berdasarkan latar belakang permasalahan yang dijawab oleh landasan teori dan penelitian terdahulu sebagai pondasi dasar. Kerangka pemikiran ini merupakan panduan dalam peyusunan metode penelitian sehingga peneliti dapat merinci susunan kerangka kerja secara singkat padat dan sistematis.

Pada intinya penelitian ini adalah kegiatan untuk mengimplementasikan ilmu pertahanan siber pada sistem aplikasi statistik penanganan kejahatan TPPU dan TPPT untuk menghindari salah satu ancaman siber yaitu kebocoran data dengan mengimplementasikan teknik kriptografi. Teknik kriptografi yang akan diimplementasikan adalah kriptografi kunci simetri menggunakan algoritma AES 256 untuk mengenkripsi sekaligus dekripsi database dan file lampiran yang bersifat rahasia. Kunci dari Algoritma AES 256 yang digunakan untuk mengenkripsi database dan file adalah *password* dari operator enkripsi yang disimpan pada tabel di database. Kemudian kunci public AES 256 tersebut dienkripsi kembali menggunakan algoritma kunci asimetri yaitu RSA. Dengan menggunakan kriptografi kunci asismetri RSA tersebut diharapkan dapat memperkuat keamanan dalam menyimpan kunci Algoritma AES yang diterapkan pada enkripsi database dan file tersebut. Operator kriptografi dapat memberikan otoritas siapa pengguna tertentu dapat membaca *record* tertentu. Untuk menguji aplikasi fitur atau fungsi aplikasi secara keseluruhan menggunakan metode pengujian *white box* dan *black box*.



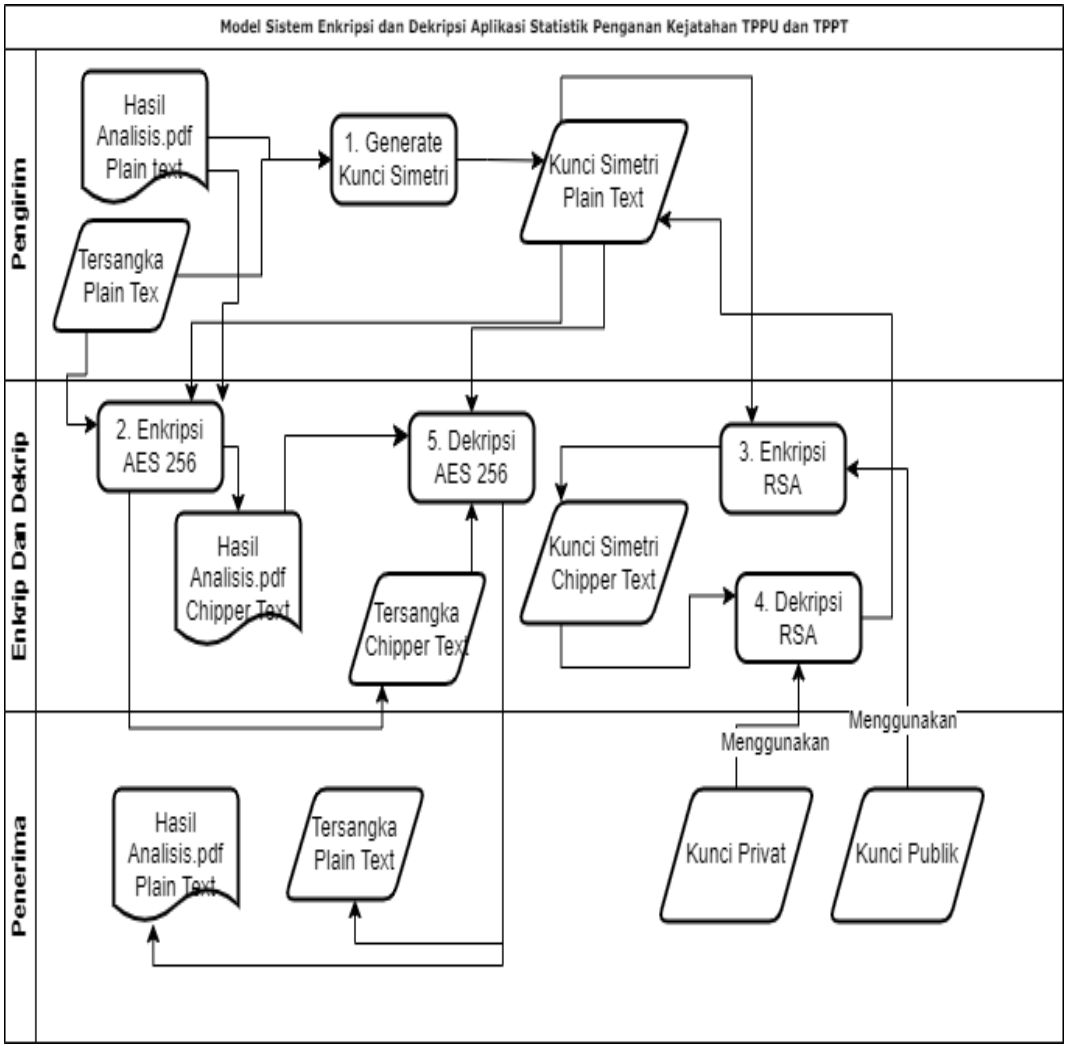
Gambar 2.3 Arsitektur Aplikasi Hybrid Kriptografi AES 256 dan RSA

Prosedur atau urutan langkah bekerjanya penerapan kriptografi pada sistem aplikasi statistik TPPU dan TPPT secara garis besar adalah sebagai berikut :

1. Pengguna input data atau upload file (word atau pdf) masih berupa *plain text* didalamnya termasuk menginput kepada siapa hasil analisis ditujukan, misalnya kepada Kepolisian atau Kejaksaan. Maka nantinya hanya yang dituju saja yang bisa membuka data yang diinputkan oleh PPATK.
2. Inputan di proses secara otomatis menggunakan algoritma AES 256 dalam Bahasa pemrograman php untuk mengenkripsi file dan/atau field tertentu yang sifatnya rahasia misalnya nama tersangka, alamat tersangka.
3. Hasil pemrosesan Algoritma AES 256 tersebut adalah berupa *chipper text* pada database field nama dan alamat atau jika dilampirkan file lampiran (word atau pdf) juga akan menghasilkan *chipper text* dari file dokumen lampiran (word dan atau pdf) hasil analisis yang diupload dan kunci rahasia untuk mendekripsi data dan file hasil analis.

4. Kunci rahasia dari hasil enkripsi AES 256 tersebut kemudian menjadi input untuk proses enkripsi selanjutnya yaitu enkripsi Kriptografi kunci publik RSA.
5. Pemrosesan dilakukan dengan membangkitkan kunci publik dan kunci privat pengirim dalam hal ini PPATK dan kunci publik penerima dan kunci privat penerima dalam hal ini instansi penyidik TPPU dan TPPT misalnya Kepolisian atau Kejaksaan.
6. Kepada masing-masing pengguna password aksesnya digunakan sebagai kunci privat untuk membuka file (word dan atau pdf) juga termasuk data pada database.
7. Supaya pengguna lain bisa membuka akses data atau file operator membagikan kunci akses public yaitu berupa *chipper text* dari algoritma AES kepada pengguna yg lain yg ditunjuk.
8. Pengguna yang bisa membuka kunci akses RSA menggunakan password pada aplikasi yang sudah dienkripsi sebelumnya sebagai kunci privatnya sedangkan kunci public adalah *chipper text* hasil enkripsi password operator enkripsi yang bersifat tidak rahasia.

Untuk memudahkan ilustrasinya bisa dilihat pada gambar *flow chart* berikut ini:



Gambar 2.4 Model Sistem Hybrid Kriptografi AES 256 dan RSA

Adapun data yang akan dienkripsi pada database sistem aplikasi statistik penanganan kejatahan TPPU adalah beberapa kolom tertentu yang sifatnya sangat rahasia diantaranya seperti pada tabel berikut ini :

Tabel 2.6 Daftar Informasi Rahasia yang disimpan pada Database

No.	Nama Tabel atau Entitas	Kolom yang dienkripsi
1.	Tersangka	Nama terlapor
		Lokasi
		Pekerjaan
2.	Asset	Identitas Asset
		Perkiraan Nilai Asset

Kerahasiaan dari informasi tersebut diatas karena dalam hukum harus menerapkapkan prinsip *presumption of innocent* atau praduga tak bersalah.

Sedangkan file yang akan dienkripsi merupakan usulan penulis kedepan agar supaya aplikasi bisa saling terintegrasi antara pengiriiman hasil analisis dengan statistik. Adapun file yang diunggah adalah hasil analisis PPATK atau pemeriksaan dalam bentuk file format pdf atau doc yang diupload ke dalam aplikasi. Pada penelitian ini diharapkan mampu merubah file yang akan disimpan di dalam sistem dari semua *plain text* bisa dibaca dan bermakna menjadi *chipper text* yang tidak dapat ditemukan maknanya. Sehingga file hasil analis yang disimpan di serer nantinya sudah dalam bentuk dienkripsi atau dalam bentuk *chipper text*.

2.3. Hipotesis

Dugaan sementara penulis dari masalah penelitian hybrid kriptografi ini adalah sebagai berikut:

1. Pelaku kejahatan siber akan lebih sulit membocorkan data rahasia dengan kondisi database yang dienkripsi dua tingkat dibandingkan dengan tidak dienkripsi atau dienkripsi satu tingkat.
2. File lampiran yang sifatnya rahasia yang disimpan di server dalam keadaan dienkripsi lebih terjaga kerahasiannya daripada file yang tersimpan dalam bentuk *plain text* di server.
3. Semakin singkat waktu yang diperlukan untuk melakukan proses enkripsi dan dekripsi maka akan semakin dapat diterima oleh pengguna aplikasi.