SHAMAQ: Algoritma *block cipher* berbasis SHA-3

Stefanus Ardi Mulia[1], Aliffiqri Agwar[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517119@std.stei.itb.ac.id, 13517107@std.stei.itb.ac.id

Abstrak. Enkripsi merupakan sesuatu yang kita sadari atau tidak banyak digunakan dalam komunikasi digital yang kita lakukan dalam keseharian kita. Salah satu keluarga algoritma enkripsi yang sering digunakan dalam perpindahan informasi digital tersebut adalah keluarga algoritma *block cipher*. Algoritma *block cipher* bekerja dengan melakukan enkripsi pada satu blok bit dengan panjang tertentu. Kami mengusulkan suatu algoritma *block cipher* baru bernama SHAMAQ, sebuah algoritma *block cipher* yang dapat menerima kunci dengan panjang variabel tanpa mengurangi kekuatannya.
Kata kunci: kriptografi, enkripsi, dekripsi, *block cipher*, jaringan feistel, *hash*, SHA-3.

1. Pendahuluan

Dalam era digital ini, kehidupan kita tidak bisa terlepas dari interaksi-interaksi melalui internet. Pada dasarnya interaksi-interaksi tersebut merupakan pertukaran informasi yang terjadi antar mesin maupun perangkat pada jaringan Internet. Ketika berinteraksi, kita melakukan pemberian maupun permintaan informasi. Informasi ini disimpan pada sebuah paket data. Data yang membawa informasi tersebut ditransmisikan dan merambat melalui medium tertentu seperti kabel tembaga, sinyal radio, dan lain-lain. Bukan tidak mungkin bagi pihak tidak bertanggung jawab untuk menyadap medium yang kita gunakan dan membaca data dengan leluasa.

Untuk menanggulangi permasalahan tersebut, diperlukan suatu cara untuk menyembunyikan data yang kita transmisikan dan cara ini adalah menggunakan kriptografi. Transmisi informasi yang umumnya dilakukan pada media digital sering dienkripsi menggunakan salah satu keluarga algoritma krptografi yang disebut *block cipher*. *Block cipher* bekerja pada sekumpulan bit yang disebut blok, kemudian melakukan transformasi pada blok tersebut.

Kami mengusulkan suatu algoritma *block cipher* baru bernama SHAMAQ, terinspirasi dari nama sihir yang digunakan pada serial televisi Re:Zero kara Hajimeru Isekai Seikatsu yang menyebabkan siapapun yang terpengaruh sihir tersebut menjadi tidak dapat merasakan lingkungan sekitarnya. SHAMAQ bekerja dengan menerima panjang kunci yang variabel kemudian melakukan transformasi menggunakan jaringan feistel dengan upa kunci yang dibangkitkan dengan algoritma *hash* aman SHA-3 256.

1.1. Penelitian terkait

Terdapat beberapa algoritma *block cipher* yang telah diteliti sebelumnya, beberapa dari algoritma tersebut antara lain adalah sebagai berikut.

1.1.1. AES

Advanced Encryption Standard (AES) merupakan algoritma *block cipher* yang dirilis oleh *National Institute of Standards and Technology* (NIST). AES merupakan variasi dari algoritma Rijndael dengan ukuran blok tetap 128 bit dan ukuran kunci 128, 192, atau 256 bit. Berbeda dengan DES yang didesain berdasarkan jaringan feistel, AES didesain berdasarkan jaringan substitusi-permutasi. Di dalam AES

operasi-operasi dilakukan pada blok yang disusun dalam matriks kolom-baris 4x4. Secara garis besar, algoritma AES berjalan seperti berikut:

1. Dilakukan ekspansi kunci yang menghasilkan upa kunci 128 bit untuk setiap putaran,
2. Blok masukan di-xor-kan dengan upa kunci,
3. Berdasarkan panjang kunci, 128, 192, atau 256 bit, dilakukan 9, 11, atau 13 putaran yang terdiri atas:
    a. Substitusi byte pada blok menggunakan s-box AES,
    b. Geser byte-byte pada blok,
    c. Melakukan pencampuran kolom blok dengan mentransformasikannya dengan suatu matriks tetap,
    d. Menambahkan upa kunci pada blok,
4. Pada putaran terakhir dilakukan operasi yang sedikit berbeda dibandingkan putaran sebelumnya, yaitu:
    a. Substitusi byte pada blok menggunakan s-box AES,
    b. Geser byte-byte pada blok,
    c. Menambahkan upa kunci pada blok.

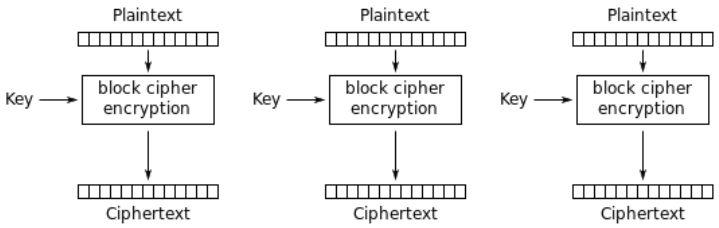
2. Dasar Teori

2.1. Block Cipher

Block Cipher merupakan salah satu keluarga algoritma kunci simetris dalam kriptografi yang banyak digunakan dalam kriptografi modern. *Block Cipher* bekerja pada blok-blok berukuran tetap, misal 128 bit, dari suatu masukan. Block cipher memiliki beberapa mode operasi dalam penggunaannya, beberapa di antaranya adalah:

2.1.1. Electronic Codebook (EBC)

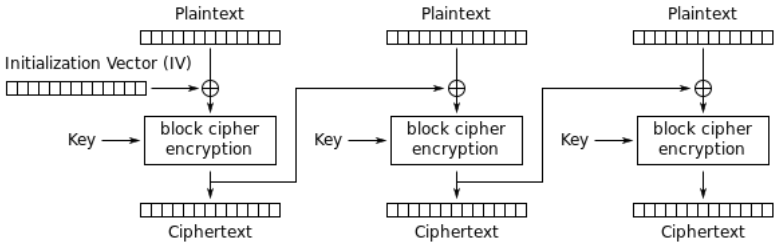
Teknik paling sederhana pada enkripsi adalah penggunaan *electronic codebook* yang diberi nama dari *codebook* (buku kode) yang benar-benar dipakai pada enkripsi klasik. ECB membagi pesan masukannya menjadi blok-blok dengan panjang yang tetap lalu melakukan operasi secara terpisah pada masing-masing blok.



Gambar 1. Enkripsi mode ECB.

2.1.2. Cipher Block Chaining (CBC)

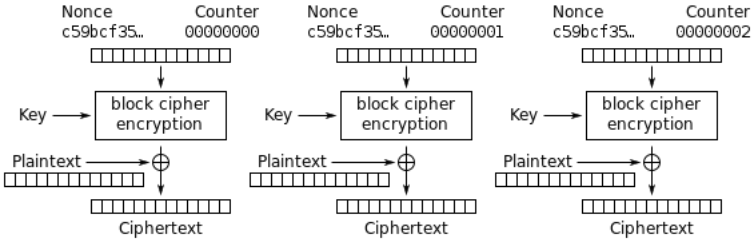
Mode CBC bekerja dengan cara melakukan *chaining* terhadap enkripsi tiap blok pesan. Pada enkripsi tiap blok pesan, plaintext yang diterima di-xor-kan dengan ciphertext dari blok sebelumnya. Khusus untuk blok pertama, di bangkitkan suatu blok tambahan IV (*Initialization Vector*) acak yang berukuran sebesar satu blok untuk di-xor-kan dengan blok pertama mode ini merupakan salah satu implementasi yang menerapkan prinsip *diffusion* Shannon.



Gambar 2. Enkripsi mode CBC.

2.1.3. Counter (CTR)

Berbeda dengan mode lain yang algoritmanya mengubah langsung pesan menjadi pesan terenkripsi, mode CTR melakukan enkripsi pada suatu nilai cacah tertentu (counter) sehingga menghasilkan suatu kunci alir yang kemudian dibagi-bagi menjadi blok-blok dan di-xor-kan dengan pesan.



Gambar 3. Enkripsi mode CTR.

2.2. Confusion and Diffusion

Dalam mendesain suatu algoritma *block cipher*, terdapat sejumlah konsep yang perlu diperhatikan untuk meningkatkan kekuatannya. Salah satu dari konsep tersebut adalah konsep *confusion* dan *diffusion* yang diperkenalkan oleh Shanon [1].

Confusion merupakan suatu sifat statistik dari suatu fungi transformasi yang mengukur keterhubungan suatu nilai masukan terhadap keluarannya. Pada dasarnya, konsep *confusion* memastikan bahwa dengan hanya melihat suatu nilai keluaran kita hanya mendapatkan sedikit atau bahkan tidak ada informasi mengenai nilai masukannya.

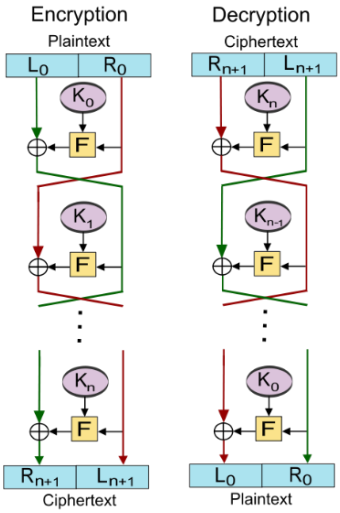
Diffusion merupakan sifat dari suatu fungsi transformasi yang menyebabkan setiap simbol yang digunakan sebagai masukannya mempengaruhi berbagai simbol lain pada keluarannya. Hal ini dimaksudkan untuk meningkatkan kesulitan analisis pola pada keluaran suatu algoritma enkripsi untuk menemukan masukannya.

2.3. Jaringan Feistel

Jaringan feistel merupakan suatu struktur yang sering digunakan dalam pembangunan *block cipher* [2] yang diberi nama sesai penemunya yaitu Horst Feistel, seorang kriptografer IBM berkebangsaan Jerman. Jaringan feistel memiliki keuntungan dalam *block cipher* karena sifatnya yang simetris, yaitu proses dalam enkripsi dan dekripsi yang sangat mirip bahkan untuk beberapa kasus sama persis, hanya membutuhkan pembalikan urutan kunci yang digunakan setiap putaran. Secara garis besar, cara kerja jaringan feistel klasik adalah:

1. Membagi blok masukan menjadi dua bagian kiri (L_0) dan kanan (R_0).
2. Bagian R dimasukkan ke dalam suatu fungsi f beserta kunci putaran tersebut.
3. Hasil dari langkah 2 ($f_0(R_0, K_0)$) di-xor-kan dengan bagian L.
4. Hasil dari langkah 3 ($L_0 \oplus f_0(R_0, K_0)$) diteruskan sebagai bagian kanan putaran selanjutnya (R_1) dan bagian kanan putaran ini (R_0) menjadi bagian kiri putaran selanjutnya (L_1).

5. Langkah 2 hingga 4 dilakukan terus menerus sebanyak putaran yang diinginkan.
6. Untuk putaran terakhir, langkah 4 dimodifikasi, bagian kanan (R_{n-1}) tetap dijadikan bagian kanan (R_n) dan hasil langkah 3 ($L_{n-1} \oplus f_{n-1}(R_{n-1}, K_{n-1})$) menjadi bagian kiri (L_n).



Gambar 4. Struktur Jaringan Feistel.

2.4. Kotak Substitusi (S-box)

Kotak substitusi atau *substitution box* (*s-box*) merupakan suatu tabel yang digunakan pada operasi substitusi dalam berbagai algoritma kriptografi yang menggunakan kunci simetris. Kotak substitusi digunakan sebagai "pengecoh" dalam suatu algoritma kriptografi sebab dengan penggunaannya hubungan antara pesan dengan hasil enkripsi semakin sulit dicari, menerapkan prinsip *confusion* Shannon.

AES S-box

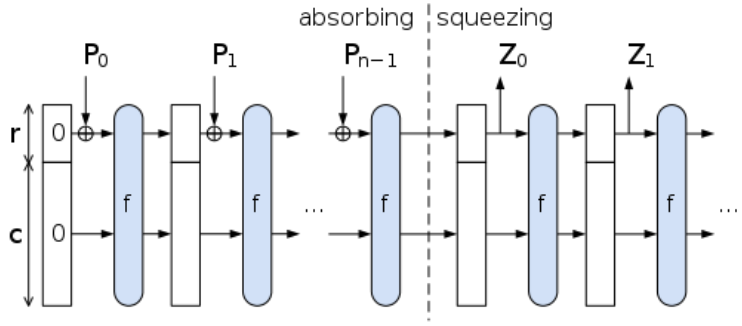
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84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

Gambar 5. S-box yang digunakan oleh AES.

2.5. Algoritma SHA-3

SHA-3 merupakan salah satu algoritma yang termasuk ke dalam keluarga algoritma *Secure Hash Algorithm* (SHA) yang dirilis oleh NIST. SHA-3 sendiri sebenarnya merupakan upa set dari keluarga kriptografi primitif Keccak yang didesain oleh Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.

Algoritma SHA-3 menggunakan konstruksi spons yang “menyerap” data dan hasilnya “diperas” keluar. Pada fase menyerap, blok-blok masukan diekspansi kemudian di-xor dengan upa set *state* yang tersimpan. Hasil tersebut kemudian ditransformasi dengan sebuah fungsi *f*. Pada fase peras, blok-blok untuk keluaran dibaca dari upa set dari fase sebelumnya diselingi dengan fungsi *f*.



Gambar 6. Konstruksi spons yang digunakan oleh SHA-3.

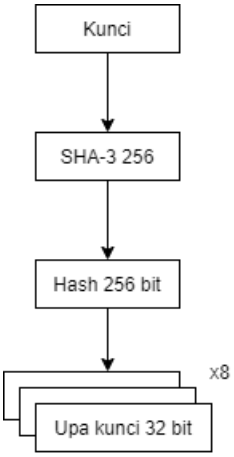
3. Block cipher yang diusulkan

Algoritma *block cipher* SHAMAQ merupakan sebuah algoritma *block cipher* yang berbasis pada *cipher* jaringan feistel. Algoritma ini menerima kunci dengan panjang bebas. Namun agar kekuatannya tidak terpengaruh oleh panjang kunci yang diberikan, dilakukan ekspansi kunci. Kunci utama dimasukkan ke dalam fungsi hash SHA3-256 sehingga membentuk sebuah kunci turunan dengan panjang 256 bit. Kunci yang telah diekspansi ini kemudian dibagi-bagi menjadi beberapa upa kunci yang digunakan pada setiap putaran jaringan feistel.

Enkripsi dengan jaringan feistel dilakukan dalam 8 putaran. Untuk setiap putaran dilakukan enkripsi terhadap blok pesan yang berukuran 64 bit yang kemudian dibagi menjadi dua bagian yang diproses dengan beberapa fungsi f dari jaringan feistel yang dipakai. Fungsi f pertama pada jaringan feistel tersebut adalah operasi xor antara satu bagian pesan dengan upa kunci pertama. Fungsi f kedua melakukan substitusi pada satu bagian pesanan menggunakan kotak substitusi yang juga digunakan oleh algoritma enkripsi AES.

3.1. Pembangkitan upa kunci

Algoritma *block cipher* SHAMAQ melakukan 8 putaran enkripsi menggunakan jaringan feistel. Untuk setiap putaran tersebut digunakan upa kunci berbeda-beda yang diturunkan dari kunci utama yang dimasukkan oleh pengguna. Upa kunci dibangkitkan dari kunci utama tersebut dengan cara memasukkannya ke dalam fungsi *hash* SHA-3 256 yang akan menghasilkan *hash* dengan panjang 256 bit. *Hash* tersebut kemudian dibagi-bagi menjadi 8 bagian yang digunakan pada setiap putaran jaringan feistel.



Gambar 7. Pembangkitan upa kunci SHAMAQ.

3.2. Jaringan feistel SHAMAQ

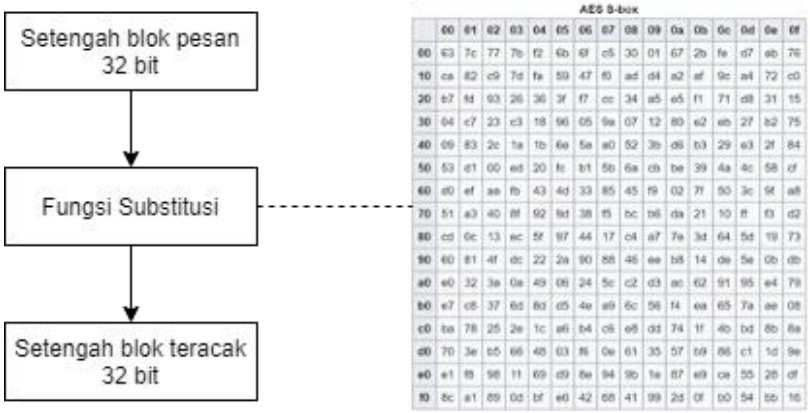
Pada inti dari algoritma *block cipher* SHAMAQ adalah sebuah implementasi jaringan feistel. Di dalam jaringan ini blok pesan yang berukuran 64 bit dibagi menjadi dua bagian yang masing-masing berukuran 32 bit. Dua bagian blok pesan tersebut kemudian dimasukkan pada sebuah fungsi peubah atau fungsi f yang terdiri dari operasi xor, substitusi, serta permutasi. Enkripsi menggunakan jaringan feistel ini dilakukan sebanyak 8 putaran dengan 8 kunci berbeda yang telah dibangkitkan pada langkah sebelumnya.

3.2.1. Xor

Pada operasi ini, dilakukan operasi xor terhadap bagian pesan yang dimasukkan ke dalam operasi ini dengan upa kunci putaran yang telah dibangkitkan sebelumnya.

3.2.2. Substitusi

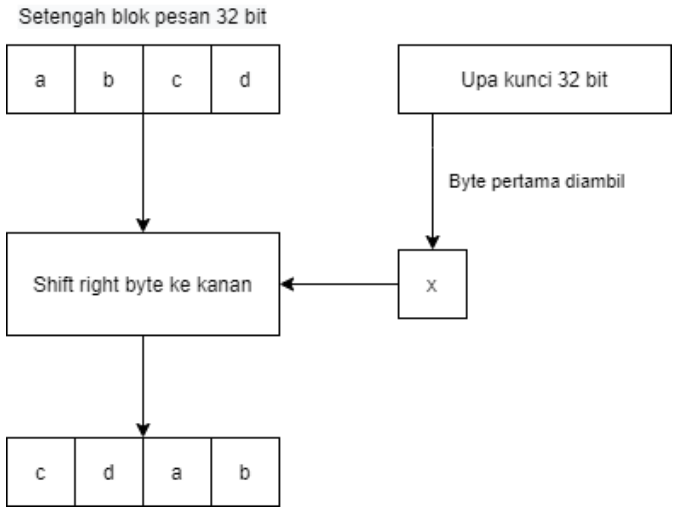
Pada operasi ini dilakukan substitusi terhadap hasil dari bagian sebelumnya menggunakan kotak s yang juga digunakan oleh algoritma enkripsi AES. Substitusi ini dilakukan untuk setiap byte atau 8 bit, menggunakan 4 bit signifikannya sebagai baris dan 4 bit kurang signifikannya sebagai kolom. Proses ini menghasilkan sebuah byte baru (8 bit) yang sangat berbeda dengan data yang masuk.



Gambar 8. Operasi substitusi SHAMAQ.

3.2.3. Permutasi

Pada operasi ini dilakukan pengubahan urutan dari byte pada setengah blok pesan yang masuk. Cara operasi ini melakukan pengubahan adalah dengan cara “memutar” byte yang ada ke arah kanan. Operasi ini dilakukan dengan pergeseran sebesar nilai byte pertama upa kunci pada putaran dimodulo dengan total byte pada setengah blok pesan yang di proses. Hal ini akan berakibat perputaran yang tidak menentu berdasarkan upa kunci yang dibangkitkan.



Gambar 9. Operasi permutasi SHAMAQ.

4. Pengujian dan Analisis

4.1. Pengujian

Pengujian dilakukan terhadap tiga mode enkripsi yang diimplementasikan yaitu mode EBC, CBC, dan Counter Mode. Untuk setiap mode juga dilakukan pengujian dan pembandingan jika dilakukan pengubahan 1 bit pesan maupun kunci yang digunakan. Ketiganya diuji dengan panjang bit masing-masing blok adalah 64 bit. Pengujian ini menampilkan waktu eksekusi preproses (pembacaan file dan penyimpanannya pada block), proses (eksekusi enkripsi dan dekripsi) dan posproses (penulisan kembali pada suatu file). Pengujian ini juga menghasilkan sebuah file baru berisi hasil enkripsi atau dekripsi. Pengujian dilakukan dengan variabel-variabel sebagai berikut:

Tabel 4.1. Kunci dan plainteks pengujian

Kunci sebelum SHA
alicebobkripto
Plainteks
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "u de Finibus Bonorum et Malorum" by Cicero are also reproduced

in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

4.1.1. Mode ECB

Mode ECB melakukan enkripsi/dekripsi pada masing-masing blok tanpa ketergantungan antar blok. Sehingga setiap blok langsung dimasukkan ke fungsi enkripsi/dekripsi. Hasil pengujian menggunakan ECB menghasilkan keluaran sebagai berikut:

Tabel 4.2. Hasil pengujian dengan mode ECB

Ciphertext hasil enkripsi	
⊳ှ틋⊡ᅴ網织뻭ᴪ塱陕릅蚝봒稚무칢뤓罙흌灗ᛱ鑿统烟✚□㷊□褈∫⑲ꭗ킐茂뼥行멵ـٓ驢撚ذ̀ძघ 滙糘蚷弫七ɞ◌熙□呸둓만읒꼛鬤틧탡ǂ痪玃耞騨□蕐〓겏茕iv□瀰⁼귾豋⁖〜ɣʕ勺뎏╖騤□ɫ驕疏 鯤摵뗻◌䊾徵⊡渾胕탱怅중□猪炄□□ꩡↄ퀮　丶堀㊨麑纐□₤涼⇞捻⊡ 뙲犒□□텔囲何틍♨鏲□ 厍呗漉偶작□陳퀜泠쇡ↀ□暄쐟□뒩《⺘ᘛ⊼盿쬔무쾖뤓罙腏꽌弊-.Ż□讴덴↛攘ṅ·皾딴瀘뒐罴 尅□쐽鈐ႏ꾁멦□㊉떻昝삯沛□嬞繱뚄만읒꼛欥篏ᶠ攏ຽ><樺□□螢齇뺫□ᦏ검ᶍ□ᔕ뢣喇ʙ瞼闒□ 隠賬涓樣竽□□舡□躐邠蜼瓒닏娗↿嫪烝□ᷝ◌ㅓ擊鲑迪넴뭙태◌□뭇/ㄷ嚶□⸢ ᠊ᠵ□皀귅칤□쟀킁廖 ⋜倚旟ę抗□ᅇ□箜灟粪□篙佢倸Ļ□’곰셉鏝诙Ǝ€礂鉀✎蓍縋堉□꾐□慌□覼즠縓栌科桐腕玓뗬 駕瑪縭鱠□⩧銬附□旟钯꽃僳帀袂□覑贠砿캳⑥唝뾍垄□꽷懸桒挞ﷺ□샦□壤槷奓⇷睠熅˚뵉逖 摯勢鐮瀝踵螟蜋鉁隺ↄ浮乭桔占輙涍榎艋-.ׄ杨□□ᗱ擨ۍⁿ앚謽□　□ᡧ□晋⧟좪勦关奇□煊ⱶ喇ʙ 瞼%□□쑛饅避邛□⍖聡□龏̣□싼遣ོ拐□読□瞀늖穧繰耽朌鼊⒣멩隆ـٰ麥갹뢳릚柑ၟ崩珽ɫ낙墺 □잤ؚ椿슝꽈鲑迪넴뭙태◌□뭇/ㄷ嚶□⸢ ᠵ蹇姑ȡ咘刂蕜ᆗ□□僛熛븱뢣獵□쬀□蝅桲鐮□□⊲꾒ⴛ□	
Waktu eksekusi enkripsi (detik)	**Waktu eksekusi dekripsi (detik)**
Pre-process time : 0.0009992122650146484 Process time : 0.011989831924438477 Post-process time : 3.4070076942443848 Total time : 3.419996738433838	Pre-process time : 0.0 Process time : 0.011991739273071289 Post-process time : 21.302626132965088 Total time : 21.31461787223816

Pengujian juga dilakukan dengan mengubah 1 bit dari plainteks sebelum enkripsi, kunci dan cipherteks sebelum dekripsi. Hasil dari pengujian ini adalah sebagai berikut:

Tabel 4.3. Hasil pengujian perubahan 1 bit

Jenis Perubahan	Letak Perubahan	Perbedaan Byte	Persentase Kesamaan
Kunci	alicebnbkripto	1067/1072	0,47%
Plainteks	Byte 771	8/1072	99,25%
Cipherteks	Byte 768	16/1072	98,50%

4.1.2. Mode CBC

Mode CBC melakukan enkripsi/dekripsi pada masing-masing blok tanpa ketergantungan antar blok. Sehingga setiap blok langsung dimasukkan ke fungsi enkripsi/dekripsi. Hasil pengujian menggunakan CBC menghasilkan keluaran sebagai berikut:

Tabel 4.4. Hasil pengujian dengan mode CBC

Ciphertext hasil enkripsi

<table>
<tr><td colspan="2">蛲ƘƤ僧□□□□▦□ᄄᄌAJ얫쵉볟鎩□ɣ秥ὤ*昱Ƀ嵱□□扢ᆰᆼ℘멎熯□鋟□□□□味̊̈̌ᄔ微荸Y粀梘喨
썽Ɥ礫憕㎖□ᅡᅱ嶷桕瓊쇅慉숦処̊뾂賍豕̊ᔑ醹⋞□쾽□뗑℠ヺッシェル梺纗Ꝺ帨□俺頸ᄋ襟驛쥤栙瘶顣孛
梈貝墣ᵿ韙轀□燚□□⑳藝飀ᔕᔕ朒쾳□↔ᵻ̊扈闟쏙□旂炃莏弓浐뎲ᵾ쾂ꜛ楸怛읷餜罡□柤犉□∯
涧⒣才簭歕䙡↕□蹩□鷉謏締骰熢美産□稅쥐茉瘜뎼ݓ浮□辕레翺᳗藠J軺躙苩□□榴ϳ□□쾲뚇
잔섦藝睫磏⟶纩□□镸뚱□ᶐ̒릉ⓕ⏏̣□□□胚裟蹰嫔鯐掙窜㉖藾竝ᆺ넑ダーズ멎□뾂□熖訹ᐤ□ᔓ̊
□긕긎쾿餅堉痖□Ａ□粡□깑ᆿ涚硧득鱆体誦锠신淸□囍□,嫪蘺룱婕瀹季酶囌声苗徼걸嬬≑쬚□
ᄅᇹP□d읬匧ᴥ覜돩□膟苁ᄆ腽□욢뒷‖덛峥架麮쬼□뢥ψҩℱ纵嫊□쬑뚈J碒뤼〈擡倮□鴅□둳尤弻
芊Ïᴍ孖뗘□□伬뷉벡□□鼹匜篘糀㕁쬗梾絾쏲劉毬鲜□뷇荑讃⊲□濟咬□Γ綮璵韃鐕炯≬뤕覈痃麟
╁咼包槝佹稹蝙ᆞ醙腹쬚ᄾᅷᅡᅲ人□棐ℳ阁읪嗞郥갦ホ□窟□葱뤽禺갌 龖爈魩ᔕ뻴쾼姟騐鏂□垝ᘐ
轂妁□↳□◉□□□睹筶點□ₒ하≪舱嚏朓옐□□抖ξ뤂ᒍ膐ᘐ쬤娭袿ṅᏌ舡뒷가꾏睩뤌嫓挩쥡看棆櫘窗</td></tr>
<tr><td>Waktu eksekusi enkripsi (detik)</td><td>Waktu eksekusi dekripsi (detik)</td></tr>
<tr><td>Pre-process time : 0.0
Process time : 0.010991334915161133
Post-process time : 5.421861171722412
Total time : 5.432852506637573</td><td>Tidak dilakukan</td></tr>
</table>

4.1.3. Mode Counter

Tabel 4.5. Hasil pengujian dengan mode Counter Mode

<table>
<tr><td colspan="2">Ciphertext hasil enkripsi</td></tr>
<tr><td colspan="2">¿¥rEG¶‹¥nx²í¥¹ò\¼+tMî”ÛCj]ÕsÆƒ’5ýÿn‹¿áÃ›Ê%;ÉEA!BÛ,ÌÁ†OAô5“ö·KÒà“‘´Ï¨»§ýÔM"ò†Ê¯¥
°l<ˆŠÎòà36&÷çä(n~SZGGÔZªG| ü†¶gNø!óPÕ®¯ÜÓ›(iöÀÇ–R°£ò_lÌÕW ü¶h—
)ªKžëÉ<,è|wþÕÁÆî©Ý¾È6UÐµî“WÞÜYï)¿æ©"œnâUç’ž—
kºË¦ÄI‹”Ý¶©É™ò#AGy&dP-¹Nûåzïüfæ¶òQåëºXž¦a›Ûß²¦êß;£š`Œ_"Sw:_ˆko7}áfyƒjÕÂ™^Ÿkˆ×’Ð<br>õ³¼²¼…7²ªÉ‹ûOF×ªßúµÿ MÙÂî–ìõPËßT»¢eR×æ33Ä%«ÿ;Qðê© Jc#mû¥szöÝU<br>D<br>r¤B©Cÿ”ÍA®&`jU«¶ÅÛ‹Ts©…‹dQdÁïÕ-Äén»¦´Ýbœ¨n›¦åôµãõ{pÍÛXqwsWÎm¬šæ(ƒÜ01ìNuÅQÍi
á,MQÐ[îÚ4|f…»˜¸Vm¡áeò-?ÂT¨ûmpšËõ»ÐÓX¸æþB?ªn!Øv£d
ëë±5Rså…õmYkÏeµEñÎ-ç¨Dš3é}ŸöPÞædYÏ” ¨7ÛBÅ~ €c¡Öc§ÃÊlqƒ¯ãƒ\—·|VÌmã•ÿêÎ¦w„–
Ï˜¶4Ìr<%Jþ7
œ4/‹Q¹
£T»eQÛ[P9B³«ñ[Èµ™a–úÂÁ-e tŸeN§ Ë7áQ]××aÊùÔºüt™jÑ¡
/rò±J¢9Õ,3âd€5æêéALõÊ-ÃH(Ùø¥cêÓù=åÎI›eSÈÝn¬–¦#üjMQÎ,Édd»ž6ôZ,&ËÂf¦:QuÏã\à’2–
Ì-üá_JòÓ°ã2Ëal¼©Œ%øùÐ7MÐPÀB/ëíæš²;TUðÔ¹i!,iS÷(oá(ÉEÛ{tï¦ œ‘a?µ¿m»ÏÛÉEö¢³
LÎæª®it$wNE‰4……X/+í{÷9ž§™=Ü–¹2™dønAáúËî‘sì&"%
ÏÎ …g‰D¡òƒÑ\ït
Íý_´È…F¿Â’êù³®¬ÓOÝ6 ·xÜâ_ëÈpD×;&÷=m¹-Y<ŸRws/Y2øè7ÇV
!M&Tå¾Gq_à,±HQNŠ¯uò3^¹÷î'Ø1ª·þE»Ìö”‘â€ÖÁ¸Uˆzƒ›½|¸¿#mâ™§rI¼¿‘r*ŒÏ³°,½-
ð°4ƒ4þdAR<eîpae¤&Ð-!Ñ©>ô</td></tr>
<tr><td>Waktu eksekusi enkripsi (detik)</td><td>Waktu eksekusi dekripsi (detik)</td></tr>
<tr><td>Pre-process time : 0.0
Process time : 0.012990236282348633
Post-process time : 3.94853138923645
Total time : 3.961521625518799</td><td>Tidak dilakukan</td></tr>
</table>

5. Referensi

[1] C. E. Shannon, “Communication Theory of Secrecy Systems,” *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[2] K. Nyberg, “Generalized feistel networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1996, vol. 1163, pp. 91–104, doi: 10.1007/bfb0034838.

Pernyataan

Puji syukur kami panjatkan ke hadirat Tuhan yang Maha Esa atas berkatnya penulis dapat menyelesaikan makalah SHAMAQ: Algoritma block cipher berbasis SHA-3 ini. Terima kasih juga ditujukan penulis kepada Dr. Ir. Rinaldi Munir, MT., selaku dosen pengampu mata kuliah IF4020 Kriptografi yang telah memberikan pengetahuan dasar mengenai kriptografi modern kepada kami. Selain itu penulis juga berterima kasih kepada semua pihak yang turut membantu selesainya makalah ini secara langsung maupun tidak langsung.