BAB 2
TINJAUAN PUSTAKA

1.1 Landasan Teori

Landasan teori adalah penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dengan perancangan dan pembangunan aplikasi enkripsi dan dekripsi AES pada UAV dan GCS. Landasan teori ini akan menjadi dasar yang kuat dalam penelitian yang akan dilakukan, terkait teori-teori dalam mendukung penelitian ini akan dibahas pada bab ini.

1.2 Unmanned Aerial Vehicle (UAV)

UAV adalah kependekan dari *Unmanned Aerial Vehicle*, yang merupakan pesawat tanpa pilot. UAV dapat berupa pesawat yang dikendalikan dari jarak jauh misal diterbangkan oleh pilot di stasiun pengendali darat(GCS) atau dapat terbang secara mandiri berdasarkan rencana penerbangan yang diprogram sebelumnya atau sistem otomasi dinamis yang lebih kompleks. UAV saat ini digunakan untuk sejumlah misi, termasuk peran pengintaian dan serangan. Akronim UAV telah diperluas dalam beberapa kasus menjadi UAV (*Unmanned Aircraft Vehicle System*). FAA telah mengadopsi akronim UAS (*Unmanned Aircraft System*) untuk mencerminkan fakta bahwa sistem yang rumit ini mencakup stasiun darat dan elemen lainnya selain kendaraan udara yang sebenarnya[22]. UAV dapat terbagi dalam beberapa tipe dibawah ini :

1. *Target and decoy* - melakukan penembakan darat dan udara dengan target yang mensimulasikan pesawat musuh atau rudal.
2. *Reconnaissance* – memberikan informasi pada medan perang.
3. *Combat* - memberikan kemampuan serangan untuk misi berisiko tinggi.
4. *Research and development* - pengembangan pesawat UAV unuk penelitian.
5. *Civil and Commercial UAVs* - UAV dirancang khusus untuk kegunaan sipil dan komersial.

1.3 Ground Control System (GCS)

Gorund Control System dibuat untuk mengntrol dan memantau penerbangan, serta menerima visualisasi dan perekaman gambar selama penerbangan secara *realtime*. Gorund control system kompatibel dengan berbagai jenis UAV. Dapat memungkinkan untuk menggunakan satu GCS dengan beberapa UAV bahkan itu adalah target atau UAV lainya. Bergantung kepada kebutuhan penggunaan, perangkat lunak dari GCS dapat dikhususkan untuk kebutuhan tertentu. GCS menyediakan antarmuka operator didarat. GCS dan UAV dapat terhubung melalui radio frekuensi maupun jaringan internet[23].

1.4 Communication System (CS)

Sistem Komunikasi atau *Communication System* (CS) terdiri dari sistem modul radio dengan antena pada *gain* yang diinginkan (2 dBi, 10 dBi, 30 dBi) dan frekuensi (900-922MHz, 2,4 GHz, 1,3 GHz) melalui komunikasi serial. Radio *Receiver* adalah bagian dasar terdapat pada *ground*. Sistem bisa berupa komunikasi satu arah atau dua arah yang menerima sinyal dari pesawat(UAV) ke darat(GCS). Dapat mendemodulasi, mendekripsi dan *stream* video ke GCS melalui jaringan internal. Salah satu fungsi CS adalah untuk melacak UAV. Antena di UAV meningkatkan kualitas dan jangkauan sinyal secara dramatis. Dengan informasi yang dikumpulkan dari UAV, CS memerintahkan dan mengarahkan antena *ground* untuk video dan C2 ke UAV pada waktu tertentu secara otomatis[24].

1.5 Pengertian Sistem

Sistem dapat diartikan sebagai satu kesatuan yang terdiri dari komponen-komponen atau subsistem yang tertata dengan teratur, saling interaksi, saling ketergantungan satu dengan yang lainnya, dan tidak dapat dipisahkan (integratif) untuk mewujudkan suatu tujuan. Dibawah ini adalah karakteristik sistem dan klasifikasi sistem, dalam suatu proses dasar penelitian [25]:

1. Karakteristik Sistem

Sesuatu dikatakan sebagai suatu sistem apabila memiliki sifat-sifat tertentu seperti dikemukakan oleh Jogiyanto, sistem memiliki karakteristik atau sifat-sifat tertentu, yakni berikut ini [25]:

a. Mempunyai komponen-komponen (components)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas sistem (boundary)

Setiap sistem memiliki batas-batas luar yang memisahkannya dari lingkungannya. Batas sistem adalah wilayah yang membatasi antara satu sistem dengan sistem yang lainnya atau dengan lingkungannya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

c. Lingkungan luar sistem (enviromments)

Lingkungan luar adalah lingkungan di luar batas sistem yang mempengaruhi operasi sistem. Pengaruh tersebut dapat bersifat positif atau negatif suatu sistem tersebut. Pengaruh yang positif dapat dipelihara dan dijaga, sedangkan pengaruh negatif harus dikendalikan karena dapat mengganggu sistem.

d. Penghubung sistem (interface)

Penghubung adalah media yang menghubungkan atau mengintegrasikan antara satu subsistem ke subsistem yang lainnya menjadi satu kesatuan.

e. Masukan sistem (input)

Masukan adalah serangkaian data (signal input) atau maintenance input dari dalam atau dari luar lingkungan untuk diolah dalam sistem untuk dioperasikan. Contoh di dalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran sistem (output)

Keluaran adalah hasil dari proses dan diklasifikasi menjadi keluaran yang berguna. Keluaran merupakan masukan untuk subsistem yang lain. Informasi adalah keluaran yang dihasilkan dari proses.

g. Pengolah sistem (pemrosesan)

Pengolah merupakan suatu yang merubah masukan menjadi keluaran. Contoh Sistem akuntansi akan mengolah data-data transaksi menjadi laporan keuangan yang diperlukan oleh manajemen.

h. Sasaran sistem

Sistem yang baik tentu memiliki sasaran yang ingin dicapai. Sasaran adalah sesuatu yang menjadi target yang ingin dicapai dari suatu sistem. Sasaran yang dicapai dari suatu sistem menentukan masukan yang dibutuhkan. Suatu sistem dikatakan berhasil apabila sasaran yang telah ditentukan dapat dicapai dengan baik.

2. Klasifikasi Sistem

Klasifikasi sistem merupakan kesatuan antara satu komponen dengan satu komponen lainnya, tujuan dari sistem tersebut memiliki akhir tujuan yang berbeda untuk setiap kasus yang terjadi dalam setiap sistem. Suatu sistem dapat diklasifikasikan sebagai berikut [25]:

a. Sistem abstrak (abstract system) dan sistem fisik (physical system).

Sistem abstrak adalah sistem berupa pemikiran atau ide-ide yang tidak tampak secara fisik, seperti sistem teologia. Sistem fisik adalah sistem yang nyata secara fisik, seperti sistem komputer, sistem akuntansi, sistem informasi.

b. Sistem alamiah (natural system) dan sistem buatan manusia (human made system).

Sistem alamiah adalah sistem yang terjadi secara alami, tidak dibuat oleh manusia, misal sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang dan dibuat oleh manusia, misal

sistem informasi akuntansi, sistem pendidikan. Apabila sistem dirancang dan dibuat manusia berinteraksi dengan mesin maka disebut humanmachine system.

c. Sistem tertentu (deterministic system) dan sistem tidak tentu (probabilistic system).

Sistem tertentu adalah sistem yang beroperasi dengan perilaku yang sudah dapat diprediksi. Interaksi antarbagian dapat dideteksi dengan pasti sehingga keluaran dari sistem sudah dapat diramalkan, misal sistem komputer. Sistem tak tentu adalah sistem di mana kondisi ke depannya tidak dapat diprediksi karena mengandung teori kemungkinan.

d. Sistem tertutup (closed system) dan sistem terbuka (open system).

Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luar. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Namun, sebenarnya tidak ada sistem yang tertutup, yang ada adalah relatif tertutup, tidak benar-benar tertutup. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

1.6 Keamanan Informasi

Keamanan informasi merupakan salah satu aspek yang penting dalam berkomunikasi, namun keamanan sering kali dilupakan ketika melakukan komunikasi. Informasi yang jatuh ke tangan yang salah dapat menimbulkan masalah yang besar seperti penipuan, pencurian, pemerasan dan masih banyak lagi

masalah yang akan ditimbulkan.

Keamanan informasi mempunyai beberapa aspek yang harus dipenuhi, agar informasi dapat terjamin keaslian dan keamanannya. Aspek-aspek umum

tersebut meliputi *confidentiality, integrity, authentication, availability* dan *non repudiation*[26].

1. Confidentiality

Confidentiality atau kerahasiaan yaitu keamanan informasi harus dapat menjamin kerahasiaan dari informasi. Salah satu cara yang dapat dilakukan yaitu dengan membatasi hanya orang yang mempunyai hak saja yang dapat membaca atau mengubah suatu informasi.

2. Integrity

Integrity berhubungan dengan keaslian informasi, *integrity* yaitu menjamin keutuhan dan keaslian informasi yang dikirimkan, agar informasi tidak dirusak atau

diubah oleh orang yang tidak berhak. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain ke dalam data yang sebenarnya.

3. Authentication

Authentication yaitu usaha atau metode untuk mengetahui keaslian dari informasi, dan memastikan bahwa informasi diterima oleh orang yang benar. Untuk

menjaga otentikasi terhadap informasi, dapat digunakan digital "*signature"* untuk memastikan keaslian informasi.

4. Availability

Availability atau ketersediaan data yaitu keamanan informasi harus dapat menjamin bahwa data atau informasi harus tersedia ketika akan digunakan. Untuk menjamin ketersediaan data, sistem harus mempunyai cadangan data jika data tersebut hilang.

5. Non Repudiation

Non repudiation yaitu menjamin bahwa seorang pengirim informasi tidak dapat menyangkal keaslian dari informasi yang dikirimnya. Sehingga penerima informasi dapat memastikan bahwa informasi yang diterima merupakan informasi yang asli.

1.7 Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi, dan otentikasi data[26]. Kriptografi bukan satu-satunya cara untuk menyediakan keamanan informasi, melainkan satu set teknik yang dapat digunakan
untuk mengamankan informasi.

Secara umum, kriptografi terdiri dari dua buah bagian utama yaitu bagian enkripsi dan bagian dekripsi. Enkripsi adalah proses transformasi informasi menjadi bentuk lain sehingga isi pesan yang sebenarnya tidak dapat dipahami, hal ini dimaksudkan agar informasi tetap terlindung dari pihak yang tidak berhak menerima. Sedangkan dekripsi adalah proses kebalikan enkripsi, yaitu transformasi
data terenkripsi ke data bentuk semula. Proses transformasi dari *plaintext* menjadi *ciphertext* akan dikontrol oleh kunci. Peran kunci sangatlah penting, kunci bersama-sama dengan algoritma matematisnya akan memproses *plaintext* menjadi *ciphertext* dan sebaliknya.

Kriptografi tidak memenuhi semua aspek dari keamanan informasi. Kriptografi hanya memenuhi empat aspek dalam keamanan informasi yang merupakan tujuan dari kriptografi. Ke empat aspek tersebut yaitu kerahasiaan (*confidentiality*), integritas data (*integrity*), otentikasi data (*authentication*), dan *non-repudiation*[27].

1. Kerahasiaan

Kerahasiaan adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah disandi. Kriptografi memenuhi aspek kerahasiaan karena informasi tidak dapat secara langsung diketahui.

2. Integritas data

Integritas data adalah layanan yang berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus

memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain ke dalam data yang sebenarnya.

3. Otentikasi

Otentikasi adalah layanan yang berhubungan dengan identifikasi/ pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui jaringan harus diotentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain. Untuk alasan ini aspek kriptografi biasanya dibagi menjadi dua kelas utama yaitu otentikasi entitas dan otentikasi data asal.

4. Non-repudiation

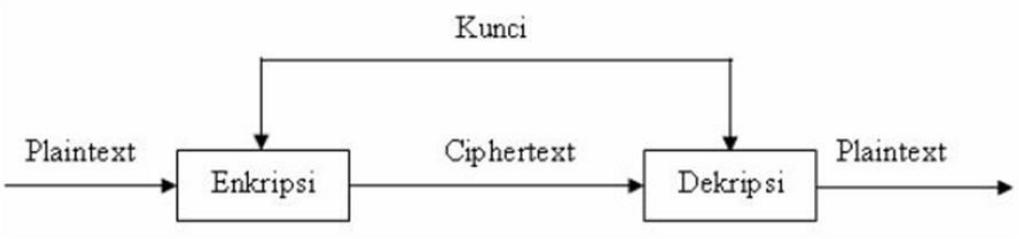
Non-repudiation adalah layanan yang mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan atau membuat. Sebagai contoh, satu entitas dapat mengizinkan pembelian properti oleh entitas lain dan kemudian berusaha menyangkal otorisasi tersebut diberikan. Sebuah prosedur yang melibatkan pihak ketiga yang tepercaya diperlukan untuk menyelesaikan sengketa tersebut.

1.7.1 Algoritma Kriptografi

Algoritma kriptografi yang handal adalah algoritma kriptografi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Berdasarkan jenis kuncinya, algoritma kriptografi dibagi menjadi dua jenis yaitu algoritma simetris dan algoritma asimetris[27].

1.7.1.1 Algoritma Kunci Simetris

Algoritma Kunci simetris (*symmetric key algorithm*) adalah suatu algoritma di mana kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key algorithm*[27]. Ilustrasi penggunaan algoritma kriptografi dengan kunci simetris dapat terlihat pada gambar 2.1 berikut.

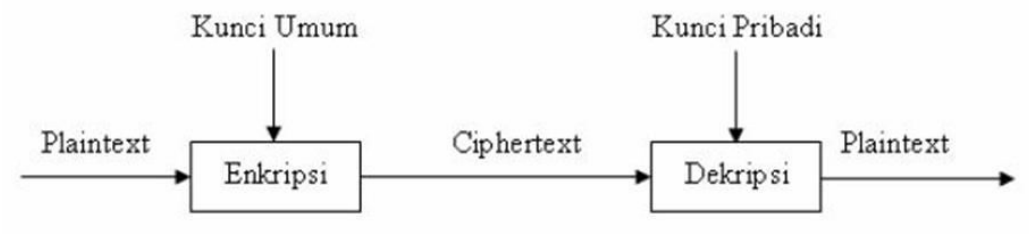


Gambar 0.1 Ilustrasi Kriptografi Dengan Kunci Simetris.

Algoritma kunci simetris banyak digunakan karena lebih cepat dan lebih simpel, namun penggunaan kunci simetris juga mempunyai kekurangan, karena jika kunci dapat diketahui, maka informasi pun dapat diketahui. Beberapa algoritma kriptografi yang termasuk pada algoritma kunci simetris yaitu DES, AES, Blowfish, dan IDEA.

1.7.1.2 Algoritma Kunci Asimetris

Algoritma kunci asimetris (*asymmetric key algorithm*) adalah suatu algoritma di mana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi[18]. Algoritma ini menggunakan dua kunci yakni kunci publik (*public key*) dan kunci privat (*private key*). Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Ilustrasi penggunaan algoritma kriptografi dengan kunci asimetris dapat terlihat pada gambar 2.2 berikut.



Gambar 0.2 Ilustrasi Kriptografi Dengan Kunci Asimetris.

Algoritma asimetris mempunyai keamanan yang lebih baik, karena jika *public key* diketahui, informasi belum tentu dapat diketahui karena *private key* kemungkinan berbeda. Namun akan menjadi sulit dan lama ketika implementasinya. Beberapa algoritma kriptografi yang termasuk pada algoritma asimetris yaitu Diffie - Hellman, RSA, ElGamal, dan DSA.

1.7.2 Advanced Encryption Standard (AES)

Advanced Encryption Standard adalah cipher blok simetris yang bisa memproses blok data 128 bit, menggunakan kunci sandi dengan panjang 128, 192, dan 256 bit. Algoritma AES dapat digunakan dengan tiga panjang kunci yang berbeda yang ditunjukkan di atas, dan oleh karena itu "kunci" yang berbeda ini dapat disebut sebagai "AES-128", "AES-192", dan "AES-256".

Untuk algoritma AES, jumlah putaran yang harus dilakukan selama eksekusi algoritma tergantung pada ukuran kunci. Jumlah putaran diwakili oleh Nr, di mana Nr = 10 saat Nk = 4, Nr = 12 saat Nk = 6, dan Nr = 14 saat Nk = 8. Tabel 2.1 adalah jumlah proses yang harus dilakukan untuk masing-masing kunci.

Tabel 0.1 Kombinasi Key-Block-Round

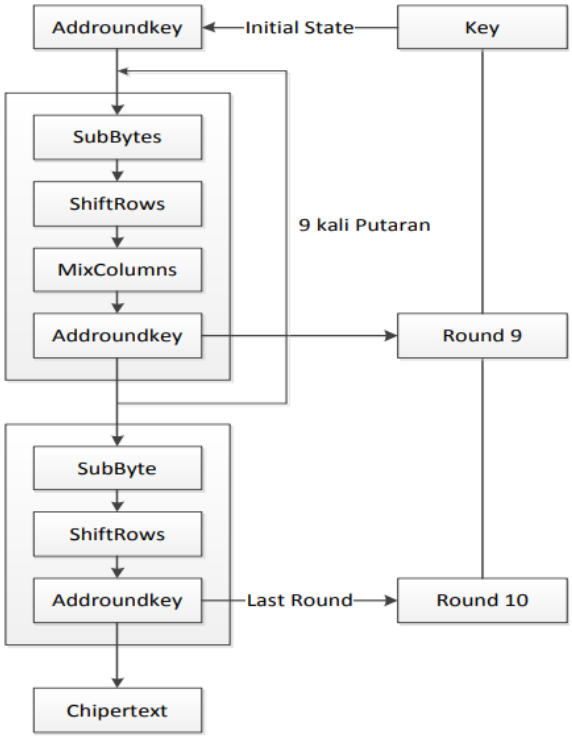
	Panjang Kunci (NK Words)	**Ukuran Block (Nb Words)**	**Jumlah Ronde/Proses (Nr)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran cipher text dinamakan dengan state. Setiap state akan mengalami proses yang terdiri dari empat tahap yaitu, *Add Round*
Key, Sub Bytes, Shift Rows, dan Mix Columns. Kecuali pada tahap *Mix Columns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *Mix Columns*

tidak akan dilakukan pada tahap terakhir[19].

1.7.2.1 Proses Enkripsi Algoritma AES

Proses enkripsi algoritma AES terdiri atas empat jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah dikopikan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Selanjutnya, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey* secara berulang-ulang sebanyak *Nr*. Proses tersebut dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir berbeda dengan *round* sebelumnya di mana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*[29]. Gambar 2.3 merupakan langkah-langkah proses enkripsi AES.



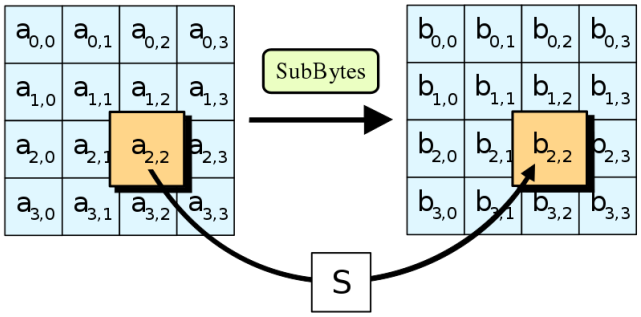
Gambar 0.3 Proses enkripsi algoritma AES

1. **Transformasi SubBytes**

Transformasi *SubBytes()* memetakan setiap byte dari array state menggunakan tabel subtitusi S-box. Tabel S-box yang digunakan adalah seperti pada gambar 2.4 dan proses SubBytes pada gambar 2.5.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

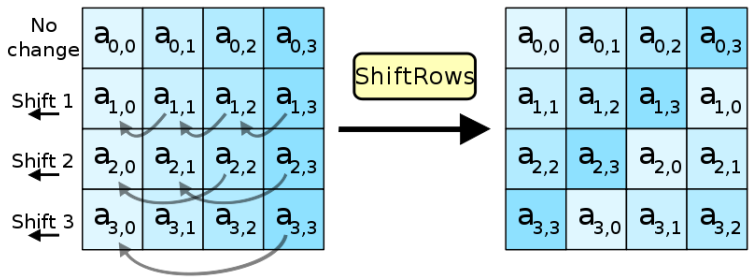
Gambar 0.4 Tabel S-box



Gambar 0.5 Proses SubBytes

2. **Transformasi *ShiftRows***

Tahap *ShiftRows* akan menggeser ke kiri secara berputar setiap *bytes* dalam setiap baris dari *state*. Jumlah pergeseran setiap *byte* berbeda untuk setiap barisnya. Baris pertama akan tetap pada keadaan semula. Setiap *byte* dari baris kedua digeser satu langkah ke kiri. Baris ketiga dan keempat digeser ke kiri sebanyak dua dan tiga langkah. Proses pergeseran *ShiftRows* pada gambar 2.6.



Gambar 0.6 Proses ShiftRows

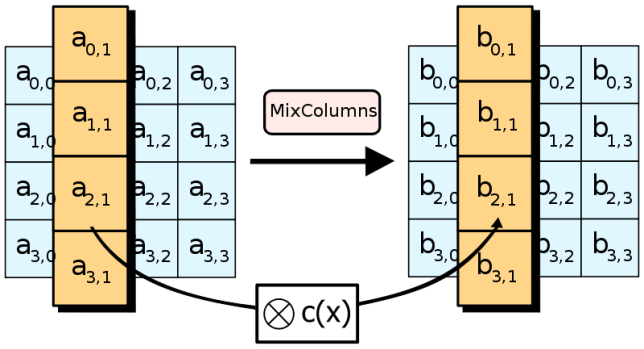
3. Transformasi MixColumns

Tahap MixColumns dapat dilakukan dengan mengalikan empat angka dari kolom state dalam **GF**(2^8) dengan perkalian matriks:

s'(x) = a(x) ⊗ s(x)

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

Karena perhitungan ini dilakukan dalam galois field milik Rijndael **GF**(2^8), operasi penjumlahan sebenarnya adalah operasi XOR. Proses transformasi *MixCoulmns* pada gambar 2.7.

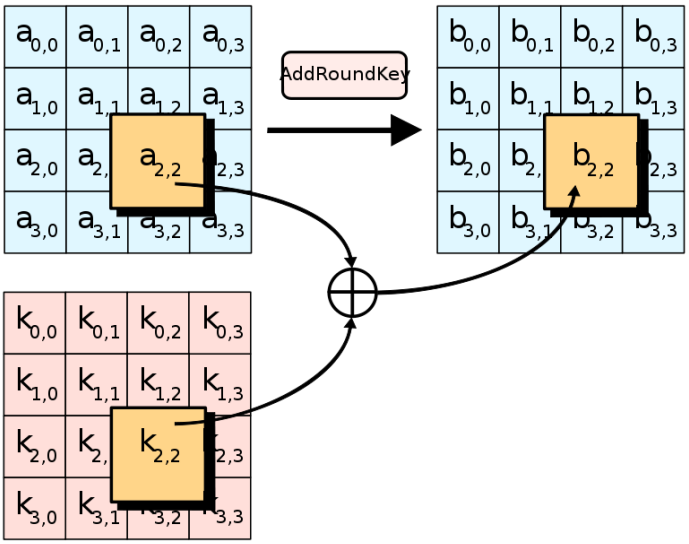


Gambar 0.7 Proses MixColumns

4. AddRoundKey

Pada enkripsi dan dekripsi AES, Pada proses ini *subkey* digabungkan dengan *state*. Proses penggabungan ini menggunakan operasi XOR untuk setiap *byte* dari *subkey* dengan *byte* yang bersangkutan dari *state*. Untuk

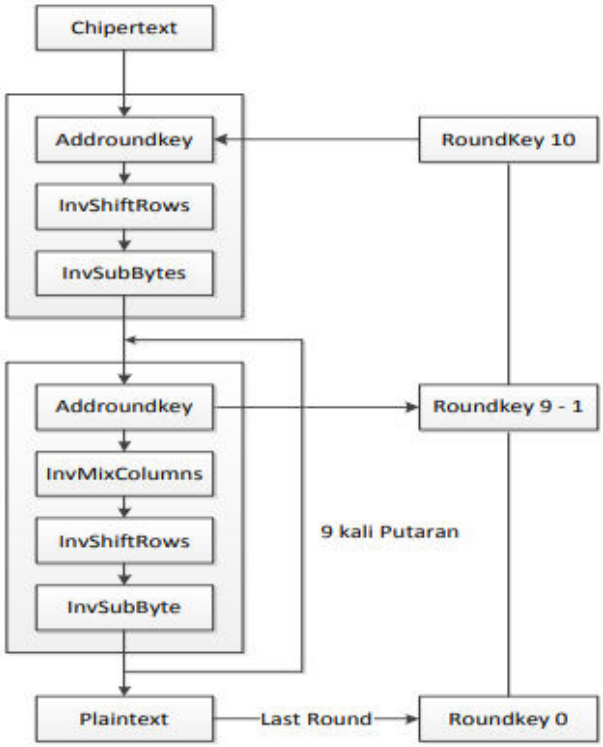
setiap tahap, *subkey* dibangkitkan dari kunci utama dengan menggunakan proses *key schedule*. Setiap *subkey* berukuran sama dengan *state* yang bersangkutan. Proses *AddRoundKey* diperlihatkan pada Gambar 2.8.



Gambar 0.8 Proses AddRoundKey

1.7.2.2 Proses Dekripsi Algoritma AES

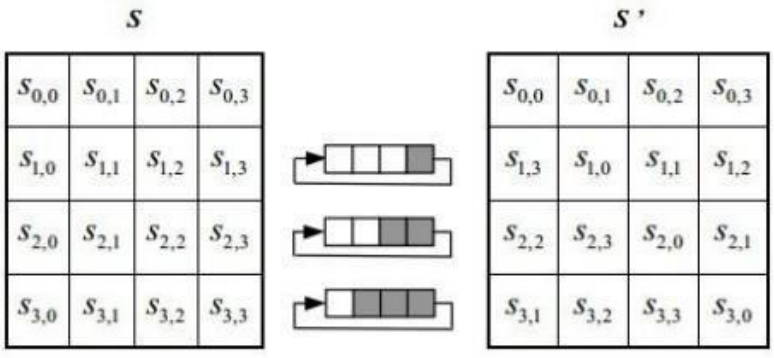
Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*[29]. Gambar 2.9 merupakan langkah-langkah proses dekripsi AES.



Gambar 0.9 Proses dekripsi algoritma AES

1. **Inverse ShiftRows**

   InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows* dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran bit ke kiri. Ilustrasi transformasi *InvShiftRows* dapat pada Gambar 2.10.



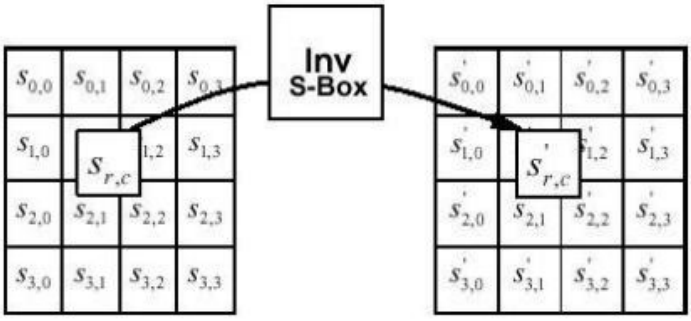
Gambar 0.10 Proses InvShiftRows

2. Inverse SubBytes

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes* tiap elemen pada state dipetakan dengan menggunakan table Inverse *S-Box*. Tabel *Inverse S-Box* akan ditunjukkan dalam gambar 2.11. dan proses *InvSubBytes* pada gambar 2.12.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Gambar 0.11 Tabel InvS-Box



Gambar 0.12 Proses InvSubBytes

3. Inverse MixColumns

Setiap kolom dalam state dikalikan dengan matriks perkalian dalam AES. Operasi *InvMixColumns* memperlakukan setiap kolom sebagai polinomial 4 suku dalam Galois field dan kemudian dikalikan dengan c(x) modulo (x4+1) c(x)=11x3+13x2+9x+14. Operasi *InvMixColumns* juga dapat dipandang sebagai perkalian matrix. Langkah *InvMixColumns* dapat

ditunjukkan dengan mengalikan 4 bilangan di dalam Galois field oleh matriks berikut ini:

$$\begin{bmatrix} r0 \\ r1 \\ r2 \\ r3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 14 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \end{bmatrix}$$

Atau

$$\begin{bmatrix} r0 \\ r1 \\ r2 \\ r3 \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \end{bmatrix}$$

Dimana r0, r1, r2 dan r3 adalah hasil setelah transformasi. a0 - a3 dapat diperoleh dari matriks setelah data mengalami proses substitusi dalam InvSbox. Untuk memperoleh nilai r0, rumusnya seperti dibawah ini:

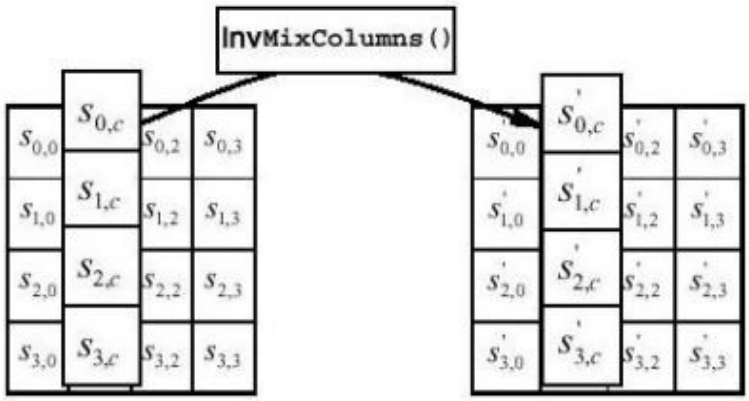
$r0 = \{0e.\ a0\} xor \{0b.\ a1\} xor \{0d.\ a2\} xor \{09.\ a3\}$

$r1 = \{09.\ a0\} xor \{0e.\ a1\} xor \{0b.\ a2\} xor \{0d.\ a3\}$

$r2 = \{0d.\ a0\} xor \{09.\ a1\} xor \{0e.\ a2\} xor \{0b.\ a3\}$

$r3 = \{0b.\ a0\} xor \{0d.\ a1\} xor \{09.\ a2\} xor \{0e.\ a3\}$

Proses Inverse MixColumns ditunjukan pada gambar2.13.



Gambar 0.13 Proses Inverse MixColumns

1.8 Man In The Middle

Man In The Middle Attack adalah salah satu teknik dalam keamanan jaringan dimana penyusup menempatkan dirinya berada di tengah-tengah dua

perangkat atau lebih yang saling berkomunikasi. Hasil dari teknik *Man In The Middle Attack* ini adalah penyadapan informasi[30].

1.9 Modul nRF24L01

NRF24L01 merupakan modul komunikasi jarak jauh yang menggunakan frekuensi pita gelombang radio 2.4-2.5 GHz ISM (Industrial Scientific and Medical). nRF24L01 memiliki kecepatan sampai 2Mbps dengan pilihan opsi date rate 250 Kbps, 1 Mbps, dan 2 Mbps. Transceiver terdiri dari synthesizer frekuensi terintegrasi, kekuatan amplifier, osilator kristal, demodulator, modulator dan Enhanced ShockBurst ™ mesin protokol. output daya, saluran frekuensi, dan setup protokol yang mudah diprogram melalui antarmuka SPI. Konsumsi arus yang digunakan sangat rendah, hanya 9.0mA pada daya output -6dBm dan 12.3mA dalam mode RX. Built-in Power Down dan mode standby membuat penghematan daya dengan mudah realisasi[31].

1.10 Sensor pada UAV

1. Gyroscope

Gyroscope adalah perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut, alat ini bekerja sama dengan *accelerometer*. Mekanismenya adalah sebuah roda berputar dengan piringan di dalamnya yang tetap stabil. Alat ini sering digunakan pada robot atau *drone* serta alat-alat canggih lainnya.

2. Magnetometer

Magnetometer adalah sensor yang bekerja atas dasar pendeteksian gaya magnet bumi. Salah satu kegunaan dari *magnetometer* adalah untuk menentukan arah mata angin. Sensor magnetik mengukur medan magnet. Dengan menghitung sudut medan magnet bumi yang terdeteksi, dan membandingkan sudut pengukuran dengan gravitasi yang diukur dengan *accelerometer*, dapat memungkinkan untuk mengukur arah perangkat berkenaan dengan Utara dengan tingkat akurasi yang tinggi.

3. Accelerometer

Sensor ini mengukur akselerasi, yang meliputi komponen akselerasi dengan gerak perangkat dan akselerasi akibat gravitasi. *Accelerometer* juga tersedia dengan sumbu tunggal, ganda atau tiga, yang didefinisikan dalam sistem koordinat X, Y, Z. Periode gerakan kompleks perangkat dapat menyebabkan perhitungan orientasi menjadi sulit, terutama selama gerak cepat dan kompleks, di mana sinyal mencakup penjumlahan percepatan linear, percepatan sentripetal, dan gravitasi.

1.11 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat(*tool*) untuk mendukung pengembangan sistem tersebut. UML juga sebagai dasar bagi perangkat desain berorientasi objek dari IBM.

UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

Berikut akan dijelaskan empat macam diagram yang akan digunakan dalam penelitian skripsi ini, yaitu *use case* diagram, *activity* diagram, *class* diagram dan *sequence* diagram [32].

1.11.1 Use Case Diagram

Use Case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use Case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* mempresentasikan orang yang akan mengoperasikan atau orang yang berinterkasi dengan sistem aplikasi. *Use Case* mempresentasikan operasioperasi yang dilakukan oleh aktor. *Use Case*

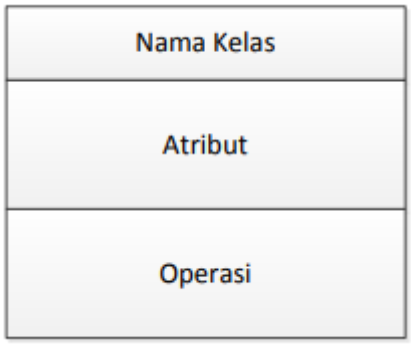
digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke *use case*[32].

1.11.2 Activity Diagram

Activity diagram merupakan diagram yang menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini memiliki struktur diagram yang mirip dengan *flowchart* atau data *flow* diagram pada perancangan terstruktur. Manfaat dari diagram ini yaitu dapat membantu dalam memahami proses secara keseluruhan. *Activity* diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram[32].

1.11.3 Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. *Class* seperti juga objek adalah sesuatu yang membungkus (*encapsulate*) informasi (baca: atribut) dan perilaku (baca: operasi) dalam dirinya.



Gambar 0.14 Notasi Kelas Dalam UML

Pada gambar 2.14 memperlihatkan notasi kelas dalam UML, berikut ini adalah penjelasannya:

a. Bagian paling atas memuat nama kelas.
b. Bagian tengah mendaftarkan atribut-atribut yang dimiliki sebuah kelas, atribut seperti yang kita ketahui, adalah informasi-informasi yang

berkaitan dengan suatu kelas. Sebagai contoh, kelas Mahasiswa mungkin memiliki NIM, Nama Mahasiswa, Tgl Lahir, Alamat dan sebagainya, yang merupakan atribut dari kelas Mahasiswa tersebut.

c. Sedangkan bagian paling bawah mendaftarkan operasi-operasi yang dimiliki oleh sebuah kelas. Operasi berhubungan dengan perilaku yang berhubungan dengan suatu kelas. Operasi biasanya memuat 3 bagian, yaitu: nama operasi itu sendiri, parameter-parameter operasi, serta tipe kembalian. Dalam hal ini, parameter-parameter merupakan argumenargumen yang diterima oleh suatu operasi sebagai asupannya (*input*), sedangkan tipe kembalian adalah luaran (*output*) operasi yang bersangkutan.

Pada *class* diagram, kita dapat melihat baik nama operasi, parameterparameter, serta tipe luarannya. Untuk mengurangi kerumitan saat menghasilakan kode-kode dalam bahasa pemrograman kelak pada *class* diagram, adalah sangat membantu jika *class* diagram dilengkapi dengan ketiga bagian operasi di atas [32].

1.11.4 Sequence Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*, interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Masing-masing *sequence* diagram akan menggambarkan aliran-aliran pada suatu *use case*. Kita dapat membaca diagram ini dengan melihat pada objek-objek dan pesan-pesan (*message*). Objek-objek yang berperan dalam aliran diperlihatkan pada kotak empat persegi panjang yang melintas pada bagian atas diagram.

Setiap objek memiliki garis hidup (*lifeline*), yang digambarkan sebagai garis vertikal dibawah nama suatu objek. Garis hidup dimulai saat suatu objek terbentuk (diinstansiasi) dan berakhir saat objek yang bersangkutan dihancurkan. Pesan-pesan digambarkan diantara garis hidup yang dimiliki dua objek untuk

memperlihatkan bagaimana objek-objek itu saling berkomunikasi. Masing-masing pesan menggambarkan suatu objek yang membuat pemanggilan fungsi dari objek lainnya. Dikemudian hari, saat kita mendefinisikan operasi-operasi untuk suatu kelas, masingmasing pesan akan menjadi operasi. Pesan-pesan dapat juga bersifat refleksif; suatu objek memanggila operasi yang dimilikinya sendiri [32].