PENERAPAN ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES) PADA BEBERAPA MODE ENKRIPSI UNTUK KEAMANAN JAWABAN *COMPUTER BASED TEST* (CBT)

SKRIPSI

CHRISTINA PAULINNA S

H011201071

PROGRAM STUDI MATEMATIKA DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2024

PENERAPAN ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES) PADA BEBERAPA MODE ENKRIPSI UNTUK KEAMANAN JAWABAN *COMPUTER BASED TEST* (CBT)

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

CHRISTINA PAULINNA S

H011201071

PROGRAM STUDI MATEMATIKA DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2024

HALAMAN PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul

Penerapan Algoritma *Advanced Encryption Standard* (AES) Pada Beberapa Mode Enkripsi Untuk Keamanan Jawaban *Computer Based Test* (CBT)

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 31 Mei 2024

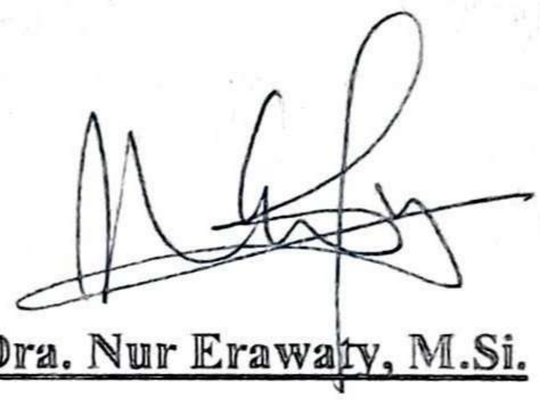
Christina Paulinna S

H011201071

PENERAPAN ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES) PADA BEBERAPA MODE ENKRIPSI UNTUK KEAMANAN JAWABAN *COMPUTER BASED TEST*

Disetujui oleh:

Dosen Pembimbing



Dra. Nur Erawaty, M.Si.

NIP. 196909121993032001

Pada 31 Mei 2024

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Christina Paulinna S

NIM : H011201071

Program Studi : Matematika

Judul Skripsi : Penerapan Algoritma *Advanced Encryption Standard* (AES) Pada Beberapa Mode Enkipsi Untuk Keamanan Jawaban *Computer Based Test* (CBT)

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Ketua : Dra. Nur Erawaty, M.Si. ( )

Anggota : Prof.Dr. Amir Kamal Amir, M.Sc. ( )

Anggota : Muhammad Sadno, S.Si., M.Si. ( )

Ditetapkan di : Makassar

Tanggal : 31 Mei 2024

HALAMAN PENGESAHAN

PENERAPAN ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES) PADA BEBERAPA MODE ENKRIPSI UNTUK KEAMANAN JAWABAN *COMPUTER BASED TEST* (CBT)

Disusun dan diajukan oleh

CHRISTINA PAULINNA S

H011201071

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

Pada tanggal 31 Mei 2024

dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Dosen Pembimbing	Ketua Program Studi
Dra. Nur Erawaty, M.Si.	Dr. Firman,S.Si., M.Si
NIP. 196909121993032001	NIP. 196804292002121001

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa atas segala berkat dan rahmat-Nya sehingga tugas akhir skripsi dengan judul " Penerapan Algoritma *Advanced Encryption Standard* (AES) Pada Beberapa Mode Enkipsi Untuk Keamanan Jawaban *Computer Based Test* (CBT)" dapat penulis selesaikan dengan segala keterbatasan yang dimiliki. Skripsi ini merupakan tugas akhir untuk mencapai gelar Sarjana Sains pada Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Dalam penyusunan skripsi ini, penulis banyak mendapat bantuan dari berbagai pihak terutama dari **Bapak Muhammad Sadno, S.Si., M.Si** selaku pembimbing yang juga selaku penguji dan **Ibu Dra. Nur Erawaty, M.Si.** selaku pembimbing utama dan penasihat akademik, terima kasih yang sebesar-besarnya atas bimbingan serta petunjuk yang diberikan selama penyusunan skripsi ini.

Penulis juga mengucapkan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya kepada:

1. Ayahanda **Sakarias Rada** dan Ibunda **Theresia Nursia** yang dengan penuh kesabaran memberikan kasih sayang, dorongan, semangat, dan doa dengan penuh keikhlasan sehingga penulis dapat menyelesaikan studinya hingga mencapai gelar Sarjana Sains.
2. Saudara penulis, **Christo Forus Sewa** dan **Thomas Raynaldo Agutisto, S.M.** yang telah membantu serta memberikan semangat dan motivasi kepada penulis selama proses perkuliahan.
3. **Bapak Dr. Firman, S.Si., M.Si.** selaku Ketua Departemen Matematika.
4. **Bapak Prof. Dr. Amir Kamal Amir, M.Sc.** selaku dosen tim penguji yang memberikan masukan dalam penyusunan skripsi ini.
5. **Seluruh dosen** dan **Staf Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin** yang telah membantu banyak dan memberikan ilmunya selama masa perkuliahan.

6. **Veronika Masseng** selaku saudara tak sedarah penulis yang telah senantiasa menjadi tempat pulang penulis, tempat berkeluh kesah penulis, dan senantiasa memberikan pelajaran, motivasi, serta semangat selama 10 tahun ini.
7. **Febriana Petra Tombang** dan **Auxilia Moga Layuk** yang telah banyak membantu dan menghibur penulis selama proses penulisan skripsi ini. Secara khusus untuk **Febriana Petra Tombang** yang telah banyak menemani dan membantu penulis selama proses penulisan skripsi ini.
8. **Koslet Squad** yang selama ini menghibur penulis lewat cerita-cerita yang ada, lewat kerandoman yang diciptakan, serta memberikan penulis semangat dan motivasi selama proses penulisan skripsi ini.
9. **Teman seperjuangan Matematika 2020** khususnya **Butuh Ayang** yang ternyata penulis tidak membutuhkan pacar selama proses penulisan skripsi ini. **Yelny Alisya Upa', Angri Dhea Insani,** dan **Elsah Maria Dafosha jauung** terima kasih untuk kebersamaannya dalam suka dan duka, terima kasih telah menjadi pendengar selama di kampus, dan terim kasih sudah mau selalu direpotkan oleh penulis.
10. **Teman-teman KKN Unhas Gelombang.110** khususnya Desa Salenrang Kabupaten Maros yang memberi warna serta pelajaran selama masa KKN dan **teman-teman OMK Gotong-gotong** yang senantiasa menjadi tempat belajar dan berbagi cerita.
11. **Untuk Christina Paulina S** terima kasih sudah bertahan sampai di titik ini. Kamu hebat, kamu kuat, kamu luar biasa. Selalu libatkan Tuhan dan ingat Orang Tua untuk langkah selanjutnya.

Sebagai manusia biasa yang tak luput dari kesalahan, penulis menyadari bahwa masih banyak yang perlu dibenahi dan disempurnakan dalam skripsi ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang konstruktif dari semua pihak.

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMISI

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Christina Paulinna S
NIM	: H011201071
Program Studi	: Matematika
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Nonekslusif** (***Non-exclusive Royalty-Free Right***) atas karya saya yang berjudul:

PENERAPAN ALGORITMA *ADVANCED ENCRYPTION STANDARD* (AES) PADA BEBERAPA MODE ENKRIPSI UNTUK KEAMANAN JAWABAN *COMPUTER BASED TEST* (CBT)

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak Universitas berhak menyimpan, mengalih-media/format-kan, mengelolah dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya,

Dibuat di Makassar pada 31 Mei 2024

Yang menyatakan

Christina Paulinna S

ABSTRAK

Penerapan *Computer Based Test* (CBT) telah berkembang pesat dalam beberapa tahun terakhir sebagai metode ujian yang efisien dan fleksibel. Namun, keamanan data dalam CBT menjadi perhatian utama terutama dalam mengamankan jawaban peserta. Kriptografi modern, seperti *Advanced Encryption Standard* (AES), dapat digunakan untuk meningkatkan keamanan data dalam CBT. Penelitian ini bertujuan untuk menerapkan algoritma AES pada beberapa mode enkripsi, yaitu *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB), untuk meningkatkan keamanan jawaban CBT. Metode penelitian yang digunakan adalah gabungan antara kualitatif dan kuantitatif. Metode kualitatif digunakan untuk menganalisis struktur aljabar dari AES dan keempat mode enkripsi tersebut. Sedangkan metode kualitatif digunakan untuk menganalisis keamanan dan efisiensi masing-masing mode enkripsi. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi terbaik untuk mode enkripsi yang dapat digunakan dalam sistem CBT untuk keamanan jawaban.

Kata kunci: CBT, Kriptografi, AES, ECB, CBC, CFB, OFB.

Judul : Penerapan Algoritma *Advanced Encryption Standard* (AES) Pada Beberapa Mode Enkripsi Untuk Keamanan Jawaban *Computer Based Test* (CBT)
Nama : Christina Paulinna S
NIM : H011201071
Program Studi : Matematika

Universitas Hasanuddin

ABSTRACT

The rapid development of Computer Based Test (CBT) implementation has been notable in recent years as an efficient and flexible examination method. However, data security in CBT remains a major concern, particularly in securing participants' answers. Modern cryptography, such as the Advanced Encryption Standard (AES), can be utilized to enhance data security in CBT. This research aims to implement the AES algorithm in several encryption modes, namely Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB), to improve the security of CBT answers. The research methodology combines qualitative and quantitative approaches. Qualitative methods are used to analyze the algebraic structure of AES and the four encryption modes. Meanwhile, quantitative methods are employed to analyze the security and efficiency of each encryption mode. The results of this research are expected to provide the best recommendations for encryption modes that can be used in CBT systems to secure answers.

Keywords: *CBT, cryptography, AES, ECB, CBC, CFB, OFB.*

Title : *Implementation Of The Advanced Encrytion Standard (AES) Algorithm In Several Encryption Modes for Securing Computer Based Test (CBT) Answers*
Name : *Christina Paulinna S*
Student ID : *H011201071*
Study Program : *Mathematics*

DAFTARI ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi dan komunikasi sudah berkembang sedemikian pesat hampir di setiap sendi-sendi kehidupan masyarakat. Zaman sekarang perkembangan teknologi telah memberikan pengaruh yang besar di berbagai aspek kehidupan seperti dalam bidang politik, ekonomi, kebudayaan seni, sampai dibidang pendidikan. Dalam bidang pendidikan, teknologi memberikan manfaat pada peningkatan kualitas, kecepatan, kepraktisan, dan juga kemudahan. Penggunaan komputer merupakan salah satu contoh yang memberikan manfaat dalam segi kualitas, kecepatan, kepraktisan, dan kemudahan dalam dunia pendidikan. Contoh yang dapat dilihat pada zaman sekarang ialah penggunaan media kertas pada saat ujian (*paper based test*) lebih beralih ke arah komputerisasi atau biasa disebut *Computer Based Test* (CBT).

Computer Based Test (CBT) merupakan sistem pelaksanaan ujian yang menggunakan komputer sebagai media utama yang dilakukan secara online dan sistemnya dikelola oleh server. Pada umumnya, melaksanakan ujian dengan CBT ini memberikan soal dalam bentuk elektrik yang mampu menciptakan pengujian yang lebih efektif jika dibandingkan dengan tes yang dikerjakan menggunakan kertas (*paper based test*).

Sajati, Ayuningtyas, dan Kholistyanto (2017), penggunaan *eigenface* dapat memberikan keuntungan dalam menjaga keamanan CBT dengan menerapkan sistem deteksi wajah pada pengguna yang mengakses CBT. Pada tahun 2020, Hernowo, Seta, dan Widi melakukan penelitian mengenai keamanan CBT dengan metode kriptografi AES serta teknik dari steganografi EoF dan pada tahun yang sama, Rosmasari, F. Agus, dan F. Khairumman juga melakukan penelitian terhadap aplikasi CBT dengan menggunakan algoritma blowfish.

Dari beberapa penelitian yang telah dilakukan, dapat disimpulkan bahwa peningkatan keamanan terhadap jawaban peserta saat melaksanakan ujian berbasis komputer dapat dilakukan dengan berbagai metode agar tidak adanya campur tangan dari pihak yang tidak bersangkutan saat ujian berlangsung. Sehingga penelitian ini akan membahas mengenai bagaimana algoritma *Advanced Standard Encryption* (AES) diterapkan pada beberapa mode enkripsi untuk menjaga keamanan jawaban peserta agar tidak adanya campur tangan pihak ketiga.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, maka diperoleh rumusan masalah sebagai berikut :

1. Bagaimana menerapkan algoritma AES pada CBT?
2. Bagaimana merancang algoritma AES terhadap mode enkripsi pada CBT ?
3. Bagaimana perbandingan keamanan masing-masing mode enkripsi?

1.3 Batasan Masalah

Adapun batasan masalah yang diteliti sehingga jangkauan penelitian tidak akan melewati permasalahan sebenarnya :

1. Aplikasi CBT menggunakan soal pilihan ganda.
2. AES-128 bit dengan panjang sandi maksimal 16 karakter.
3. Kriptografi simetris.
4. Mode enkripsi yang digunakan yaitu ECB, CBC, CFB, dan OFB.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah dijelaskan, maka penelitian ini bertujuan untuk :

1. Mampu mengimplementasikan penerapan algoritma AES pada CBT.
2. Mampu mengimplementasikan rancangan algoritma AES terhadap mode enkripsi pada CBT.
3. Mengetahui perbedaan keamanan terhadap mode enkripsi pada CBT.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat dimanfaatkan dalam pengamanan hasil ujian berbasis komputer dengan menggunakan algoritma AES pada beberapa

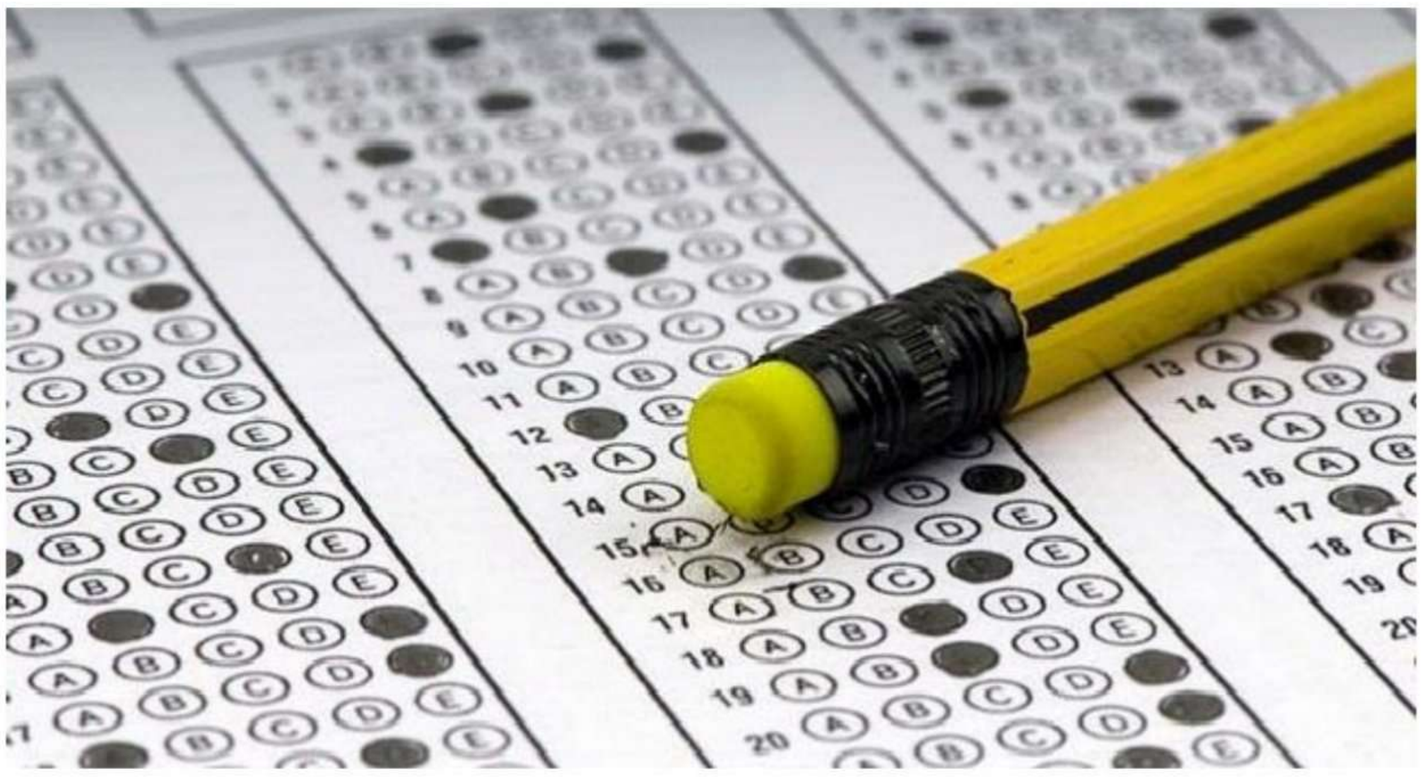
mode enkripsi, sehingga hasil jawaban tidak mengalami perubahan dan keaslian tetap terjaga.

BAB II

TINJAUAN PUSTAKA

2.1 Ujian

Ujian merupakan salah satu langkah yang dilakukan guna mengetahui dan atau menilai hasil belajar peserta didik. Tidak hanya peserta didik, ujian atau tes juga berlaku untuk para calon mahasiswa yang akan melanjutkan pendidikan ke perguruan tinggi negeri maupun swasta. Menurut Sugiono (1996), ujian adalah kegiatan untuk mengetahui totalitas para pihak yang melaksanakan ujian yang tidak terpisahkan dari tes.



Gambar 2.1 Ujian menggunakan kertas

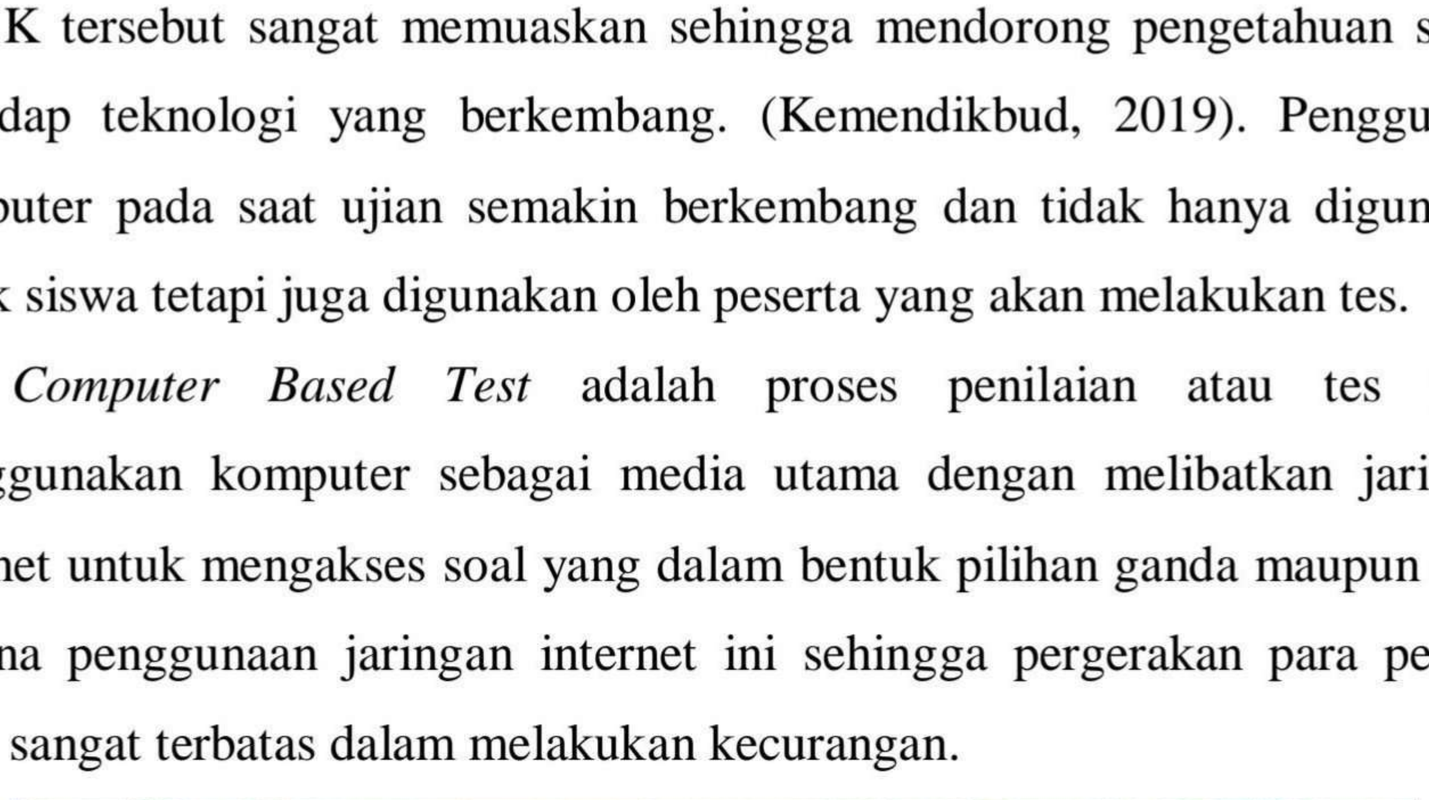
Sumber : kompas

Pada mulanya ujian atau tes yang diberikan umumnya menggunakan kertas sebagai lembar jawaban. Penggunaan kertas pada saat ujian atau *Paper Based Test* memiliki tingkat kecurangan yang lebih tinggi yang dilakukan oleh peserta ujian. Seiring berkembangnya teknologi, *Paper Based Test* (PBT) kemudian dialihkan ke komputerisasi atau penggunaan komputer pada saat melakukan tes atau umumnya dikenal dengan istilah *Computer Based Test* (CBT).

2.2 *Computer Based Test* (CBT)

Ujian Nasional Berbasis Komputer (UNBK) pertama kali diselenggarakan pada tahun 2014 secara online dan dibatasi di SMP Indonesia Singapura dan SMP Indonesia Kuala Lumpur. Hasil dari penyelenggaraan UNBK tersebut sangat memuaskan sehingga mendorong pengetahuan siswa terhadap teknologi yang berkembang. (Kemendikbud, 2019). Penggunaan komputer pada saat ujian semakin berkembang dan tidak hanya digunakan untuk siswa tetapi juga digunakan oleh peserta yang akan melakukan tes.

Computer Based Test adalah proses penilaian atau tes yang menggunakan komputer sebagai media utama dengan melibatkan jaringan internet untuk mengakses soal yang dalam bentuk pilihan ganda maupun esai. Karena penggunaan jaringan internet ini sehingga pergerakan para peserta ujian sangat terbatas dalam melakukan kecurangan.



Gambar 2.2 Ujian berbasis komputer

Sumber : Pendidikan Kedokteran

Menurut Azhar Arsyad (2014: 93) terdapat proses instruksional yang harus diikuti dalam menjadikan komputer sebagai media pembelajaran, diantaranya :

1. Merencanakan, mengatur, dan mengorganisasikan jadwal pengajaran.
2. Melakukan penilaian peserta ujian
3. Mengumpulkan data peserta ujian
4. Melakukan analisis statistik mengenai data ujian

5. Membuat catatan perkembangan setelah ujian

Sama halnya dengan *Paper Based Test*, ujian berbasis komputer juga memiliki keuntungan. Menurut Schreyer Institute, keuntungan menggunakan CBT dalam penilaian adalah :

1. *Inclusion of multimedia; Graphics, short video clips or sound files can be included in question stems, responses or feedback.*
2. *Item format; CBT allows for item types that can't be processed by scanning paper bubble sheets, such as "check all that apply".*
3. *Reduce paper costs; Computer-based test for large classes avoid what can be a substantial cost in producing paper tests.*
4. *Scoring; Many item types can be automatically scored.*

Dari beberapa keuntungan di atas, dapat disimpulkan bahwa CBT memiliki bentuk soal lebih menarik karena disampaikan secara multimedia, tidak menggunakan pena dan kertas, mengurangi biaya, uji pengetahuan skor valid, menghemat waktu, serta lebih cepat dalam pengambilan keputusan sebagai hasil dari pelaksanaan tes.

Laporan dan hasil tes CBT lebih cepat diolah, tetapi tidak menutup kemungkinan terjadi berbagai kecurangan yang tidak dapat dihindari. Oleh karena itu, sistem CBT yang dibuat harus memenuhi standar untuk menjamin keamanan pada setiap ancaman yang akan terjadi.

2.3 Kriptografi

Menurut Bruce Scheiner, kriptografi adalah ilmu pengetahuan dan seni menjaga pesan agar tetap aman. Pada peradaban Mesir dan Romawi, secara sederhana konsep kriptografi telah digunakan. Kriptografi merupakan ilmu yang mempelajari tentang cara untuk menjaga kerahasiaan pesan atau data agar tetap aman saat dikirimkan dan dapat sampai ke penerima tanpa ada gangguan dari pihak ketiga. Untuk menjaga kerahasiaan data, kriptografi memiliki beberapa prinsip, diantaranya :

1. *Confidentiality* (kerahasiaan) yaitu layanan agar data atau pesan yang dikirimkan dapat terjaga kerahasiaannya dan tidak diketahui oleh pihak-pihak yang tidak memiliki izin.
2. *Integrity* (keutuhan data) yaitu layanan yang dapat mengetahui jika data tersebut telah dimanipulasi oleh pihak lain.
3. *Authentication* (keotentikan) yaitu layanan yang mengidentifikasi pihak-pihak yang terlibat dalam pengiriman data dan keaslian data.
4. *Non-repudiation* (anti penyangkalan) yaitu layanan untuk mencegah pihak lain yang mengaku bahwa pesan tersebut berasal dari dirinya.

Sedangkan komponen dari kriptografi yaitu :

1. *Plaintext*, pesan asli yang dapat dibaca dan yang akan diubah untuk dikirimkan ke penerima.
2. *Ciphertext*, pesan *plaintext* yang telah diubah dan diacak sehingga tidak dapat dibaca.
3. *Key*, kunci untuk membuka *ciphertext* sehingga berubah menjadi *plaintext* begitupun sebaliknya.
4. *Algorithm*, metode yang digunakan untuk mengubah pesan dari *plaintext* ke *ciphertext* (enkripsi) atau dari *ciphertext* ke *plaintext* (dekripsi).

2.3.1 Kriptografi Klasik

Kriptografi klasik adalah algoritma kriptografi yang melakukan enkripsi dan dekripsi pada setiap karakter pesan dengan menggunakan kunci simetris dan menyandikan pesan dengan teknik substitusi atau transposisi. Salah satu contoh kriptografi klasik yaitu vigenere cipher.

Vigenere cipher pertama kali ditemukan oleh diplomat Perancis yang bernama Blaise de Vigenere (1523 – 1596) pada tahun 1586. Dalam mengenkripsi pesan, vigenere cipher menggunakan tabel 26 huruf alfabet standar yang dimulai dari A – Z. Kunci pada vigenere cipher dipakai berulang kali sebanyak pesan yang akan dienkripsi. Berikut ini rumus enkripsi dan dekripsi vigenere cipher :

Enkripsi :

$$C_i = P_i + K_i \; mod \; 26 \tag{2.1}$$

Dekripsi :

$$P_i = C_i - K_i \; mod \; 26 \tag{2.2}$$

C_i merupakan huruf ke-i pada teks tersandi, P_i merupakan huruf ke-i pada *plaintext*, K_i merupakan huruf ke-i pada kata kunci, dan mod adalah operasi modulus (sisa pembagian).

Tabel 2.1 Alfabetik 26 huruf

0	1	2	3	4	5	6	7	8
A	B	C	D	E	F	G	H	I
9	10	11	12	13	14	15	16	17
J	K	L	M	N	O	P	Q	R
18	19	20	21	22	23	24	25	
S	T	U	V	W	X	Y	Z	

Contoh :

Plaintext = SAYA MAU MAKAN KUE

Kunci = BOLU

Enkripsi : $C_i = P_i + K_i \; mod \; 26$

C_1 = S + B mod 26 = 18 + 1 mod 26 = 19 = T

C_2 = A + O mod 26 = 0 + 14 mod 26 = 14 = O

C_3 = Y + L mod 26 = 24 + 11 mod 26 = 9 = J

C_4 = A + U mod 26 = 0 + 20 mod 26 = 20 = U

C_5 = M + B mod 26 = 12 + 1 mod 26 = 13 = N

C_6 = A + O mod 26 = 0 + 14 mod 26 = 14 = O

C_7 = U + L mod 26 = 20 + 11 mod 26 = 5 = F

C_8 = M + U mod 26 = 12 + 20 mod 26 = 6 = G

C_9 = A + B mod 26 = 0 + 1 mod 26 = 1 = B

C_{10} = K + O mod 26 = 10 + 14 mod 26 = 24 = Y

C_{11} = A + L mod 26 = 0 + 11 mod 26 = 11 = L

C_{12} = N + U mod 26 = 13 + 20 mod 26 = 33 = H

Dari hasil enkripsi diperoleh *Ciphertext*nya adalah TOJU NOF GBYLH

Dekripsi : $P_i = C_i - K_i \; mod \; 26$

P_1 = T – B mod 26 = 19 – 1 mod 26 = 18 = S

P_2 = O – O mod 26 = 14 – 14 mod 26 = O = A

P_3 = J – L mod 26 = 9 – 11 mod 26 = 24 = Y

P_4 = U – U mod 26 = 20 – 20 mod 26 = 0 = A

P_5 = N – B mod 26 = 13 – 1 mod 26 = 12 = M

P_6 = O – O mod 26 = 14 – 14 mod 26 = 0 = A

P_7 = F – L mod 26 = 5 – 11 mod 26 = 20 = U

P_8 = G – U mod 26 = 6 – 20 mod 26 = 12 = M

P_9 = B – B mod 26 = 0 – 0 mod 26 = 0 = A

P_{10} = Y – O mod 26 = 24 – 14 mod 26 = 10 = K

P_{11} = L – L mod 26 = 0 – 0 mod 26 = 0 = A

P_{12} = H – U mod 26 = 7 – 20 mod 26 = 13 = N

Dari hasil dekripsi dapat diperoleh kembali *plaintext*nya SAYA MAU MAKAN.

2.3.2 Kriptografi Modern

Proses enkripsi pada algoritma kriptografi modern dilakukan pada bit-bit data. Maka, seluruh data yang bersifat digital dapat dienkripsi menggunakan algoritma kriptografi modern.

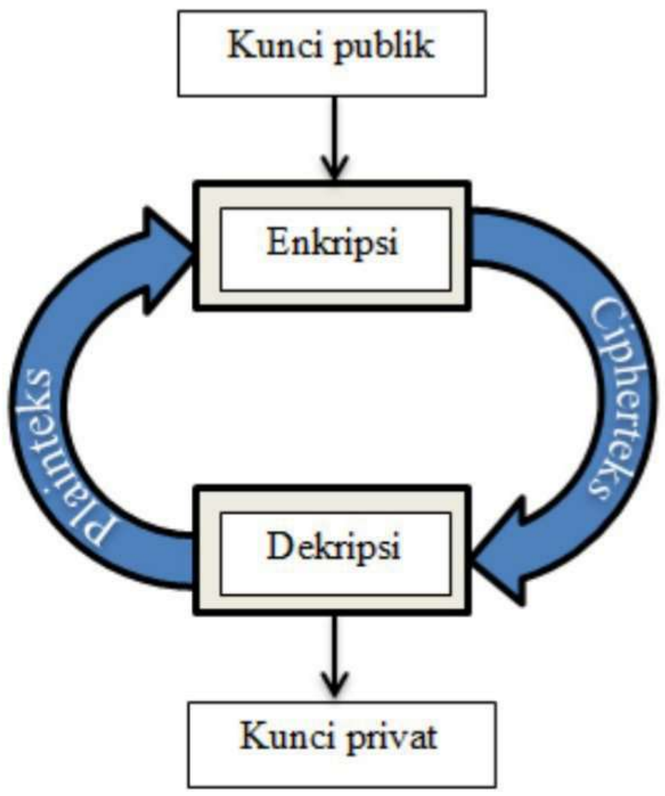
Algoritma kriptografi modern terbagi menjadi dua berdasarkan kunci yang digunakan dalam proses enkripsi dan dekripsi yaitu kriptografi kunci asimetris dan kriptografi kunci simetris.

a. Kriptografi Kunci Asimetris (Kunci Publik)

Algoritma kunci asimetris adalah algoritma kriptografi yang mempunyai sepasang kunci berbeda untuk melakukan enkripsi dan dekripsi. Algoritma kunci asimetris juga dikenal dengan nama algoritma kunci publik. Keamanan algoritma kunci publik berbasis pada kompleksitas komputasi yang menggunakan fungsi satu arah (*one way function*) yaitu lebih mudah melakukan perhitungan *f(x)* daripada mencari

nilai x dari fungsi *f(x)* yang diketahui. Yang berarti, untuk mencari nilai x dari fungsi *f(x)* membutuhkan waktu lama untuk memecahkannya karena sangat kompleks. Seiring bertambah panjangnya kunci x maka kompleksitas komputasi akan meningkat secara eksponensial.

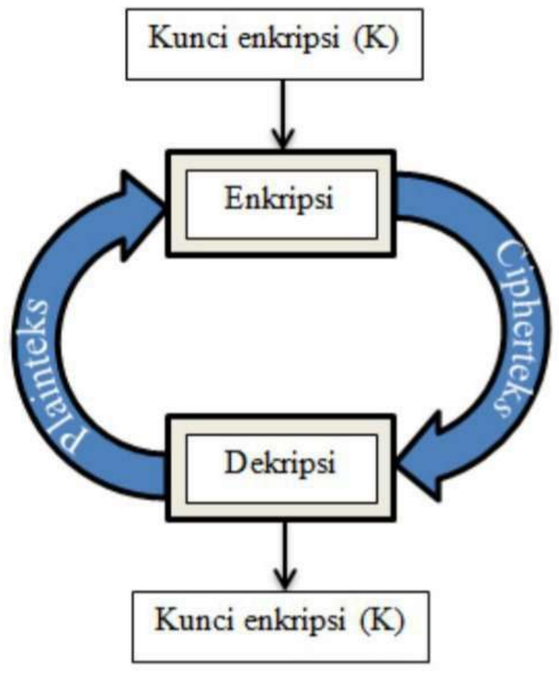
Keamanan pada algoritma kunci publik didasarkan pada kompleksitas komputasi yang perkembangan teknologi seperti komputer tercepat masih belum mampu untuk menandingi kompleksitas yang telah dibangun dan memerlukan waktu yang sangat lama untuk memecahkan algoritma tersebut.



Gambar 2.3 Proses enkripsi dan dekripsi kriptografi asimetris

b. Kriptografi Kunci Simetris

Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi.



Gambar 2.4 Proses enkripsi dan dekripsi kriptografi simetris

Berdasarkan bit yang dienkripsi, kriptografi simetris dibagi menjadi dua bagian yaitu algoritma aliran (*stream cipher*) dan algoritma blok (*block cipher*).

- *Stream Cipher*

  Stream cipher merupakan algoritma kriptografi yang beroperasi pada *plaintext* atau *ciphertext* dalam bentuk bit tunggal sehingga rangkaian bit dienkripsikan atau didekripsikan secara bit per bit.

  Secara umum, *stream cipher* membangkitkan aliran kunci dari kunci yang dimasukkan oleh pengguna. Ketika aliran kunci telah dibangkitkan, maka proses enkripsi dan dekripsi dilakukan melalui operasi XOR antara bit aliran kunci dengan bit *plaintext* atau *ciphertext*. Semakin acak keluaran yang dihasilkan dari proses pembangkit aliran kunci, maka *ciphertext* yang dihasilkan juga akan semakin sulit dipecahkan.

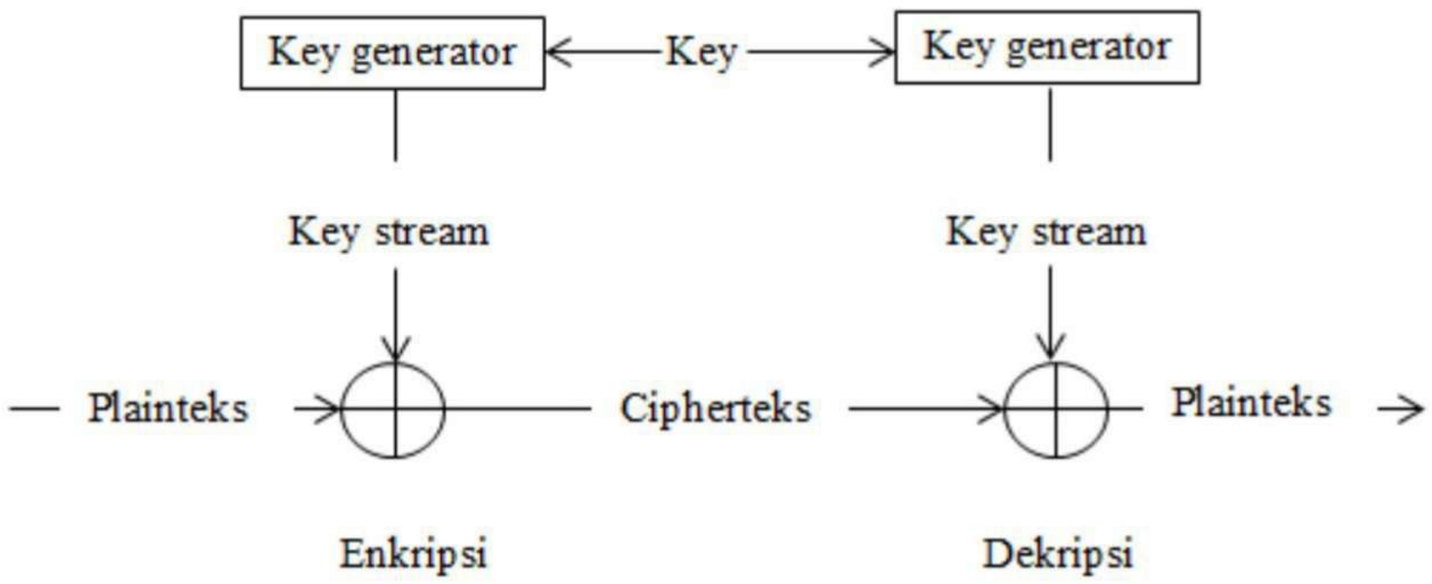
  Stream cipher menggunakan karakter sebelumnya sebagai kunci.

  Fungsi matematikanya terdiri dari :

  $$\boldsymbol{C} = (\boldsymbol{P} + \boldsymbol{K})\ mod\ n \quad (2.3)$$

  $$\boldsymbol{P} = (\boldsymbol{C} + \boldsymbol{K})\ mod\ n \quad (2.4)$$

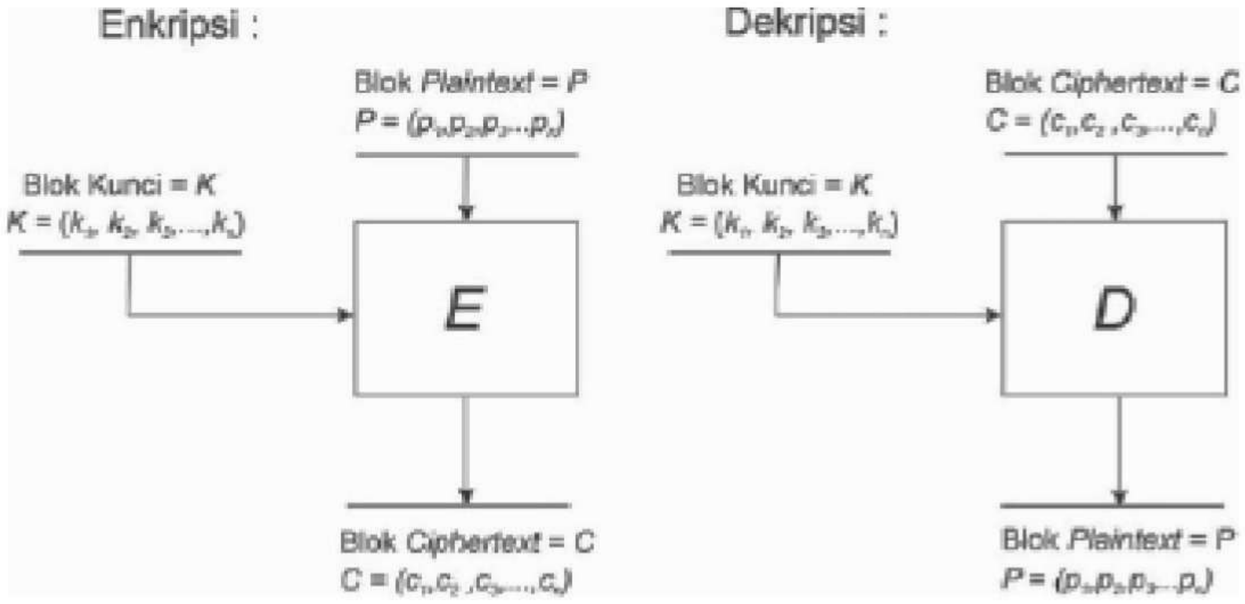
  P = *plaintext* **K** = kunci **n** = jumlah karakter **C** = *ciphertext*



Gambar 2.5 Skema *Stream Cipher*

- *Block Cipher*

Block Cipher merupakan kriptografi modern yang membagi *plaintext* menjadi beberapa blok bit untuk dienkripsikan menggunakan kunci agar menjadi blok-blok bit *ciphertext*. Proses enkripsi dan dekripsi dilakukan menggunakan operasi XOR yang menjadi perantara *plaintext*, *ciphertext*, dan kunci.



Gambar 2.6 Skema *Block Cipher*

Sumber : Fauzi & Wellem (2021)

Dari skema *block cipher* tersebut dapat diperoleh bahwa

Blok *plaintext* (p) dinyatakan sebagai

$$P = (p_1, p_2, \ldots\ldots, p_m) \quad (2.5)$$

Blok *ciphertext* (c) dinyatakan sebagai

$$C = (c_1, c_2, \ldots\ldots, c_m) \quad (2.6)$$

Kunci (k) dinyatakan sebagai

$$K = (k_1, k_2, \ldots\ldots, k_m) \tag{2.7}$$

Sehingga proses enkripsi adalah

$$EK(P) = C \tag{2.8}$$

Dan proses dekripsi adalah

$$DK(C) = P \tag{2.9}$$

Advanced encryption standard (AES) adalah algoritma blok cipher kunci simetris yang popular, aman, dan banyak digunakan. AES digunakan secara resmi sebagai standar teknologi enkripsi yang direkomendasikan di Amerika Serikat. AES beroperasi menggunakan ukuran blok 128 bit dan kunci simetris dengan panjang 128, 192, dan 256 bit. Dalam proses enkripsi AES terdapat aritmatika *finite field* yang akan digunakan pada S-Box dan mixcolumn.

2.4 *Finite Field* (Lapangan Hingga)

Finite field atau *galois field* merupakan *field* dengan jumlah elemen terbatas yang dinotasikan sebagai $\mathbb{F}_q$ atau $\mathbb{GF}(q)$. q merupakan bilangan prima atau kelipatan dari bilangan prima, $q = p$ atau $q = p^m$, yang berarti bilangan prima p disebut sebagai karakteristik dari *finite fields* dan m adalah bilangan bulat positif. *Finite field* ditentukan oleh jumlah elemen. Jika $m = 1$, maka $\mathbb{F}_{p^m}$ disebut *prime field.* Sedangkan jika $m \geq 2$, maka $\mathbb{F}_{p^m}$ disebut *extension field.*

Extension field $\mathbb{GF}(p^m)$ dihasilkan oleh suatu fungsi polinomial derajat ke-m yang tidak tereduksi. Polinomial merupakan hasil penjumlahan variabel-variabel yang koefisien dan nilai pangkatnya berbeda-beda, dengan x sebagai notasi variabel. Maksud dari polinomial tidak tereduksi adalah polinomial yang perkalian polinomnya tidak dapat difaktorkan. Pada komputer, koefisien yang digunakan hanya 0 atau 1 sehingga koefisien yang ada pada polinomial tidak digunakan. Maka hal tersebut dapat direpresentasikan dalam bentuk biner. Misalkan terdapat polinomial $x^6 + x^4 + x^3 + x^2 + 1$, maka dalam komputer, polinomial tersebut disimpan dalam byte sebagai berikut

	x^6		x^4	x^3	x^2		1	Polinomial
0	1	0	1	1	1	0	1	Koefisien
7	6	5	4	3	2	1	0	

Binary field merupakan *finite field* dengan orde 2^m. *Binary field* dibentuk dari representasi *polynomial basis*. *Binary polynomial* $\mathbb{F}_{2^m}$ dengan koefisien $\mathbb{F}_2 = \{0,1\}$ dengan derajat terbesar (m – 1) memiliki elemen

$$\mathbb{F}_{2^m} = \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2a_1x + a_0 : a_i \in \{0,1\}\} \quad (2.10)$$

dengan koefisien a_i berada pada $\mathbb{GF}$ *(p)*. *Binary polynomial* memiliki *irreducible binary polynomial* berderajat *m*. *Irreducible binary polynomial* merupakan polinomial tak tereduksi yang tidak habis dibagi dengan polinomial biner yang memiliki derajat lebih kecil dari *m*.

Pada AES, lapangan hingga memiliki 256 elemen yang dilambangkan dengan $\mathbb{GF}(2^8)$ yang dapat direpresentasikan dengan satu byte untuk transformasi S-Box dan mixcolumn.

Jika orde pada lapangan hingga bukan bilangan prima, dan 2^8 juga bukan bilangan prima, maka operasi penjumlahan dan perkalian tidak dapat direpresentasikan dengan bilangan bulat modulo 2^8. Pada lapangan hingga 2^8 yang digunakan dalam AES, setiap elemen $A \in \mathbb{GF}(2^8)$ direpresentasikan sebagai

$$A(x) = a_7x^7 + \cdots + a_1x + a_0, \quad a_i \in GF(2) = \{0,1\} \quad (2.11)$$

Setiap polinomial $\mathbb{GF}(2^8)$ dapat disimpan dalam bentuk digital sebagai vektor 8-bit

$$A = a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0$$

Terdapat empat operasi pada polinomial, yaitu penjumlahan, pengurangan, perkalian, dan invers. Dengan menggunakan *binary field* yaitu *finite field* pada orde 2^m, diberikan contoh sebagai berikut:

$$P(x) = x^4 + x + 1\ , A(x) = x^3 + x^2 + 1, \text{dan } B(x) = x^2 + x$$

- Operasi penjumlahan

$$C(x) = A(x) + B(x) = \sum_{i=0}^{m-1} c_i x^i,\ c_i \equiv a_i + b_i \ mod\ 2 \qquad (2.12)$$

$A(x) = x^3 + x^2 + \quad 1$

$B(x) = \qquad x^2 + x$

$C(x) = x^3 + x + 1$

- Operasi pengurangan

$$C(x) = A(x) - B(x) = \sum_{i=0}^{m-1} c_i x^i,\ c_i \equiv a_i - b_i \ mod\ 2 \qquad (2.13)$$

$A(x) = x^3 + x^2 + \quad 1$

$B(x) = \qquad x^2 + x$

$C(x) = x^3 + x + 1$

Pada $\mathbb{F}_2$, -1 = 1, maka $-a = a \ \forall \in \ \mathbb{F}_{2^m}$

- Operasi perkalian

Misal $A(x), B(x) \ \in \mathbb{GF}(2)$ diperoleh

$$P(x) \equiv \sum_{i=0}^{m} p_i x^i,\ p_i \in \mathbb{GF}(2) \qquad (2.14)$$

merupakan polinomial tidak tereduksi. Operasi perkalian dua elemen dapat dihitung dengan

$$C(x) \equiv A(x)\ .\ B(x)\ mod\ P(x) \qquad (2.15)$$

$A \cdot B = (x^3 + x^2 + 1) \cdot (x^2 + x)$

$= x^5 + x^4 + x^4 + x^3 + x^2 + x$

$= x^5 + x^3 + x^2 + x \ \text{mod} \qquad x^4 + x + 1$

$x^4 + x + 1 \qquad \leftrightarrow \qquad x^4 = x + 1$

$$x^5 + x^3 + x^2 + x = x(x^4 + x^2 + x + 1)$$
$$= x(x+1) + x(x^2 + x + 1)$$
$$= x^2 + x + x^3 + x^2 + x$$
$$= x^3$$

sehingga diperoleh

$$A \cdot B = (x^3 + x^2 + 1) \cdot (x^2 + x) = x^3$$

$$A \cdot B = (1101)_2 \cdot (0110)_2 = (1000)_2$$

- Invers

  Jika diberikan lapangan hingga $\mathbb{GF}(2^m)$ dan polinomial tidak tereduksi $P(x)$, maka invers $A \in \mathbb{GF}(2^m)$ didefinisikan sebagai

  $$A^{-1}(x) \,.\, A(x) \; mod \; P(x) = 1 \qquad (2.16)$$

  $A^{-1} = x^2$ karena $(x^3 + x^2 + 1) \times x^2 \bmod (x^4 + x + 1) = 1$

  $$A^{-1} = (0100)_2$$

  Invers pada $\mathbb{GF}(2^8)$ adalah inti operasi dari transformasi Subbytes yang menggunakan S-Box. Tabel pada S-Box berisi semua nilai dari $\mathbb{GF}(2^8) \; mod \;\; P(x) = \; x^8 + x^4 + x^3 + x + 1$ dalam notasi heksadesimal.

Tabel 2.2 Invers tabel S-Box

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

Contoh :

Dengan melihat tabel 2.2, maka invers dari

$x^6 + x^3 + x^2 + x = (01001110)_2 = (4E)_{hex}$, dengan digit 4 sebagai baris dan digit E sebagai kolom, sehingga diperoleh

$(E9)_{hex} = (11101001)_2 = x^7 + x^6 + x^5 + x^3 + 1$ yang berarti

$(x^6 + x^3 + x^2 + x)\ .\ (x^7 + x^6 + x^5 + x^3 + 1) \equiv 1\ mod\ P(x)$

2.5 *Advanced Encryption Standard* (AES)

Pada tanggal 2 Januari 1997, NIST (*National Institute of Standards and Technology*) melakukan proses pemilihan untuk mengganti DES. *Advanced Encryption Standard* (AES) merupakan algoritma yang ditetapkan sebagai pengganti DES pada 12 September 1997. AES memiliki panjang blok 128 bit dengan panjang kunci yang dapat digunakan yaitu 128, 192, dan 256 bit.

AES merupakan cipher iterasi yang berdasarkan penjelasan di atas, jumlah putaran ditentukan oleh panjang kunci yang digunakan. Setiap blok dienkripsi dalam sejumlah putaran tertentu. Pada umumnya, untuk

mengenkripsi dokumen atau file digunakan AES blok 128-bit atau 16 karakter.

Tabel 2.3 Putaran Tiap Blok

Varian AES	**Panjang kunci (Nk words)**	**Ukuran blok (Nb words)**	**Jumlah putaran (Nr)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Kunci AES menggunakan proses berulang yang disebut dengan ronde. Terdapat operasi untuk mencari hasil dari matriks ronde ke-1 hingga ronde ke-(Nr – 1), dengan Nr adalah jumlah putaran. Pada proses enkripsi awalnya teks asli dibentuk sebagai sebuah *state*. *State* direpresentasikan sebagai berikut

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Inisialisasi *state* didefinisikan sebagai 16 bytes dari *plaintext* x, sehingga diperoleh

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

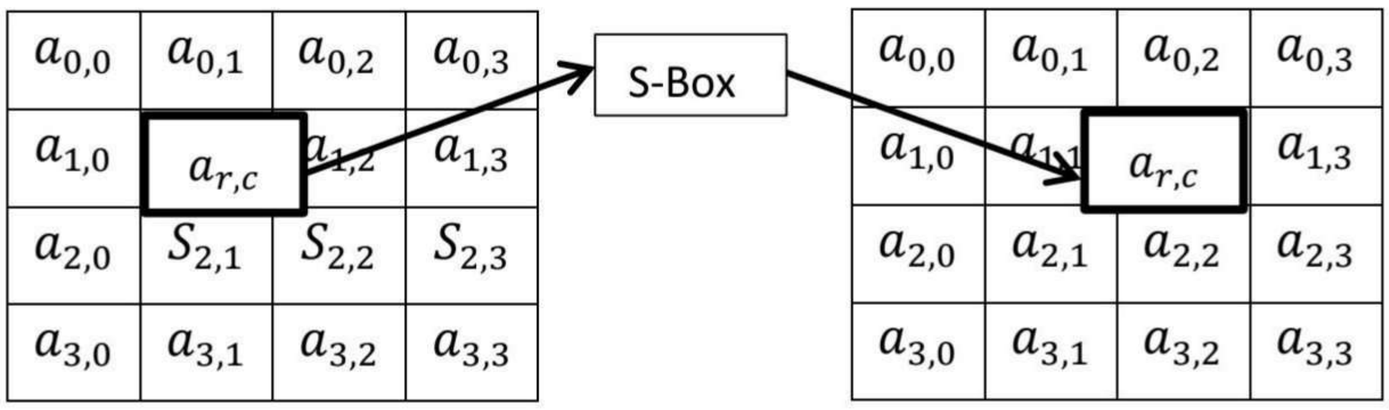
$\leftarrow$

x_0	x_1	x_2	x_3
x_4	x_5	x_6	x_7
x_8	x_9	x_{10}	x_{11}
x_{12}	x_{13}	x_{14}	x_{15}

Isi byte menggunakan notasi heksadesimal yang terdiri dari dua digit.

Terdapat 4 jenis transformasi pada AES yaitu :

1. *Subbytes* adalah menukar isi matriks atau bisa disebut dengan *Rijndael S-Box*.



Gambar 2.7 Skema *SubBytes*

Mengambil salah satu isi kotak matriks dan membandingkannya dengan angka di sebelah kiri sebagai baris dan angka di sebelah kanan sebagai kolom. Kemudian mengubah nilai dari baris dan kolom tersebut melalui *Rijndael S-Box*.

Tabel 2.4 S-Box

	Y															
X	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

Hal tersebut dilakukan secara berulang hingga seluruh blok cipher menjadi blok baru. Atau didefinisikan sebagai

$$S(A_i) = B_i \tag{2.17}$$

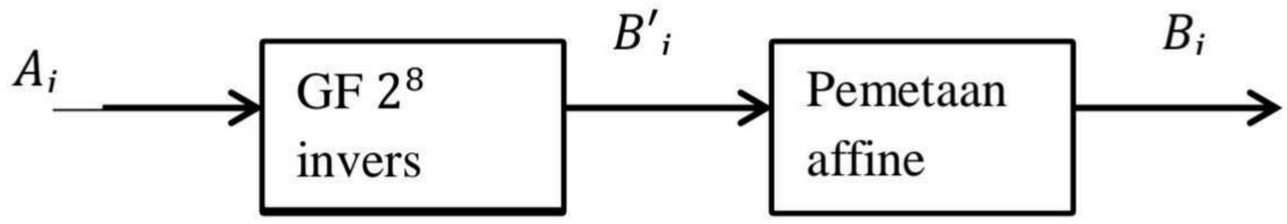
Tabel S-Box merupakan satu-satunya nonlinier pada AES yang menyatakan bahwa $subbytes\,(A) + subbytes\,(B) \neq subbytes\,(A + B)$, untuk dua *state* A dan B. Substitusi S-Box adalah pemetaan bijektif yaitu masing-masing $2^8 = 256$ elemen input yang mungkin dipetakan satu-satu ke satu elemen output.

Contoh :

Misalkan input byte ke S-Box, $A_i = (D8)_{hex}$ maka nilai substitusinya adalah

$$S((D8)_{hex}) = (61)_{hex}$$

adapun cara lain yang dapat digunakan untuk menentukan *subbytes*. Cara tersebut dapat dilihat sebagai matematis dua langkah seperti gambar berikut



Gambar 2.8 Operasi *subbytes* tanpa S-Box

Untuk setiap elemen A_i, invers dapat dihitung dengan

$$B'_i = A_i^{-1} \tag{2.18}$$

Dengan A_i dan B_i merupakan elemen dalam $GF(2^8)$ dengan polinomial tetap yang tidak dapat direduksi $P(x) = x^8 + x^4 + x^3 + x + 1$.

Untuk setiap elemen B'_i dikalikan dengan matriks bit konstan diikuti dengan penambahan vektor 8 bit konstan. Operasi ini direpresentasikan sebagai berikut

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} mod\ 2 \tag{2.19}$$

Perhatikan bahwa $B' = (b'_7, \ldots., b'_0)$ merupakan vektor dari representasi $B'_i(x) = A_i^{-1}(x)$

Contoh :

Dengan menggunakan contoh sebelumnya, $A_i = (11011000)_2$, tentukan invers dari A_i!

Jawab :

$A_i = (11011000)_2 = (D8)_{hex}$

Dengan melihat tabel 2.2, diperoleh

$A_i^{-1} = B_i' = (94)_{hex} = (10010100)_2$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} mod\ 2$$

$$\begin{pmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} mod\ 2$$

$$\begin{pmatrix} 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 2 \end{pmatrix} mod\ 2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = (0110\ 0001)_2 = (61)_{hex}$$

2. *Shift Rows* adalah proses menggeser setiap blok secara baris demi baris dengan aturan baris pertama tidak digeser, baris kedua digeser 1 *byte*, baris ketiga digeser 2 *byte*, dan baris keempat digeser 3 *byte* yang semua baris digeser ke kiri.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	Shift 0	→	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	Shift 1		$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,0}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	Shift 2		$a_{2,2}$	$a_{2,3}$	$a_{2,0}$	$a_{2,1}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	Shift 3		$a_{3,3}$	$a_{3,0}$	$a_{3,1}$	$a_{3,2}$

Gambar 2.9 Skema *Shift Rows*

3. *Mix Column* adalah mengalikan setiap elemen dari blok cipher pada matriks dengan perkalian titik, kemudian hasil perkalian tersebut dimasukkan ke dalam cipher blok baru. Blok matriks yang digunakan untuk proses perkalian yaitu

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Setiap kolom 4-byte hasil *shiftrows* akan dikalikan dengan matriks tersebut. Isi dari matriks merupakan konstan sehingga perkalian dan penjumlahan dilakukan pada $GF(2^8)$. Sebagai contoh, direpresentasikan empat byte pertama yang akan dikalikan dengan blok matriks di atas

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{pmatrix} a_{0,0} \\ a_{1,1} \\ a_{2,2} \\ a_{3,3} \end{pmatrix}$$

Output dari kolom kedua (C_4, C_5, C_6, C_7) dihitung dengan mengalikan empat byte selanjutnya $(a_{0,1}, a_{1,2}, a_{2,3}, a_{3,0})$ dengan matriks konstanta yang sama. Begitu seterusnya hingga diperoleh hasil dari *mixcolumn*.

Semua aritmatika yang menggunakan koefisien akan direpresentasikan dalam $GF(2^8)$. Untuk konstanta pada matriks, digunakan notasi heksadesimal sebagai berikut :

$$01 \rightarrow (0000\ 0001)_2 = 1$$

$$02 \rightarrow (0000\ 0010)_2 = x$$

$$03 \rightarrow (0000\ 0011)_2 = x + 1$$

Contoh :

Dengan menggunakan contoh di atas, maka diasumsikan input dari suatu kolom adalah

$a = (61, 61, \ldots., 61)$

dalam kasus ini, hanya dua operasi perkalian pada $GF(2^8)$ yang harus diselesaikan yaitu $02 \cdot 61$ dan $03 \cdot 61$ yang dapat dihitung dalam notasi polinomial berikut:

$02 \rightarrow (0000\ 0010)_2 = x$

$61 \rightarrow (0110\ 0001)_2 = x^6 + x^5 + 1$

$02 \cdot 61 = x \cdot (x^6 + x^5 + 1)$

$= x^7 + x^6 + x$

$03 \rightarrow (0000\ 0011)_2 = x + 1$

$61 \rightarrow (0110\ 0001)_2 = x^6 + x^5 + 1$

$03 \cdot 61 = (x + 1)(x^6 + x^5 + 1)$

$= x^7 + x^5 + x + 1$

Karena kedua nilai memiliki derajat lebih kecil dari 8, maka reduksi modulo $P(x)$ tidak diperlukan. Sehingga diperoleh output hasil penambahan pada $GF(2^8)$ sebagai berikut

$01 \cdot 61 = \quad x^6 + x^5 \quad + 1$

$01 \cdot 61 = \quad x^6 + x^5 \quad + 1$

$02 \cdot 61 = x^7 + x^6 + \quad x$

$03 \cdot 61 = x^7 + \quad x^5 + \quad x + 1$

$C_i = x^6 + x^5 + 1$

$= (0110\ 0001)_2 = (61)_{hex}$

$i = 0, \ldots., 15$

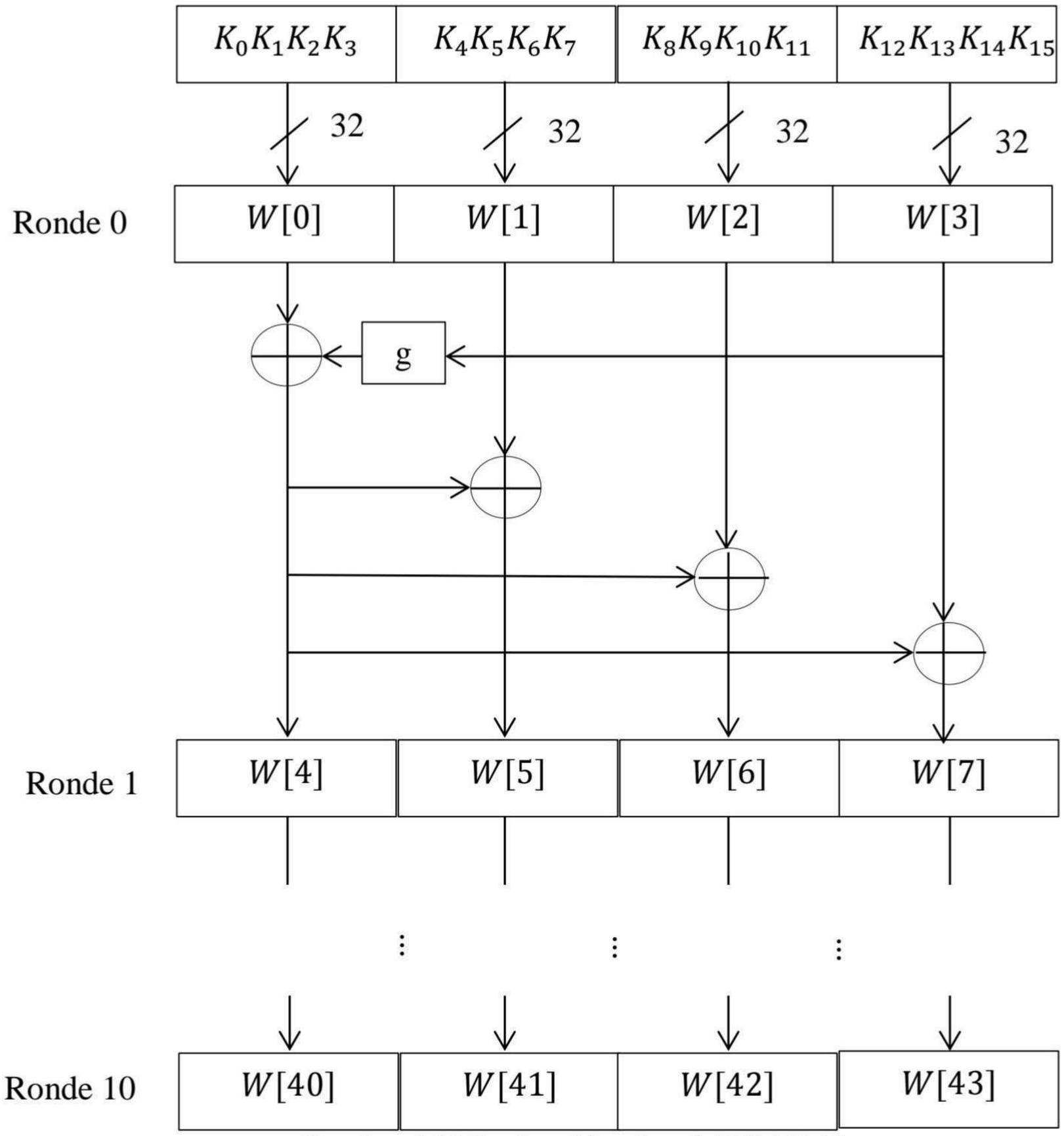
4. *Add Round Key* adalah kombinasi *ciphertext* dengan kunci enkripsi melalui operasi XOR. Kunci enkripsi yang digunakan merupakan subkunci. Jumlah subkunci yang digunakan sama dengan jumlah putaran ditambah satu. Dalam penelitian ini, AES yang digunakan yaitu AES 128-bit sehingga berdasarkan tabel 2.3 jumlah subkunci adalah 11. Subkunci tersebut disimpan dalam himpunan ekspansi kunci dengan elemen $W[0], \ldots., W[43]$. Pada ekspansi kunci terdapat dua operasi lain yang disebut dengan *rotword* dan *subword*. *Rotword* (B_0, B_1, B_2, B_3) melakukan pergeseran empat byte (B_0, B_1, B_2, B_3).

Rotword $(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0)$

sedangkan *subword* (B_0, B_1, B_2, B_3) menerapkan *subbytes* ke masing-masing empat byte (B_0, B_1, B_2, B_3)

Subword $(B_0, B_1, B_2, B_3) = (B_0', B_1', B_2', B_3')$

Kunci asli AES pada proses pembangkitan kunci dinotasikan sebagai $K_0, \ldots., K_{15}$. Sehingga subkunci dapat dihitung seperti gambar di bawah ini



Gambar 2.10 Pembangkitan kunci AES 128-bit

Dapat dilihat bahwa subkunci pertama K_0 merupakan kunci AES yang asli. Kunci tersebut kemudian disalin ke dalam empat elemen pertama himpunan kunci W. elemen paling kiri pada ronde pertama kunci W dapat dihitung sebagai berikut

$$W[4i] = W[4(i-1)] \oplus g(W[4i-1]) \tag{2.20}$$

$i = 1, \ldots, 10$ dan $g()$ merupakan fungsi nonlinier dengan input dan output masing-masing empat byte. sedangkan tiga kata tersisa dari subkunci dapat dihitung dengan cara

$$W[4i+j] = W[4i+j-1] \oplus W[4(i-1)+j] \tag{2.21}$$

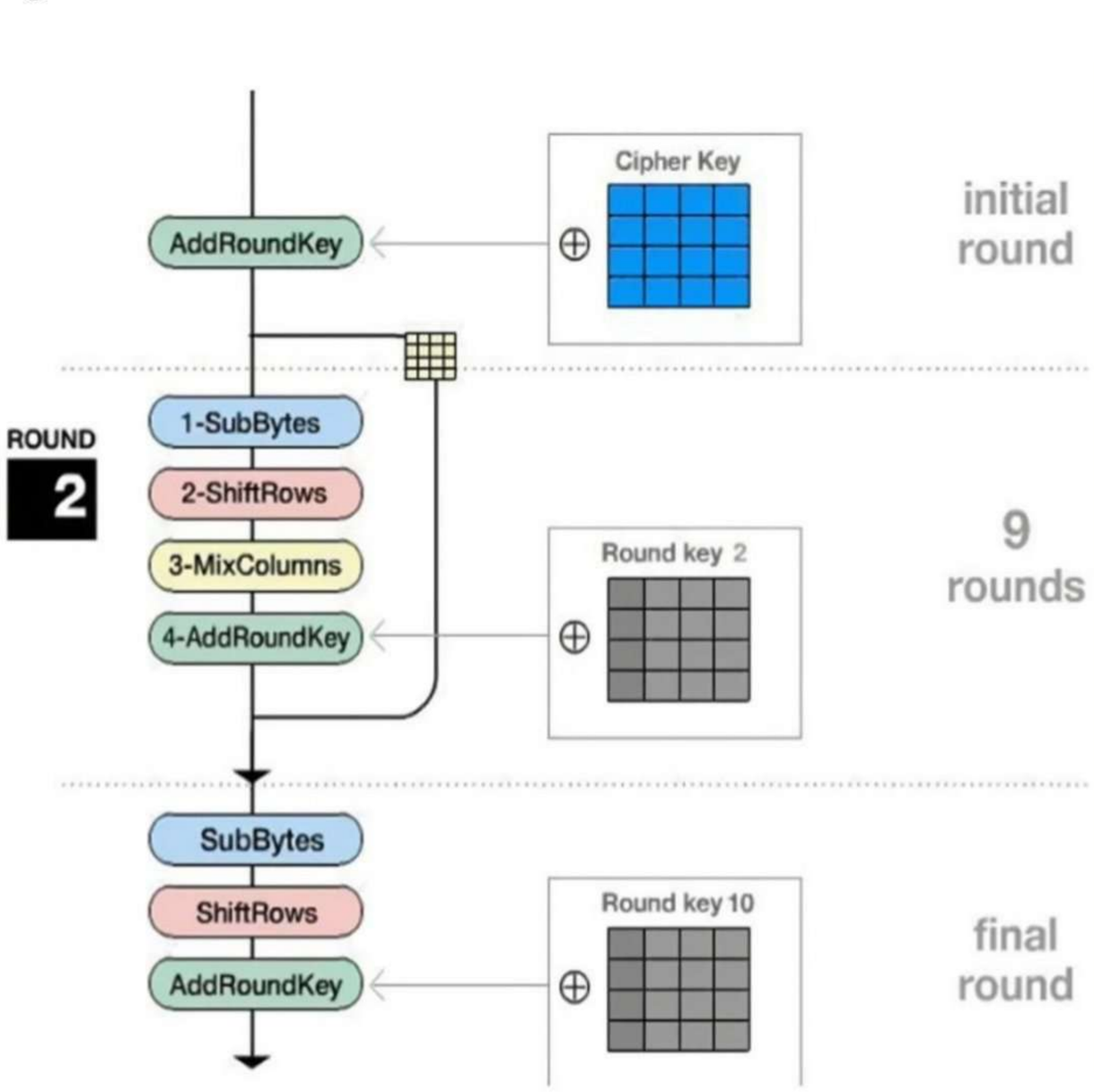
$i = 1, \ldots, 10$ dan $j = 1,2,3$.

Fungsi g() memutar empat byte inputnya kemudian melakukan substitusi S-Box berdasarkan byte dan menambahkan koefisien putaran Rcon ke dalamnya. Rcon merupakan elemen dari $GF(2^8)$ yang memiliki nilai 8 bit. Nilai Rcon dapat dilihat pada tabel di bawah ini

Tabel 2.5 Rcon (ronde)

Ronde	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Rcon[]	01	02	04	08	10	20	40	80	1b	36
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00

Berdasarkan penjelasan di atas, proses enkripsi AES dapat dilakukan seperti gambar di bawah ini



Gambar 2.11 Enkripsi algoritma AES
Sumber : Blogspot

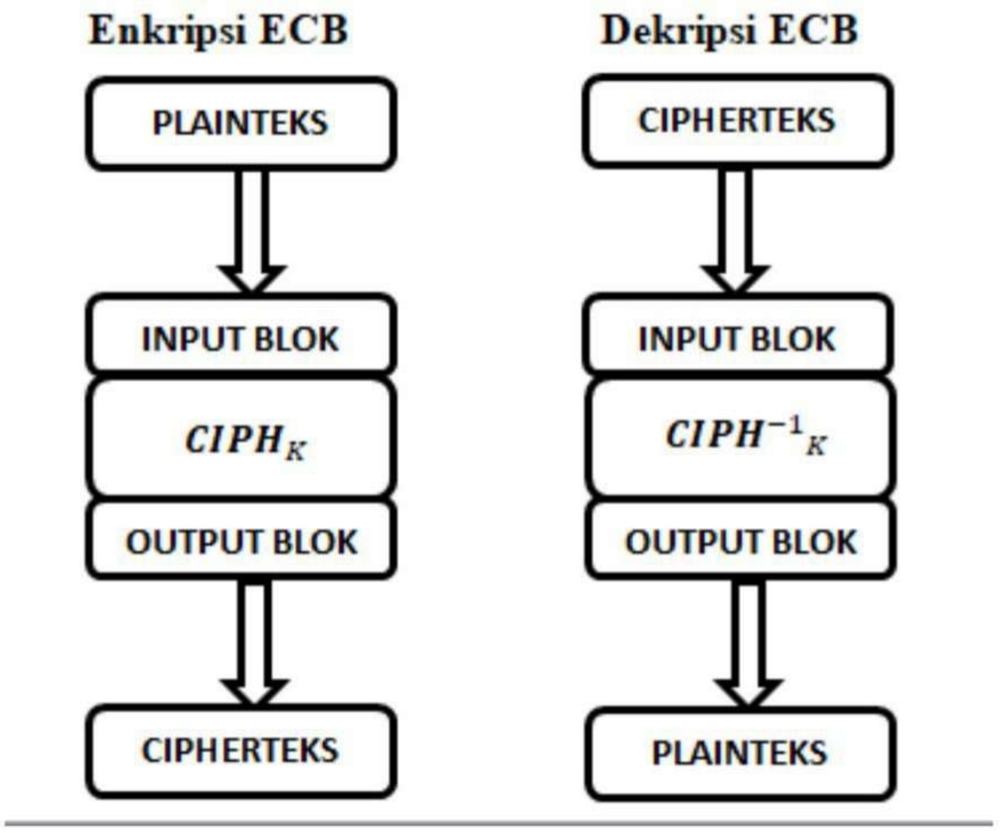
Selama proses enkripsi AES dilakukan, terdapat proses yang menggunakan tabel ASCII untuk mengubah *plaintext* dan *ciphertext* ke dalam bilangan heksadesimal

2.6 Mode Enkripsi AES

Dalam proses mengubah *plaintext* menjadi *ciphertext*, AES memiliki beberapa mode enkripsi yang dikembangkan dari empat mode enkripsi DES. Mode enkripsi ini termasuk dalam *block cipher* yang dapat digunakan untuk mengubah *plaintext* menjadi *ciphertext* dalam bentuk blok-blok tersandi. Mode enkripsi dari AES diantaranya *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), *Output Feedback* (OFB), *Counter* (CTR), *Counter with Cipher block chaining MAC* (CCM), dan *Galois Counter* (GCM).

2.6.1 *Electronic Code Book* (ECB)

Pada ECB, blok *plaintext* melalui proses enkripsi secara individual dan independen atau terpisah menjadi sebuah blok cipher. Mode ECB merupakan proses enkripsi yang paling sederhana sehingga blok pesan yang dienkripsi selalu dipetakan ke blok cipher yang sama dapat diserang dengan analisis statistik.

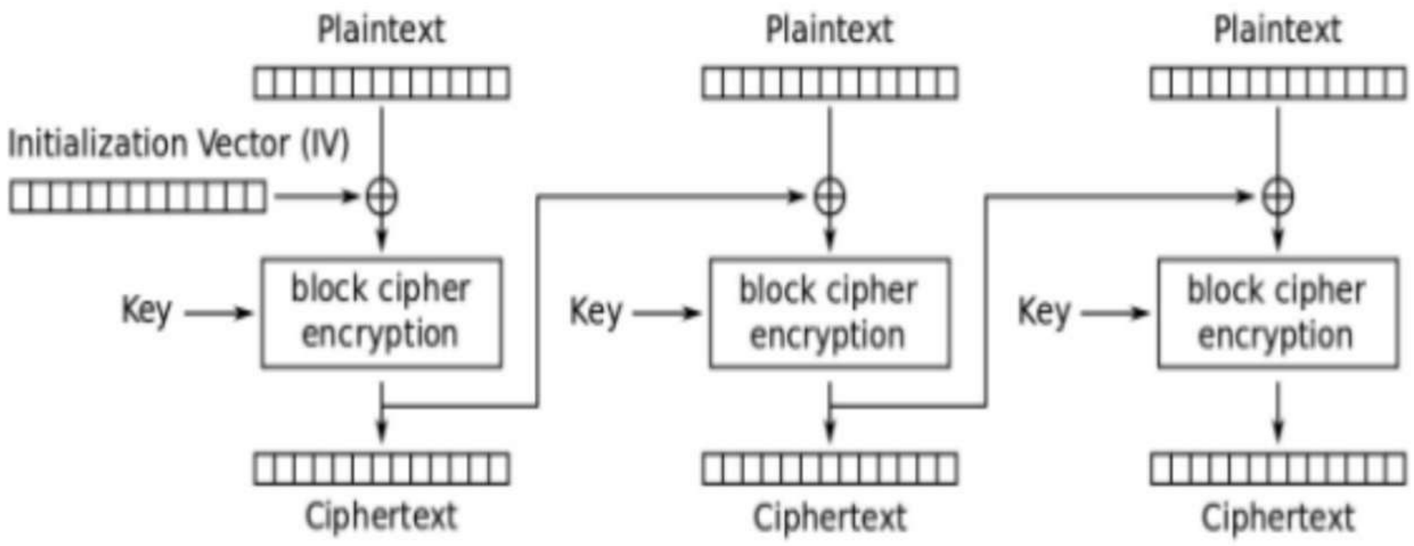


Gambar 2.12 Skema enkripsi dan dekripsi ECB

Mode ini sesuai dengan blok cipher yang digunakan. Jika terdapat *plaintext* x_1x_2 .... maka masing-masing x_i dienkripsi dengan kunci K yang sama, sehingga menghasilkan blok *ciphertext*.

2.6.2 *Cipher Block Chaining* (CBC)

Pada tahun 1976, Ehrsam, Meyer, Smith, dan Tuchman menemukan operasi CBC. Mode operasi CBC melakukan operasi XOR pada pesan pertama dengan sebuah vektor sebesar panjang blok sebelum masuk ke dalam algoritma *block cipher*. Dengan kata lain, setiap blok *ciphertext* bergantung pada semua blok *plaintext* yang diproses sampai titik tersebut. CBC menggunakan *Initialization vector* pada blok pertama. Kekurangan mode CBC adalah blok selanjutnya dapat diproses apabila blok sebelumnya sudah diproses sehingga tidak dapat dijalankan secara paralel. Penggunaan *padding* juga diperlukan untuk mode CBC dan ECB apabila pesan tidak sepanjang blok.



Gambar 2.13 Skema enkripsi CBC

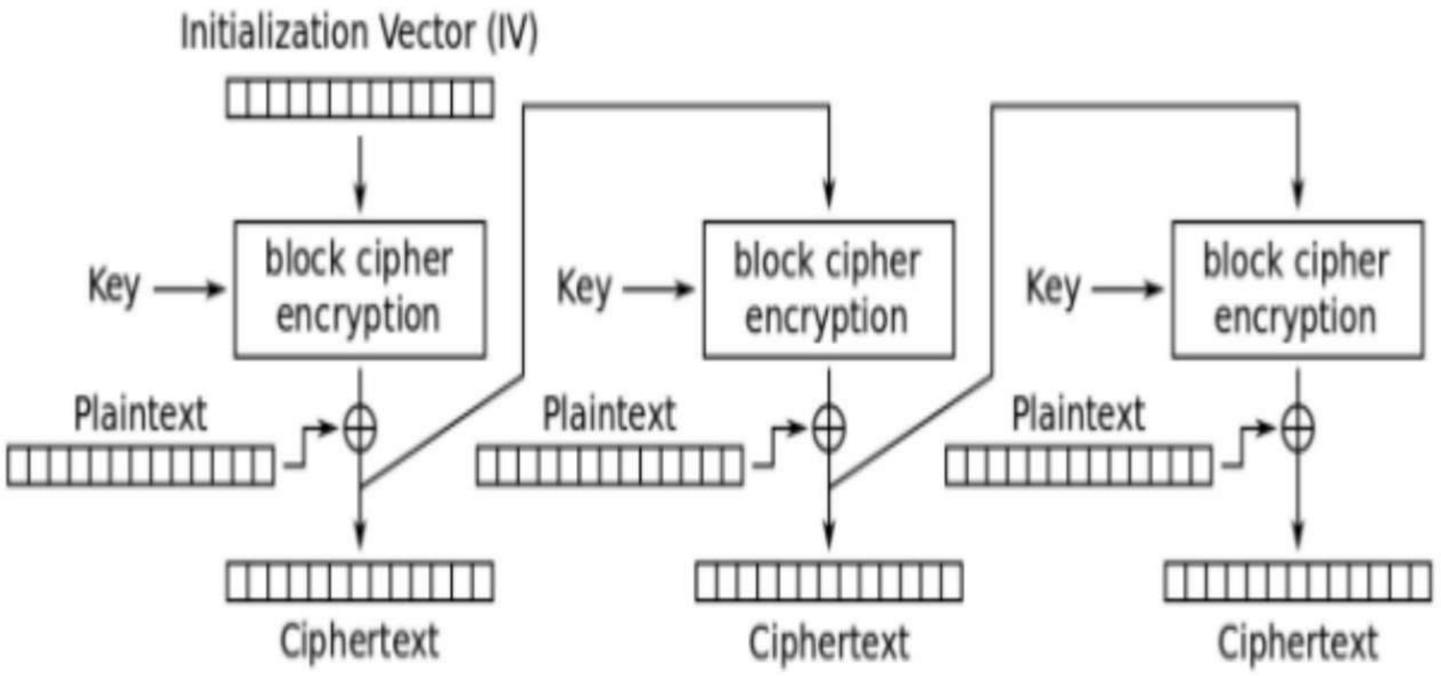
Sumber : Akbar & Pangestu (2018)

Pada mode CBC, setiap blok *ciphertext* C_i dixorkan dengan blok *plaintext* selanjutnya P_{i+1} sebelum dienkripsi dengan kunci K. Tetapi, terlebih dahulu dimulai dengan *initialization vector* yang dinotasikan sebagai IV dan didefinisikan sebagai $C_0 = IV$. Sehingga untuk mengenkripsi pesan dengan mode CBC dapat dilakukan sebagai berikut

$$C_i = (P_i \oplus C_{i-1}) \oplus K, \quad i \geq 1 \tag{2.22}$$

2.6.3 *Cipher Feedback* (CFB)

Mode CFB menggunakan sebuah vektor random sebagai input awalan dan menjadikan blok *plaintext* sebagai masukan operasi XOR diakhir. Hasil operasi XOR ini yang digunakan sebagai input pada operasi blok selanjutnya.



Gambar 2.14 Skema enkripsi CFB

Sumber : Akbar & Pangestu (2018)

Sama halnya dengan mode CBC, CFB menggunakan *initialization vector* yang berarti $C_0 = IV$.

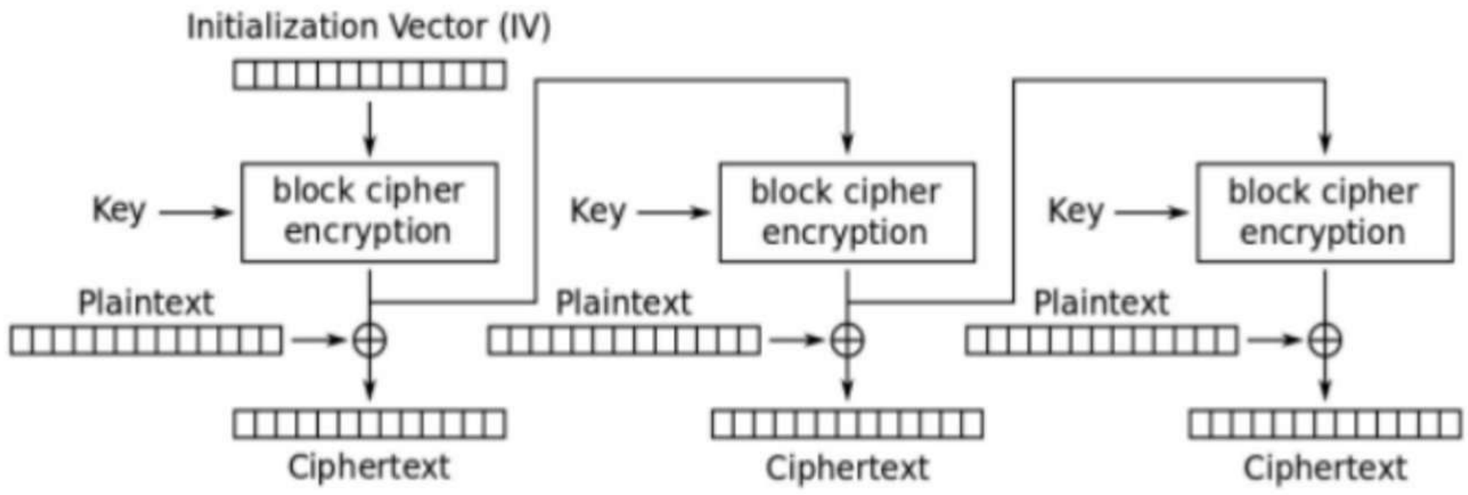
Sehingga proses enkripsi pada mode CFB dapat dilakukan dengan cara

$$C_i = P_i \oplus (E_k(C_i)),\ i \geq 1 \tag{2.23}$$

$(E_k(C_i))$ dapat diperoleh melalui operasi xor antara IV dan K.

2.6.4 *Output Feedback* (OFB)

Pada mode OFB, blok hasil algoritma *block cipher* yang akan diproses menggunakan operasi XOR pada blok *plaintext*. Mode OFB merupakan mode yang dapat diparalelkan karena operasi XOR dilakukan secara independen dan diakhir proses.



Gambar 2.15 Skema enkripsi OFB

Sumber : Akbar & Pangestu (2018)

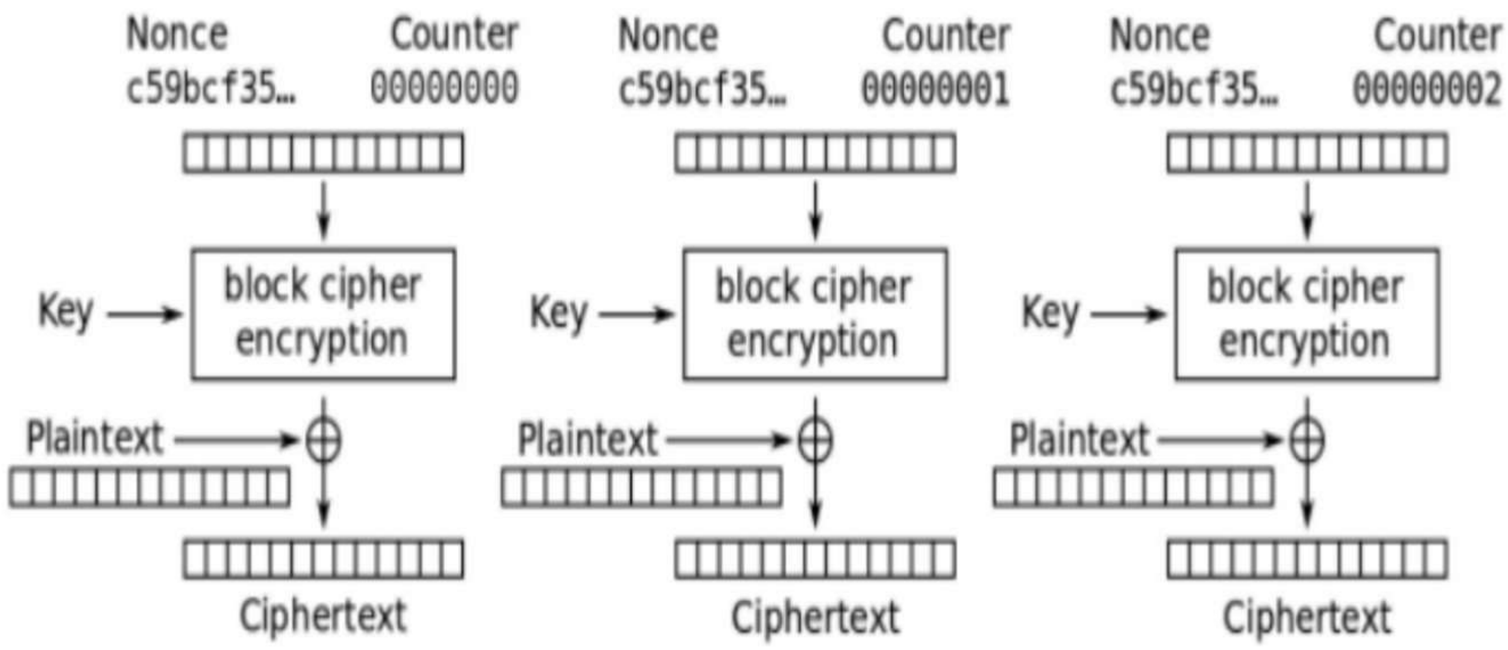
Yang membedakan mode OFB dan CFB pada proses enkripsi yaitu terletak pada pembangkitan kunci $E_k(C_i),\ i > 1$.

Sama halnya dengan mode CFB, *plaintext* x_1x_2 dapat dienkripsi dengan cara

$$C_i = P_i \oplus (E_k(C_i)),\ i \geq 1 \tag{2.24}$$

2.6.5 *Counter* (CTR)

Tiap proses blok akan menerima sebuah vektor masukan yang merupakan nilai selanjutnya dari vektor masukan proses sebelumnya. Vektor ini diperoleh dari *lossless operation* dari sebuah vektor acak terhadap bilangan hasil fungsi berurut.



Gambar 2.16 Skema enkripsi CTR

Sumber : Akbar & Pangestu (2018)

Perbedaan antara mode OFB dan CTR terletak pada cara membangkitkan *keystream*. Pada CTR ini, panjang dari blok *plaintext* dinotasikan sebagai m dan ctr dinotasikan sebagai rangkaian bit dari panjang m. Urutan dari rangkaian bit dari panjang m dinotasikan sebagai T_1T_2, didefinisikan sebagai

$$T_i = ctr + i - 1 \; mod \; 2^m, \; i \geq 1 \tag{2.25}$$

Maka, enkripsi *plaintext* x_1x_2 ....diperoleh dengan cara

$$y_i = x_i \oplus e_K(T_i), \; i \geq 1 \tag{2.26}$$

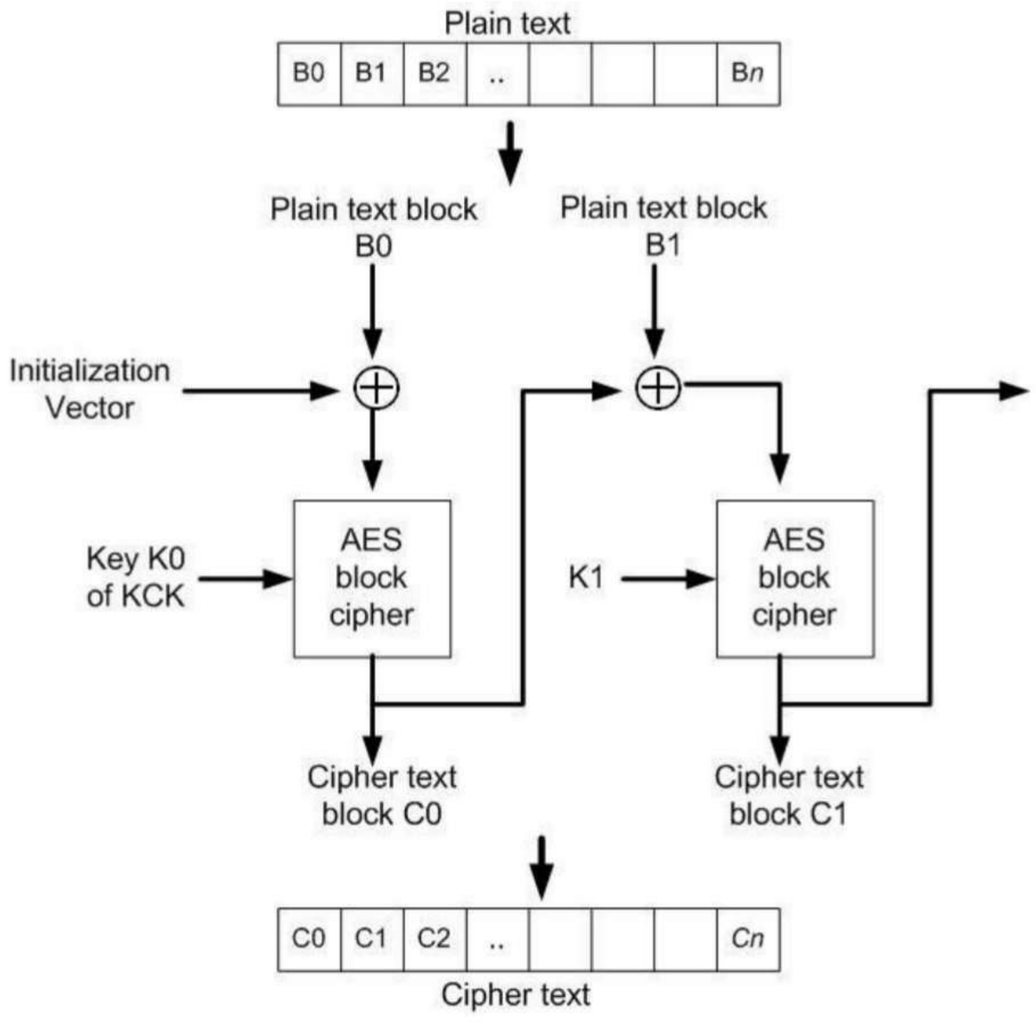
2.6.6 *Counter with Cipher block chaining MAC* (CCM)

Pada dasarnya, mode CCM merupakan gabungan dari mode CTR dan CBC. Tetapi, dalam proses enkripsi mode CCM hampir sama dengan mode CTR. Misalkan K merupakan kunci enkripsi dan $x_i = x_1 \ldots\ldots x_n$ sebagai *plaintext*. Maka *counters* $T_0, T_1, T_2, \ldots T_n$ didefinisikan sebagai

$$T_i = ctr + i \; mod \; 2^m, \; 0 \leq i \leq m \tag{2.27}$$

Maka, enkripsi *plaintext* x_1x_2 ....diperoleh dengan cara

$$y_i = x_i \oplus e_K(T_i), \; i \geq 1 \tag{2.28}$$

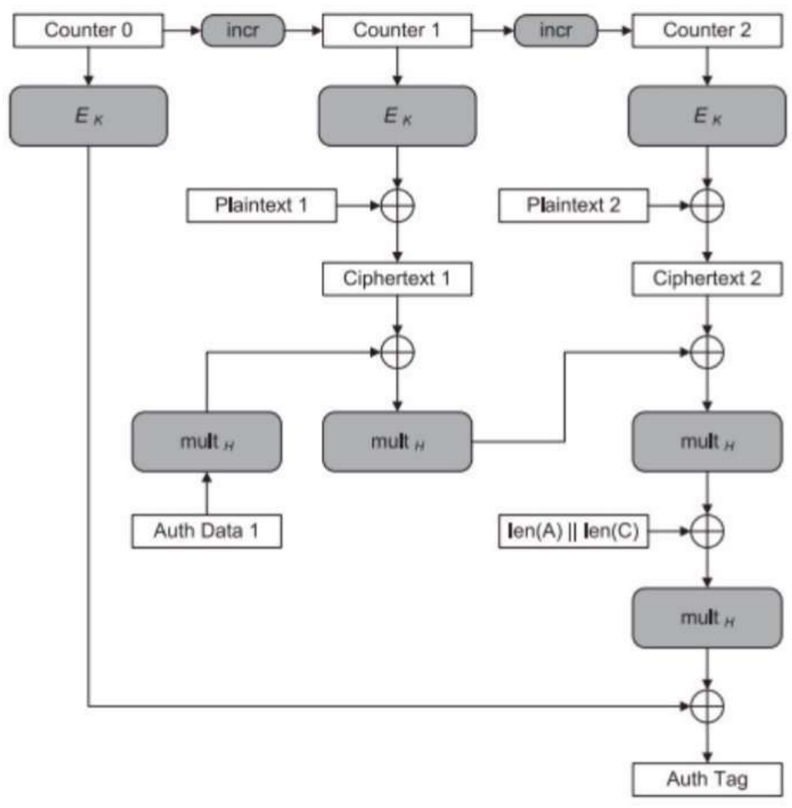


Gambar 17 : Skema CCM

Sumber : Japaneseclass

2.6.7 *Galois Counter* (GCM)

Enkripsi dilakukan dalam mode CTR menggunakan kunci AES 128-bit. Nilai awal 128-bit (*counter* 0) berasal dari *IV* yang biasa panjangnya 96 bit. *IV* ditransmisikan bersama dengan *ciphertext*, dan harus diubah setiap kali enkripsi baru dilakukan.



Gambar 18 : Skema GCM

Sumber : Douglas & Maura

Perhitungan "Auth Tag" memerlukan perkalian nilai konstan H pada $GF(2^8)$. Sedangkan nilai H ditentukan oleh proses enkripsi *counter* 0. "Auth Data 1" merupakan data terautentikasi yang tidak terenkripsi (dapat dikirimkan dalam bentuk tidak terenkripsi), namun dimasukkan ke dalam "Auth Tag". Data yang diautentikasi akan dikalikan secara berturut-turut dengan H kemudian hasil tersebut akan dixorkan dengan blok *ciphertext* yang dihasilkan. Operasi tersebut dilakukan hingga semua blok *ciphertext* diproses. Kemudian panjang dari *ciphertext* akan dikalikan dengan panjang data (*plaintext*) kemudian dixorkan dengan *counter* 0.

2.7 Sistem Bilangan

Sistem bilangan merupakan simbol-simbol khusus yang digunakan untuk membentuk sebuah bilangan baru. Secara umum, bilangan desimal merupakan sistem bilangan yang selalu digunakan dalam kehidupan sehari-hari. Sistem bilangan desimal menggunakan simbol 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9 yang menyatakan basis 10 atau jumlah digit yang terdapat pada satu sistem bilangan.

Sistem bilangan memiliki sifat *place-value* atau nilai tempat yang merupakan bobot tersendiri sesuai dengan tempat angka tersebut berada. Pada perangkat digital, sistem bilangan biner, oktal, dan heksadesimal juga biasa digunakan tetapi untuk tinjauan kali ini hanya berpusat pada pembahasan sistem bilangan biner dan heksadesimal.

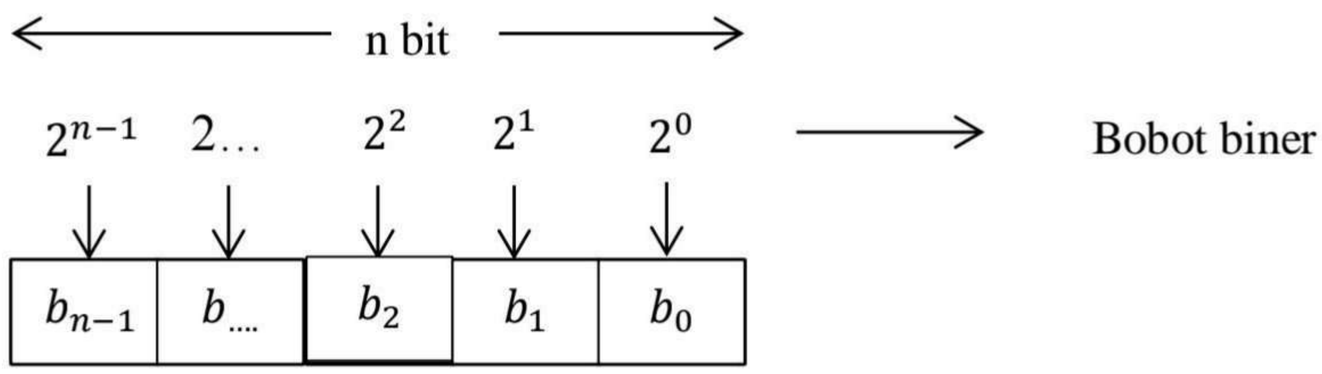
2.7.1 Sistem Bilangan Biner

Pada sistem bilangan biner, 0 dan 1 merupakan lambang yang paling sering digunakan untuk merepresentasikan sistem komputer. Digit bilangan biner disebut dengan *binary digit* atau *bit*. *Nibble* merupakan bit yang berjumlah empat sedangkan untuk delapan bit dinamakan *byte*. Komputer dapat memproses sejumlah bit yang mewakili suatu karakter berupa huruf, angka, atau lambang khusus yang dinamakan *word*. Satu *word* terdiri dari 4 sampai 64 bit.

Sistem bilangan biner merupakan sistem bilangan basis dua, yang berarti hanya dikenal dengan dua lambang, yaitu :

B = {0, 1}

Representasi bilangan bulat biner n-bit adalah sebagai berikut



Suatu bilangan biner Z dengan panjang n bit mempunyai nilai

$$Z = \sum_{i=0}^{n-1} b_i . 2^1 \qquad (2.29)$$

Pada suatu bilangan biner, format bit dikelompokkan menjadi empat-empat atau delapan-delapan. Sebagai contoh, 01101010 dapat ditulis 0110 1010. Empat bit sebelah kiri disebut dengan *nibble* tinggi dan empat bit sebelah kanan disebut *nibble* rendah. Kelompok paling kiri dapat saja mempunyai panjang bit lebih kecil. Sebagai contoh, bilangan 0101100 dapat ditulis 010 1100. *Nibble* tinggi adalah 010 dan *nibble* rendah adalah 1100.

Contoh

Konversikan 839 ke biner!

Jawab :

Tabel 2.6 Contoh konversi desimal ke biner

Nilai	Pembagi	Hasil Bagi	Sisa
839	2	419	1
419	2	209	1
209	2	104	1
104	2	52	0
52	2	26	0
26	2	13	0
13	2	6	1

6	2	3	0
3	2	1	1
1	2	0	0
839			$(0101000111)_2$

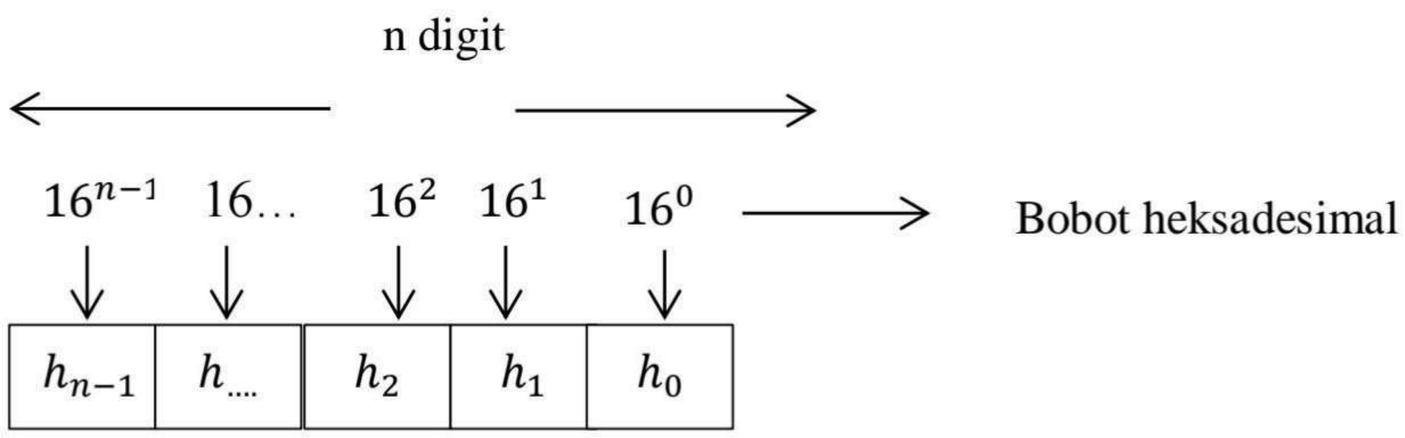
2.7.2 Sistem Bilangan Heksadesimal

Sistem bilangan heksadesimal merupakan sistem bilangan basis enam belas. Penerapan format heksadesimal banyak digunakan pada penyajian lokasi memori, penyajian isi memori, kode instruksi dan kode yang merepresentasikan alfanumerik dan karakter *non numeric*.

Pada sistem bilangan ini terdapat enam belas lambang, yaitu :

H = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Representasi bilangan bulat heksadesimal n bit adalah sebagai berikut



Sehingga suatu bilangan heksadesimal Z dengan panjang n digit mempunyai nilai

$$Z = \sum_{i=0}^{n-1} h_i . 16^1 \qquad (2.30)$$

Dalam mengkonversi biner ke heksadesimal, empat bit biner diubah menjadi satu digit heksadesimal. Sehingga untuk dapat mengkonversi bilangan biner ke heksadesimal atau sebaliknya, diperlukan tabel ASCII.

2.8 ASCII

Dalam sistem komputer, setiap informasi dapat dimengerti oleh sistem digital apabila informasi tersebut dibentuk dari serangkaian huruf, angka,

tanda baca, dan karakter kontrol. *American Standard Code for Information Interchange* (ASCII) menjadi salah satu standar internasional untuk mengkonversi berbagai macam informasi ke format biner.

Jumlah kode ASCII adalah 255 kode. Kode ASCII 0…127 dinyatakan sebagai kode untuk memanipulasi teks; sedangkan kode ASCII 128…255 merupakan kode untuk memanipulasi grafik. Kode pada ASCII dikelompokkan menjadi beberapa bagian :

- Kode yang tidak terlihat simbolnya seperti kode 10 (*line feed*), 13(*carriage return*), 8(tab), dan 32(*space*).
- Kode yang terlihat simbolnya seperti abjad (A…Z), numerik (0…9), karakter khusus (~!#$%^&*()_+?:”{}).
- Kode yang tidak ada di keyboard namun dapat ditampilkan yang umumnya digunakan untuk grafik.

ASCII memiliki pengkodean komposisi bilangan biner sebanyak 8 bit, dimulai dari $(0000\,0000)_2$ hingga $(1111\,1111)_2$ dengan total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam bilangan desimal.

Pada kode ASCII, angka dan huruf ditempatkan sedemikian rupa sehingga dapat dilakukan proses pengurutan secara sederhana. Dalam perhitungannya, kode ini bukan untuk perhitungan secara matematis. Sebagai contoh, jika untuk perhitungan digunakan format data biner 4 bit yaitu

Karakter		Kode ASCII		Nilai dalam biner
5	⇨	0000 0101	⇨	0101
4	⇨	0000 0100	⇨	0100
				⊕
1	⇦	0000 0001	⇦	0001

Contoh :

Dengan menggunakan tabel ASCII, ubahlah kata “TUGASKRIPTOGRAFI” menjadi bilangan heksadesimal lalu konversi ke bilangan biner!

Jawab :

T	S	P	R
U	K	T	A
G	R	O	F
A	I	G	I

Diubah ke dalam bentuk hex diperoleh

54	53	50	52
55	4B	54	41
47	52	4F	46
41	49	47	49

Diubah ke dalam bentuk biner diperoleh

54 = 0101 0100 53 = 0101 0011 50 = 0101 0000 52 = 0101 0010

55 = 0101 0101 4B = 0100 1011 54 = 0101 0100 41 = 0100 0001

47 = 0100 0111 52 = 0101 0010 4F = 0100 1111 46 = 0100 0110

41 = 0100 0001 49 = 0100 1001 47 = 0100 0111 49 = 0100 1001

2.9 Padding

Padding merupakan salah satu konsep dasar dalam kriptografi. Padding bekerja dengan cara menambahkan data pada suatu pesan baik pada bagian awal, tengah, atau akhir sebelum pesan tersebut dienkripsi.

Proses enkripsi *plaintext* menjadi *ciphertext* dalam blok-cipher harus disusun dalam blok-blok data dengan ukuran yang sama. AES menggunakan blok berukuran 128 bit. Karena data yang akan dimasukkan harus memiliki ukuran blok yang sama, maka dibutuhkan padding byte agar data pas dengan ukuran blok. Padding dilakukan dengan mengisi byte bernilai N bila dibutuhkan padding sebanyak N byte. sebagai contoh, bila dibutuhkan padding 2 byte, maka padding berisi 02 02, bila dibutuhkan padding 4 byte, maka padding berisi 04 04 04 04.

1A	4B	32	AC	21	AA	02	02								
EF	75	7A	AE	04	04	04	04								
3F	A4	BF	05	05	05	05	05								
10	BB	FF	4E	5C	C5	2E	03	08	08	08	08	08	08	08	08

Gambar 2.19 Contoh Padding

Dalam standar PKCS telah diatur bahwa padding tetap ditambahkan pada semua data, walaupun data tersebut sudah genap seukuran blok yang diperlukan. PKCS (*Public-Key Cryptography Standards*) dirancang dan diterbitkan pada tahun 1990-an oleh RSA Security Inc. PKCS#7 merupakan metode padding standar yang dapat digunakan sebelum mengenkripsi pesan untuk menentukan jumlah byte padding yang kemudian ditentukan sebagai nilai. Misalnya untuk ukuran blok 128-bit, maka pada saat dilakukan pengujian, terdapat 7 byte (untuk pengkodean ASCII) yang mewakili data, dan terdapat 9 (0x09) nilai padding.

Banyak metode enkripsi lama menggunakan ukuran blok 64-bit yang berarti metode tersebut akan membaca dalam 8 karakter (8-bit) ke dalam satu blok dan kemudian mengenkripsinya. Metode umum yang menggunakan ukuran blok ini adalah DES dan 3-DES. Pada umumnya,metode kriptografi simetris, sepeti AES, menggunakan ukuran blok 128-bit, yang berarti akan dibaca dalam 16 karakter sekaligus.