

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Analisis

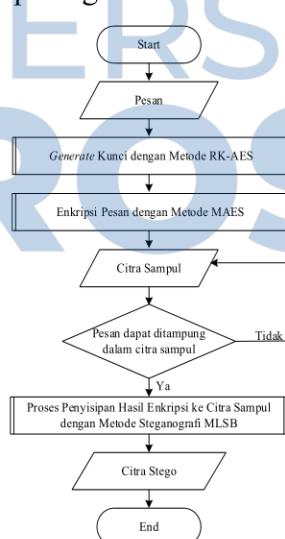
Sebelum memulai proses perancangan perangkat lunak, maka terlebih dahulu perlu dilakukan proses analisis terhadap sistem yang akan dirancang. Analisis sistem adalah teknik pemecahan masalah dari sebuah sistem sebagai prasyarat desain sistem.

##### 3.1.1 Analisis Proses

Dalam proses kerja penggabungan dari algoritma kriptografi RK-MAES dan metode steganografi MLSB ini, terdapat dua buah proses yaitu proses penempelan pesan rahasia yang akan dilakukan oleh pengirim dan ekstraksi pesan rahasia yang akan dilakukan oleh penerima.

###### 3.1.1.1 Analisis Proses Penyisipan

Prosedur kerja dari proses penyisipan dengan menggunakan penggabungan dari algoritma kriptografi RK-MAES dan metode steganografi MLSB ini dapat dideskripsikan seperti terlihat pada gambar berikut:



Gambar 3.1 Proses Enkripsi dan Penyisipan Metode Kriptografi RK-MAES dan Metode Steganografi MLSB

Langkah kerja dari proses penyisipan dengan menggunakan penggabungan algoritma kriptografi RK-MAES dan metode steganografi MLSB adalah sebagai berikut:

1. *Input* Pesan

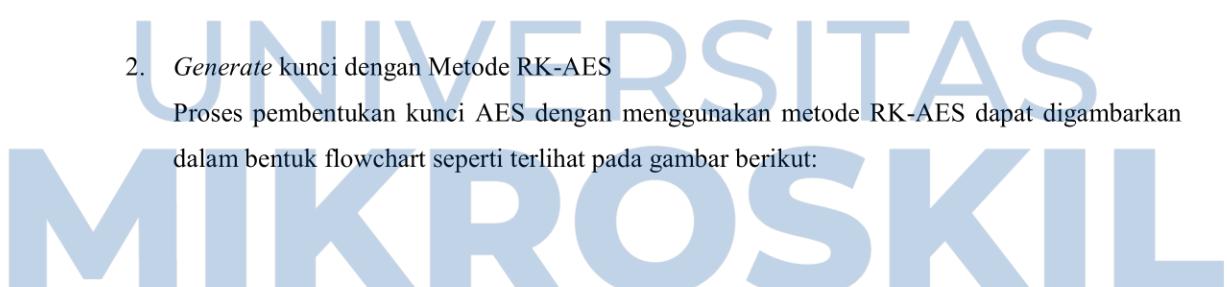
Pesan = "Password = abc12"

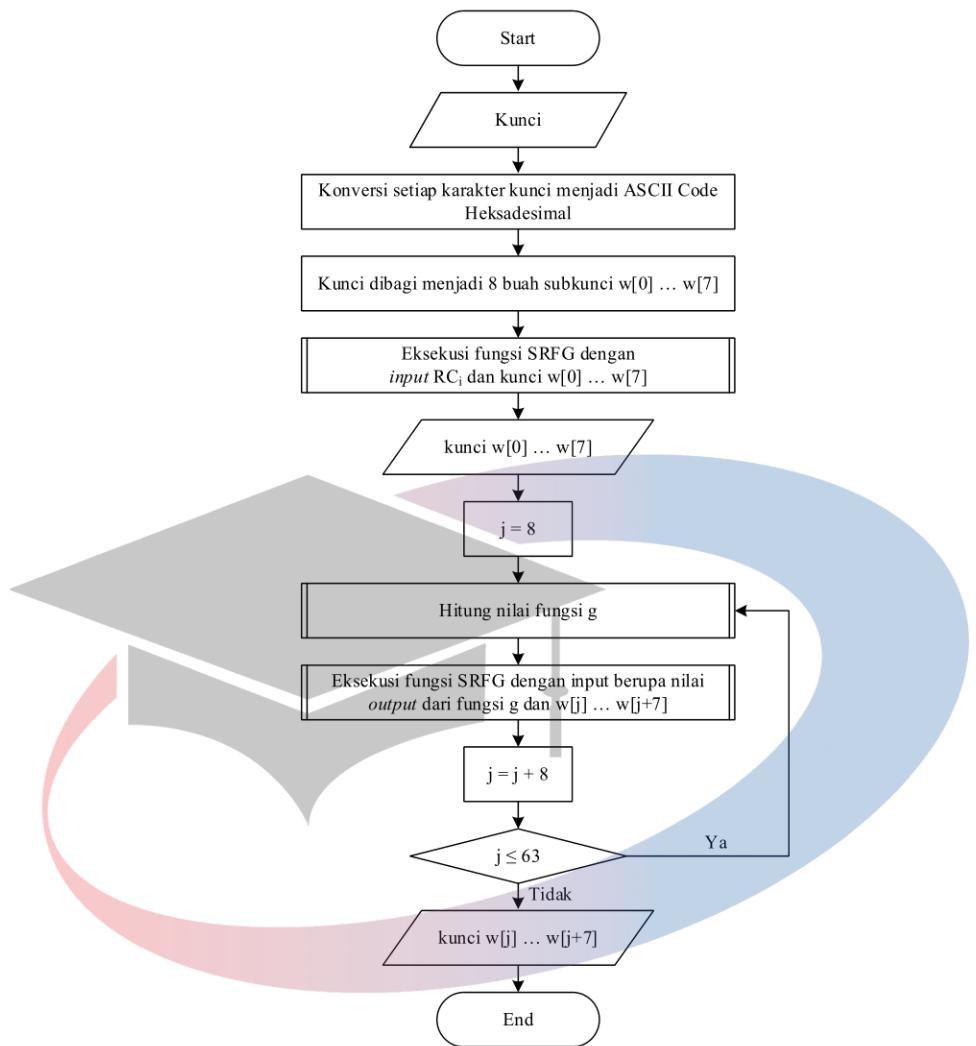
Konversi setiap karakter pesan ke ASCII Code Biner:

P	:	80	:	<b>01010000</b>
a	:	97	:	<b>01100001</b>
s	:	115	:	<b>01110011</b>
s	:	115	:	<b>01110011</b>
w	:	119	:	<b>01110111</b>
o	:	111	:	<b>01101111</b>
r	:	114	:	<b>01110010</b>
d	:	100	:	<b>01100100</b>
:	:	32	:	<b>00100000</b>
=	:	62	:	<b>00111101</b>
:	:	32	:	<b>00100000</b>
a	:	97	:	<b>01100001</b>
b	:	98	:	<b>01100010</b>
c	:	99	:	<b>01100011</b>
1	:	49	:	<b>00110001</b>
2	:	50	:	<b>00110010</b>

2. *Generate* kunci dengan Metode RK-AES

Proses pembentukan kunci AES dengan menggunakan metode RK-AES dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut:





Gambar 3.2 Flowchart Proses Pembentukan Kunci AES dengan Menggunakan Metode RK-AES

Langkah kerja dari proses pembentukan kunci AES dengan menggunakan metode RK-AES adalah sebagai berikut:

a. *Input* kunci

Kunci = KRIPTOGRAFIMETODEAESRIJNDAEL256.

b. Konversikan setiap karakter kunci ke bentuk ASCII Code Heksadesimal



Kunci setelah diubah ke notasi heksadesimal (disimbolkan dengan 'X')

$$\begin{array}{l} X \\ = \\ 4B524950544F47524146494D45544F444541455352494A4E4441454C3235362E \end{array}$$

- c. Kunci dibagi menjadi 8 buah sub kunci  $w[0] \dots w[7]$ .

$$w[0] = 4B524950$$

$$w[1] = 544F4752$$

$$w[2] = 4146494D$$

$$w[3] = 45544F44$$

$$w[4] = 45414553$$

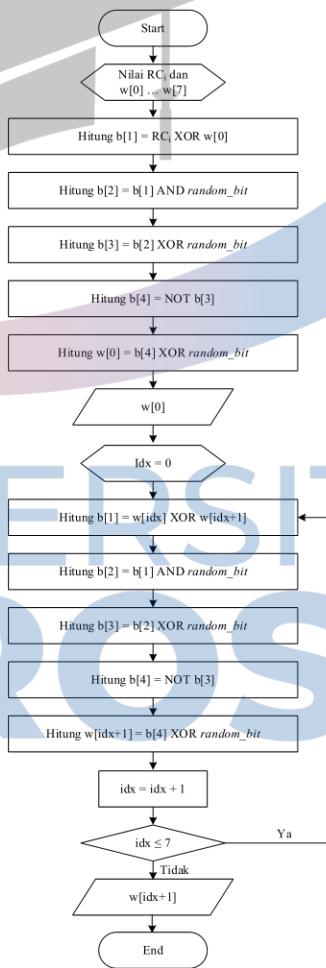
$$w[5] = 52494A4E$$

$$w[6] = 4441454C$$

$$w[7] = 3235362E$$

- d. Eksekusi fungsi SRFG dengan *input*  $RC_i$  dan  $w[0] \dots w[7]$ .

Proses kerja dari fungsi SRFG yang digunakan dapat digambarkan sebagai berikut:



Gambar 3.3 Flowchart Proses Eksekusi Fungsi SRFG dengan Penambahan  $RC_i$

Langkah kerja dari fungsi SRFG dengan penambahan  $RC_i$  adalah:

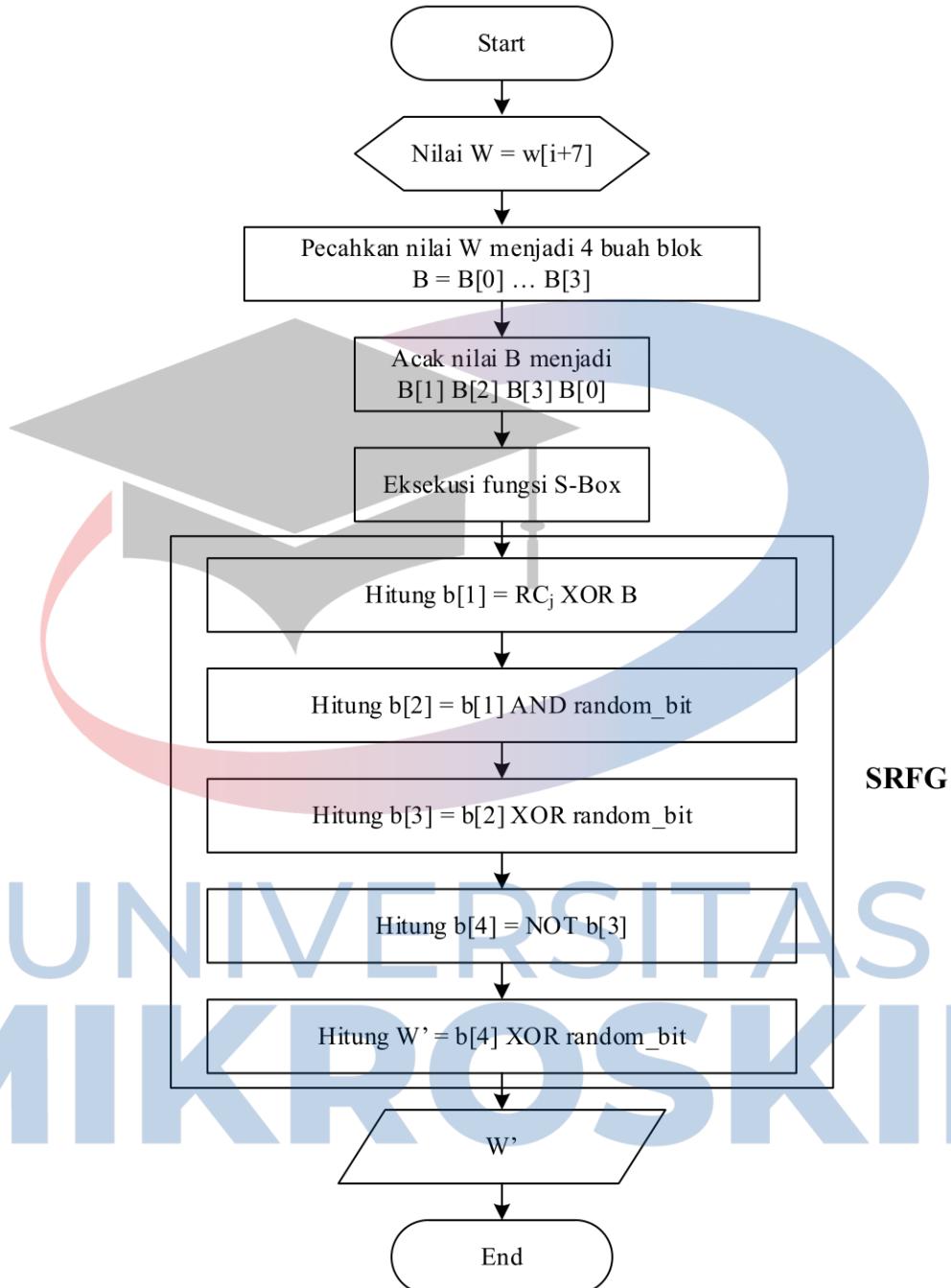
- 1) Input RC1 = 01 00 00 00  
 $w[0] = 4B524950$   
 $w[1] = 544F4752$   
 $w[2] = 4146494D$   
 $w[3] = 45544F44$   
 $w[4] = 45414553$   
 $w[5] = 52494A4E$   
 $w[6] = 4441454C$   
 $w[7] = 3235362E$
- 2) Hitung  $b[1]$   
 $b[1] = RC1 \text{ XOR } w[0]$   
 $b[1] = 01\ 00\ 00\ 00 \text{ XOR } 4B\ 52$   
 $49\ 50$   
 $b[1] = 4A\ 52\ 49\ 50$
- 3) Hitung  $b[2]$   
Misakan  $random\_bit = 4E115836$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = 4A\ 52\ 49\ 50 \text{ AND } 4E\ 11$   
 $58\ 36$   
 $b[2] = 4A\ 10\ 48\ 10$
- 4) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 4A\ 10\ 48\ 10 \text{ XOR } 4E\ 11$   
 $58\ 36$   
 $b[3] = 04\ 01\ 10\ 26$
- 5) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = FB\ FE\ EF\ D9$
- 6) Hitung  $w[0]$   
 $w[0] = b[4] \text{ XOR } random\_bit$   
 $w[0] = FB\ FE\ EF\ D9 \text{ XOR } 4E$   
 $11\ 58\ 36$   
 $w[0] = B5\ EF\ B7\ EF$
- 7) Output  $w[0]$   
 $w[0] = B5\ EF\ B7\ EF$
- 8) Hitung  $b[1]$
- 9) Hitung  $b[2]$   
Misakan  $random\_bit = 3F226846$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = E1\ A0\ F0\ BD \text{ AND } 3F$   
 $22\ 68\ 46$
- 10) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 21\ 20\ 60\ 04 \text{ XOR } 3F\ 22$   
 $68\ 46$   
 $b[3] = 1E\ 02\ 08\ 42$
- 11) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = E1\ FD\ F7\ BD$
- 12) Hitung  $w[1]$   
 $w[1] = b[4] \text{ XOR } random\_bit$   
 $w[1] = E1\ FD\ F7\ BD \text{ XOR } 3F$   
 $22\ 68\ 46$   
 $w[1] = DE\ DF\ 9F\ FB$
- 13) Output  $w[1]$   
 $w[1] = DE\ DF\ 9F\ FB$
- 14) Hitung  $b[1]$   
 $b[1] = w[1] \text{ XOR } w[2]$   
 $b[1] = DE\ DF\ 9F\ FB \text{ XOR } 41$   
 $46\ 49\ 4D$   
 $b[1] = 9F\ 99\ D6\ B6$
- 15) Hitung  $b[2]$   
Misakan  $random\_bit = 4A337755$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = 9F\ 99\ D6\ B6 \text{ AND } 4A\ 33$   
 $77\ 55$   
 $b[2] = 0A\ 11\ 56\ 14$

- 16) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 0A\ 11\ 56\ 14 \text{ XOR } 4A\ 33$   
 $77\ 55$   
 $b[3] = 40\ 22\ 21\ 41$
- 17) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = BF\ DD\ DE\ BE$
- 18) Hitung  $w[2]$   
 $w[2] = b[4] \text{ XOR } random\_bit$   
 $w[2] = BF\ DD\ DE\ BE \text{ XOR } 4A$   
 $33\ 77\ 55$   
 $w[2] = F5\ EE\ A9\ EB$
- 19) Output  $w[2]$   
 $w[2] = F5\ EE\ A9\ EB$
- 20) Hitung  $b[1]$   
 $b[1] = w[2] \text{ XOR } w[3]$   
 $b[1] = F5\ EE\ A9\ EB \text{ XOR } 45$   
 $54\ 4F\ 44$   
 $b[1] = B0\ BA\ E6\ AF$
- 21) Hitung  $b[2]$   
Misakan  $random\_bit = A421226F$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = B0\ BA\ E6\ AF \text{ AND } A4$   
 $21\ 22\ 6F$   
 $b[2] = A0\ 20\ 22\ 2F$
- 22) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = A0\ 20\ 22\ 2F \text{ XOR } A4\ 21$   
 $22\ 6F$   
 $b[3] = 04\ 01\ 00\ 40$
- 23) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = FB\ FE\ FF\ BF$
- 24) Hitung  $w[3]$   
 $w[3] = b[4] \text{ XOR } random\_bit$
- 25) Output  $w[3]$   
 $w[3] = 5F\ DF\ DD\ D0$
- 26) Hitung  $b[1]$   
 $b[1] = w[3] \text{ XOR } w[4]$   
 $b[1] = 5F\ DF\ DD\ D0 \text{ XOR } 45$   
 $41\ 45\ 53$   
 $b[1] = 1A\ 9E\ 98\ 83$
- 27) Hitung  $b[2]$   
Misakan  $random\_bit = B342113E$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = 1A\ 9E\ 98\ 83 \text{ AND } B3\ 42$   
 $11\ 3E$   
 $b[2] = 12\ 02\ 10\ 02$
- 28) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 12\ 02\ 10\ 02 \text{ XOR } B3\ 42$   
 $11\ 3E$   
 $b[3] = A1\ 40\ 01\ 3C$
- 29) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 5E\ BF\ FE\ C3$
- 30) Hitung  $w[4]$   
 $w[4] = b[4] \text{ XOR } random\_bit$   
 $w[4] = 5E\ BF\ FE\ C3 \text{ XOR } B3$   
 $42\ 11\ 3E$   
 $w[4] = ED\ FD\ EF\ FD$
- 31) Output  $w[4]$   
 $w[4] = ED\ FD\ EF\ FD$
- 32) Hitung  $b[1]$   
 $b[1] = w[4] \text{ XOR } w[5]$   
 $b[1] = ED\ FD\ EF\ FD \text{ XOR } 52$   
 $49\ 4A\ 4E$   
 $b[1] = BF\ B4\ A5\ B3$

- 33) Hitung b[2]  
 Misakan  $random\_bit =$   
 $C4663F2D$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = BF\ B4\ A5\ B3\ \text{ AND } C4\ 66\ 66\ 3F\ 2D$   
 $b[2] = 84\ 24\ 25\ 21$
- 34) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 84\ 24\ 25\ 21 \text{ XOR } C4\ 66\ 3F\ 2D$   
 $b[3] = 40\ 42\ 1A\ 0C$
- 35) Hitung b[4]  
 $b[4] = NOT b[3]$   
 $b[4] = BF\ BD\ E5\ F3$
- 36) Hitung w[5]  
 $w[5] = b[4] \text{ XOR } random\_bit$   
 $w[5] = BF\ BD\ E5\ F3\ XOR\ C4\ 66\ 3F\ 2D$   
 $w[5] = 7B\ DB\ DA\ DE$
- 37) Output w[5]  
 $w[5] = 7B\ DB\ DA\ DE$
- 38) Hitung b[1]  
 $b[1] = w[5] \text{ XOR } w[6]$   
 $b[1] = 7B\ DB\ DA\ DE\ XOR\ 44\ 41\ 45\ 4C$   
 $b[1] = 3F\ 9A\ 9F\ 92$
- 39) Hitung b[2]  
 Misakan  $random\_bit =$   
 $E76969CC$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = 3F\ 9A\ 9F\ 92\ \text{ AND } E7\ 69\ 69\ CC$   
 $b[2] = 27\ 08\ 09\ 80$
- 40) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 27\ 08\ 09\ 80\ XOR\ E7\ 69\ 69\ CC$
- 41) Hitung b[4]  
 $b[4] = NOT b[3]$   
 $b[4] = 3F\ 9E\ 9F\ B3$
- 42) Hitung w[6]  
 $w[6] = b[4] \text{ XOR } random\_bit$   
 $w[6] = 3F\ 9E\ 9F\ B3\ XOR\ E7\ 69\ 69\ CC$   
 $w[6] = D8\ F7\ F6\ 7F$
- 43) Output w[6]  
 $w[6] = D8\ F7\ F6\ 7F$
- 44) Hitung b[1]  
 $b[1] = w[6] \text{ XOR } w[7]$   
 $b[1] = D8\ F7\ F6\ 7F\ XOR\ 32\ 35\ 36\ 2E$   
 $b[1] = EA\ C2\ C0\ 51$
- 45) Hitung b[2]  
 Misakan  $random\_bit =$   
 $F69677DD$   
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = EA\ C2\ C0\ 51\ \text{ AND } F6\ 96\ 77\ DD$   
 $b[2] = E2\ 82\ 40\ 51$
- 46) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = E2\ 82\ 40\ 51\ XOR\ F6\ 96\ 77\ DD$   
 $b[3] = 14\ 14\ 37\ 8C$
- 47) Hitung b[4]  
 $b[4] = NOT b[3]$   
 $b[4] = EB\ EB\ C8\ 73$
- 48) Hitung w[7]  
 $w[7] = b[4] \text{ XOR } random\_bit$   
 $w[7] = EB\ EB\ C8\ 73\ XOR\ F6\ 96\ 77\ DD$   
 $w[7] = 1D\ 7D\ BF\ AE$
- 49) Output w[7]  
 $w[7] = 1D\ 7D\ BF\ AE$

e. Hitung nilai fungsi  $g$ .

Langkah kerja dari fungsi  $g$  dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut:



Gambar 3.4 Flowchart Proses Kerja Fungsi  $g$

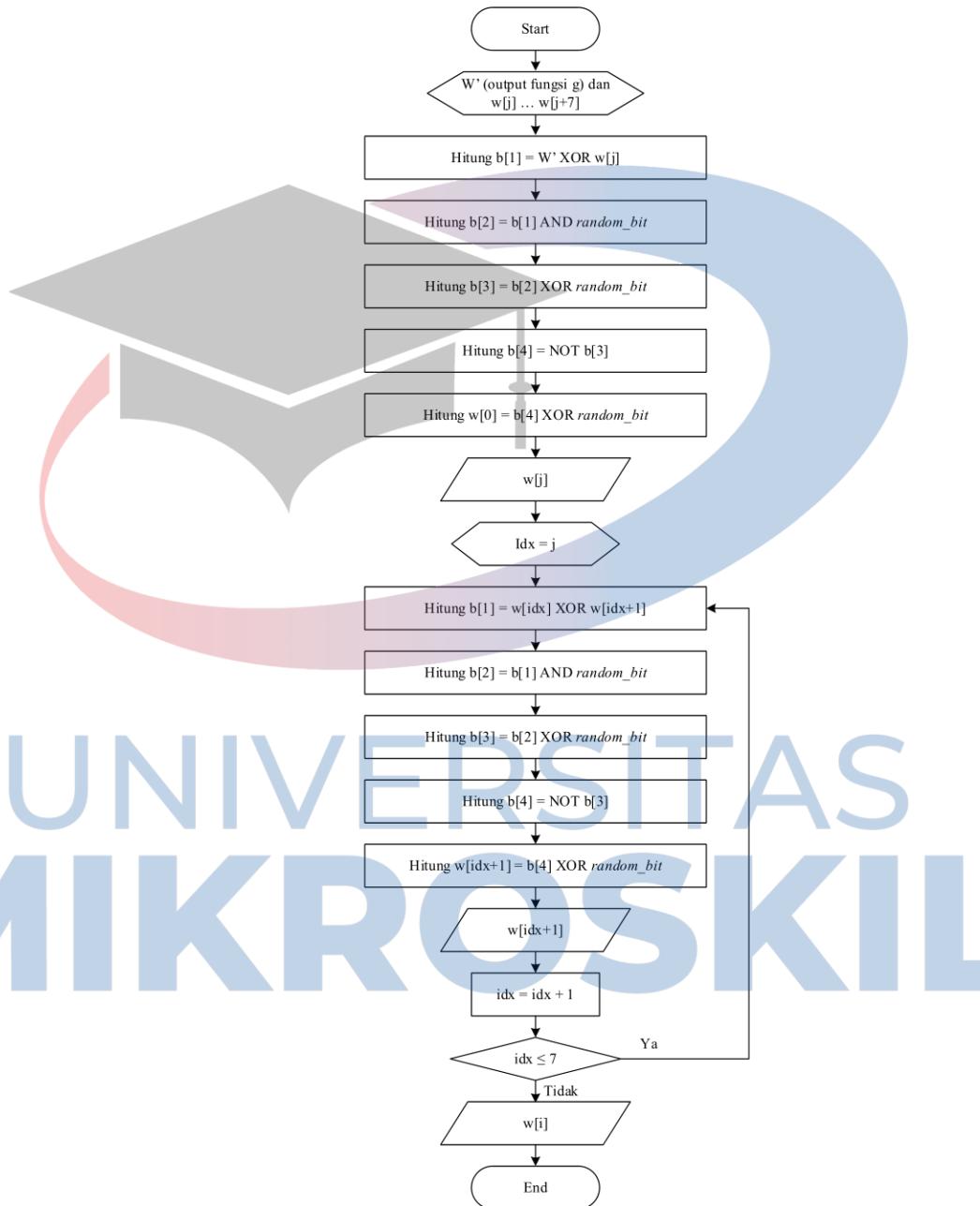
Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- 
- 1) Nilai W = w[7] = 1D 7D  
BF AE.
  - 2) Pecahkan W menjadi 4 buah blok B:  
 $B[0] = 1D$   
 $B[1] = 7D$   
 $B[2] = BF$   
 $B[3] = AE$
  - 3) Acak nilai B menjadi:  
 $B[0] = 7D$   
 $B[1] = BF$   
 $B[2] = AE$   
 $B[3] = 1D$
  - 4) Eksekusi fungsi S-Box.  
 $B'[0] = \text{SBox}(7D) = FF$   
 $B'[1] = \text{SBox}(BF) = 08$   
 $B'[2] = \text{SBox}(AE) = E4$   
 $B'[3] = \text{SBox}(1D) = A4$
  - 5) Hitung  $b[1]$   
 $b[1] = \text{RC1 XOR } B$   
 $b[1] = 01\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ \text{XOR FF}$   
 $08\ E4\ A4$   
 $b[1] = FE\ 08\ E4\ A4$
  - 6) Hitung  $b[2]$   
Misakan  $\text{random\_bit} = F69677DD$   
 $b[2] = b[1] \text{ AND }$   
 $\text{random\_bit}$   
 $b[2] = FE\ 08\ E4\ A4 \text{ AND }$   
 $F6\ 96\ 77\ DD$   
 $b[2] = F6\ 00\ 64\ 84$
  - 7) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $\text{random\_bit}$   
 $b[3] = F6\ 00\ 64\ 84 \text{ XOR }$   
 $F6\ 96\ 77\ DD$   
 $b[3] = 00\ 96\ 13\ 59$
  - 8) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = FF\ 69\ EC\ A6$
  - 9) Hitung  $W'$   
 $W' = b[4] \text{ XOR }$   
 $\text{random\_bit}$   
 $W' = FF\ 69\ EC\ A6 \text{ XOR }$   
 $F6\ 96\ 77\ DD$   
 $W' = 09\ FF\ 9B\ 7B$
  - 10) Output  $W'$   
 $W' = 09\ FF\ 9B\ 7B$

UNIVERSITAS  
MIKROSKIL

- f. Eksekusi fungsi SRFG dengan *input* berupa nilai *Output* dari fungsi  $g$  dan nilai  $w[j]$  ...  $w[j+7]$  sebelumnya.

Langkah kerja dari fungsi SRFG ini dengan *input* berupa nilai *Output* dari fungsi  $g$  dan nilai  $w[j]$  ...  $w[j+7]$  sebelumnya dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut:

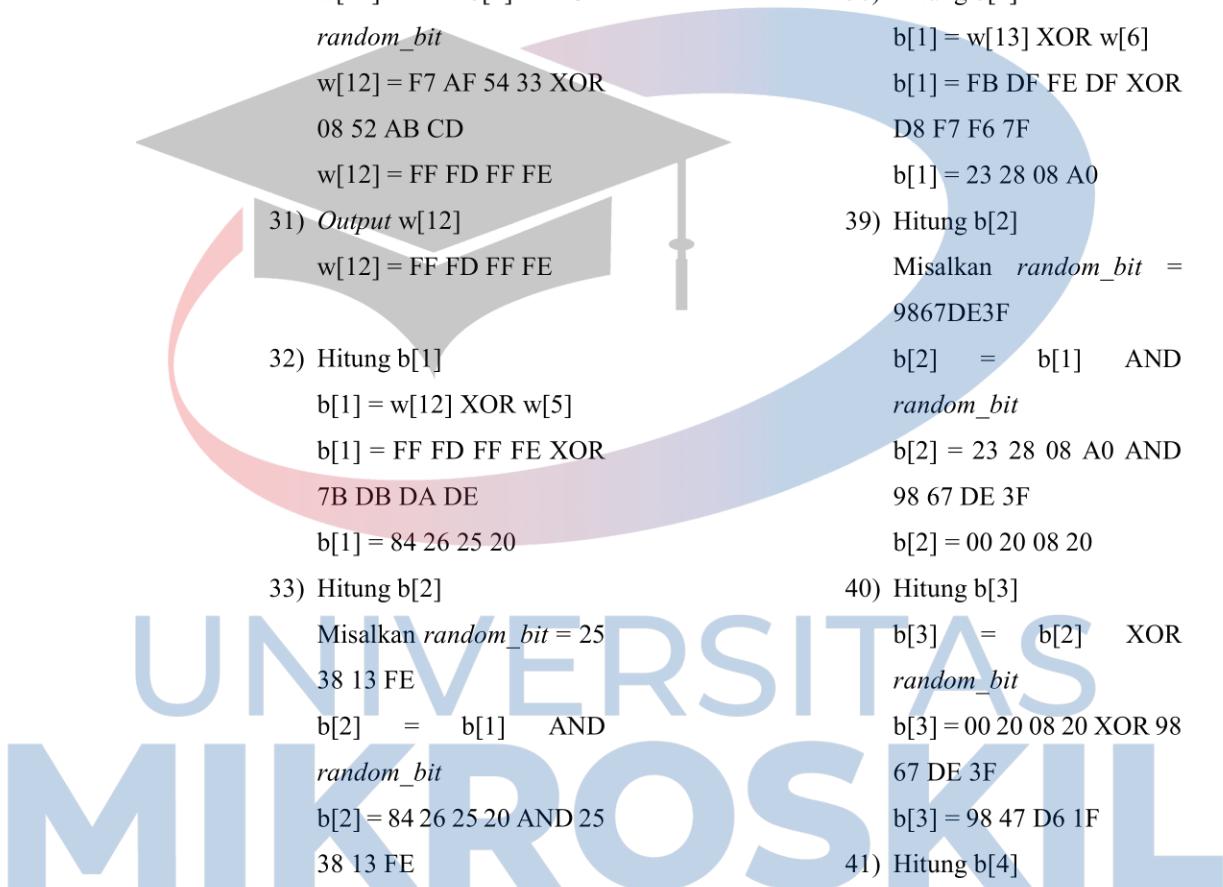


Gambar 3.5 Flowchart Proses Kerja Fungsi SRFG tanpa Penambahan RCI

Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) *Input W' = 09 FF 9B 7B*  
 $w[0] = B5 EF B7 EF$   
 $w[1] = DE DF 9F FB$   
 $w[2] = F5 EE A9 EB$   
 $w[3] = 5F DF DD D0$   
 $w[4] = ED FD EF FD$   
 $w[5] = 7B DB DA DE$   
 $w[6] = D8 F7 F6 7F$   
 $w[7] = 1D 7D BF AE$
- 2) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[0]$   
 $b[1] = 09 FF 9B 7B \text{ XOR }$   
 $B5 EF B7 EF$   
 $b[1] = BC 10 2C 94$
- 3) Hitung  $b[2]$   
Misalkan  $\text{random\_bit} = 4E115836$   
 $b[2] = b[1] \text{ AND }$   
 $\text{random\_bit}$   
 $b[2] = BC 10 2C 94 \text{ AND }$   
 $4E 11 58 36$   
 $b[2] = 0C 10 08 14$
- 4) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $\text{random\_bit}$   
 $b[3] = 0C 10 08 14 \text{ XOR }$   
 $4E 11 58 36$   
 $b[3] = 42 01 50 22$
- 5) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = BD FE AF DD$
- 6) Hitung  $w[8]$   
 $w[8] = b[4] \text{ XOR }$   
 $\text{random\_bit}$
- 7) *Output w[8]*  
 $w[8] = F3 EF F7 EB$
- 8) Hitung  $b[1]$   
 $b[1] = w[8] \text{ XOR } w[1]$   
 $b[1] = F3 EF F7 EB \text{ XOR }$   
 $DE DF 9F 7F$   
 $b[1] = 2D 30 68 10$
- 9) Hitung  $b[2]$   
Misalkan  $\text{random\_bit} = 2C679E7F$   
 $b[2] = b[1] \text{ AND }$   
 $\text{random\_bit}$   
 $b[2] = 2D 30 68 10 \text{ AND }$   
 $2C 67 9E 7F$   
 $b[2] = 2C 20 08 10$
- 10) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $\text{random\_bit}$   
 $b[3] = 2C 20 08 10 \text{ XOR }$   
 $2C 67 9E 7F$   
 $b[3] = 00 47 96 6F$
- 11) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = FF B8 69 90$
- 12) Hitung  $w[9]$   
 $w[9] = b[4] \text{ XOR }$   
 $\text{random\_bit}$   
 $w[9] = FF B8 69 90 \text{ XOR }$   
 $2C 67 9E 7F$   
 $w[9] = D3 DF F7 EF$
- 13) *Output w[9]*  
 $w[9] = D3 DF F7 EF$

- Misalkan  $random\_bit = BA\ 0B\ CC\ 55$
- 14) Hitung  $b[1]$   
 $b[1] = w[9] \text{ XOR } w[2]$   
 $b[1] = D3\ DF\ F7\ EF \text{ XOR }$   
 $F5\ EE\ A9\ EB$   
 $b[1] = 26\ 31\ 5E\ 04$
- 15) Hitung  $b[2]$   
Misalkan  $random\_bit = DD\ DA\ 10\ 3F$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 26\ 31\ 5E\ 04 \text{ AND }$   
 $DD\ DA\ 10\ 3F$   
 $b[2] = 04\ 10\ 10\ 04$
- 16) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 04\ 10\ 10\ 04 \text{ XOR }$   
 $DD\ DA\ 10\ 3F$   
 $b[3] = D9\ CA\ 00\ 3B$
- 17) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 26\ 35\ FF\ C4$
- 18) Hitung  $w[10]$   
 $w[10] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[10] = 26\ 35\ FF\ C4 \text{ XOR }$   
 $DD\ DA\ 10\ 3F$   
 $w[10] = FB\ EF\ EF\ FB$
- 19) Output  $w[10]$   
 $w[10] = FB\ EF\ EF\ FB$
- 20) Hitung  $b[1]$   
 $b[1] = w[10] \text{ XOR } w[3]$   
 $b[1] = FB\ EF\ EF\ FB \text{ XOR }$   
 $5F\ DF\ DD\ D0$   
 $b[1] = A4\ 30\ 32\ 2B$
- 21) Hitung  $b[2]$   
Misalkan  $random\_bit = BA\ 0B\ CC\ 55$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = A4\ 30\ 32\ 2B \text{ AND }$   
 $BA\ 0B\ CC\ 55$   
 $b[2] = A0\ 00\ 00\ 01$
- 22) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = A0\ 00\ 00\ 01 \text{ XOR }$   
 $BA\ 0B\ CC\ 55$   
 $b[3] = 1A\ 0B\ CC\ 54$
- 23) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = E5\ F4\ 33\ AB$
- 24) Hitung  $w[11]$   
 $w[11] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[11] = E5\ F4\ 33\ AB \text{ XOR }$   
 $BA\ 0B\ CC\ 55$   
 $w[11] = 5F\ FF\ FF\ FE$
- 25) Output  $w[11]$   
 $w[11] = 5F\ FF\ FF\ FE$
- 26) Hitung  $b[1]$   
 $b[1] = w[11] \text{ XOR } w[4]$   
 $b[1] = 5F\ FF\ FF\ FE \text{ XOR }$   
 $ED\ FD\ EF\ FD$   
 $b[1] = B2\ 02\ 10\ 03$
- 27) Hitung  $b[2]$   
Misalkan  $random\_bit = 0852ABCD$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = B2\ 02\ 10\ 03 \text{ AND }$   
 $08\ 52\ AB\ CD$   
 $b[2] = 00\ 02\ 00\ 01$
- 28) Hitung  $b[3]$



44) Hitung b[1]

$$b[1] = w[14] \text{ XOR } w[7]$$

$$b[1] = FF\ DF\ F7\ DF\ XOR$$

$$1D7D\ BF\ AE$$

$$b[1] = E2\ A2\ 48\ 71$$

45) Hitung b[2]

$$\text{Misalkan } random\_bit =$$

$$CB\ 9E\ 3F\ 5C$$

$$b[2] = b[1] \text{ AND }$$

$$random\_bit$$

$$b[2] = E2\ A2\ 48\ 71 \text{ AND }$$

$$CB\ 9E\ 3F\ 5C$$

$$b[2] = C2\ 82\ 08\ 50$$

46) Hitung b[3]

$$b[3] = b[2] \text{ XOR }$$

$$random\_bit$$

$$b[3] = C2\ 82\ 08\ 50 \text{ XOR }$$

$$CB\ 9E\ 3F\ 5C$$

$$b[3] = 09\ 1C\ 37\ 0C$$

47) Hitung b[4]

$$b[4] = \text{NOT } b[3]$$

$$b[4] = F6\ E3\ C8\ F3$$

48) Hitung w[15]

$$w[15] = b[4] \text{ XOR }$$

$$random\_bit$$

$$w[15] = F6\ E3\ C8\ F3 \text{ XOR }$$

$$CB\ 9E\ 3F\ 5C$$

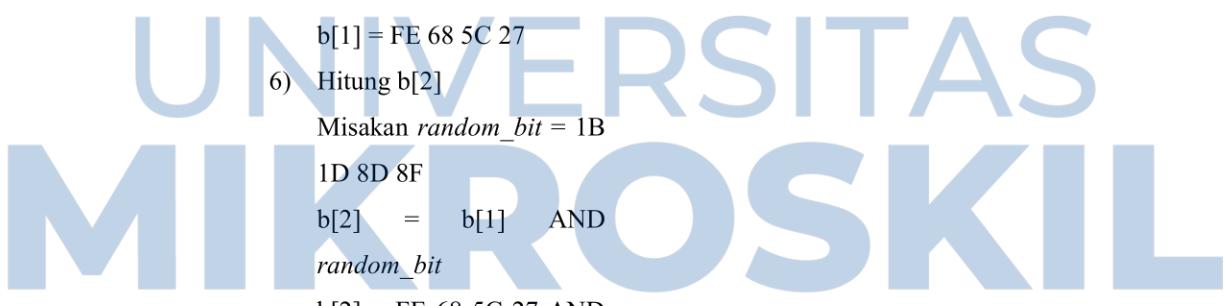
$$w[15] = 3D\ 7D\ F7\ A7$$

49) Output w[15]

$$w[15] = 3D\ 7D\ F7\ A7$$

Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- 1) Nilai  $W = w[15] = 3D\ 7D\ F7\ A7$
- 2) Pecahkan  $W$  menjadi 4 buah blok  $B$ :  
 $B[0] = 3D$   
 $B[1] = 7D$   
 $B[2] = F7$   
 $B[3] = A7$
- 3) Acak nilai  $B$  menjadi:  
 $B[0] = 7D$   
 $B[1] = F7$   
 $B[2] = A7$   
 $B[3] = 3D$
- 4) Eksekusi fungsi S-Box.  
 $B'[0] = \text{SBox}(7D) = FF$   
 $B'[1] = \text{SBox}(BF) = 68$   
 $B'[2] = \text{SBox}(AE) = 5C$   
 $B'[3] = \text{SBox}(1D) = 27$
- 5) Hitung  $b[1]$   
 $b[1] = RC1 \text{ XOR } B$   
 $b[1] = 01\ 00\ 00\ 00 \text{ XOR } FF$   
 $68\ 5C\ 27$   
 $b[1] = FE\ 68\ 5C\ 27$
- 6) Hitung  $b[2]$   
Misakan  $\text{random\_bit} = 1B$   
 $1D\ 8D\ 8F$   
 $b[2] = b[1] \text{ AND } \text{random\_bit}$   
 $b[2] = FE\ 68\ 5C\ 27 \text{ AND }$   
 $1B\ 1D\ 8D\ 8F$   
 $b[2] = 1A\ 08\ 0C\ 07$
- 7) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR } \text{random\_bit}$   
 $b[3] = 1A\ 08\ 0C\ 07 \text{ XOR }$   
 $1B\ 1D\ 8D\ 8F$   
 $b[3] = 01\ 15\ 81\ 88$
- 8) Hitung  $b[4]$
- 9) Hitung  $W'$   
 $W' = b[4] \text{ XOR } \text{random\_bit}$   
 $W' = FE\ EA\ 7E\ 7F \text{ XOR }$   
 $1B\ 1D\ 8D\ 8F$   
 $W' = E5\ F7\ F3\ F8$
- 10) Output  $W'$   
 $W' = E5\ F7\ F3\ F8$



Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1)  $Input W' = E5 F7 F3 F8$
- 2)  $w[8] = F3 EF F7 EB$   
 $w[9] = D3 DF F7 EF$   
 $w[10] = FB EF EF FB$   
 $w[11] = 5F FF FF FE$   
 $w[12] = FF FD FF FE$   
 $w[13] = FB DF FE DF$   
 $w[14] = FF DF F7 DF$   
 $w[15] = 3D 7D F7 A7$
- 3) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[8]$   
 $b[1] = E5 F7 F3 F8 \text{ XOR }$   
 $F3 EF F7 EB$   
 $b[1] = 16 18 04 13$
- 4) Hitung  $b[2]$   
Misalkan  $random\_bit = DF$   
 $FD AA BB$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 16 18 04 13 \text{ AND }$   
 $DF FD AA BB$   
 $b[2] = 16 18 00 13$
- 5) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 16 18 00 13 \text{ XOR }$   
 $DF FD AA BB$   
 $b[3] = C9 E5 AA A8$
- 6) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 36 1A 55 57$
- 7) Hitung  $w[16]$   
 $w[16] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[16] = 36 1A 55 57 \text{ XOR }$   
 $DF FD AA BB$
- 8)  $Output w[16]$   
 $w[16] = E9 E7 FF EC$
- 9) Hitung  $b[1]$   
 $b[1] = w[16] \text{ XOR } w[9]$   
 $b[1] = E9 E7 FF EC \text{ XOR }$   
 $D3 DF F7 EF$   
 $b[1] = 3A 38 08 03$
- 10) Hitung  $b[2]$   
Misalkan  $random\_bit = C3$   
 $C4 C5 FF$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 3A 38 08 03 \text{ AND }$   
 $C3 C4 C5 FF$   
 $b[2] = 02 00 00 03$
- 11) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 02 00 00 03 \text{ XOR } C3$   
 $C4 C5 FF$   
 $b[3] = C1 C4 C5 FC$
- 12) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 3E 3B 3A 03$
- 13) Hitung  $w[17]$   
 $w[17] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[17] = 3E 3B 3A 03 \text{ XOR }$   
 $C3 C4 C5 FF$   
 $w[17] = FD FF FF FC$
- 14)  $Output w[17]$   
 $w[17] = FD FF FF FC$
- 15) Hitung  $b[1]$   
 $b[1] = w[17] \text{ XOR } w[10]$

# UNIVERSITAS MIKROSKIL

- b[1] = FD FF FF FC XOR  
 FB EF EF FB  
 b[1] = 06 10 10 07
- 16) Hitung b[2]  
 Misalkan *random\_bit* = 14  
 15 89 CA  
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = 06 10 10 07 \text{ AND } 14$   
 15 89 CA  
 $b[2] = 04 10 00 02$
- 17) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 04 10 00 02 \text{ XOR } 14$   
 15 89 CA  
 $b[3] = 10 05 89 C8$
- 18) Hitung b[4]  
 $b[4] = \text{NOT } b[3]$   
 $b[4] = EF FA 76 37$
- 19) Hitung w[18]  
 $w[18] = b[4] \text{ XOR } random\_bit$   
 $w[18] = EF FA 76 37 \text{ XOR }$   
 14 15 89 CA  
 $w[18] = FB EF FF FD$
- 20) Output w[18]  
 $w[18] = FB EF FF FD$
- 21) Hitung b[1]  
 $b[1] = w[18] \text{ XOR } w[11]$   
 $b[1] = FB EF FF FD \text{ XOR }$   
 5F FF FF FE  
 $b[1] = A4 10 00 03$
- 22) Hitung b[2]  
 Misalkan *random\_bit* = BACA1D1A  
 $b[2] = b[1] \text{ AND } random\_bit$
- 23) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = A0 00 00 02 \text{ XOR }$   
 BA CA 1D 1A  
 $b[3] = 1A CA 1D 18$
- 24) Hitung b[4]  
 $b[4] = \text{NOT } b[3]$   
 $b[4] = E5 35 E2 E7$
- 25) Hitung w[19]  
 $w[19] = b[4] \text{ XOR } random\_bit$   
 $w[19] = E5 35 E2 E7 \text{ XOR }$   
 BA CA 1D 1A  
 $w[19] = 5F FF FF FD$
- 26) Output w[19]  
 $w[19] = 5F FF FF FD$
- 27) Hitung b[1]  
 $b[1] = w[19] \text{ XOR } w[12]$   
 $b[1] = 5F FF FF FD \text{ XOR }$   
 FF FD FF FE  
 $b[1] = A0 02 00 03$
- 28) Hitung b[2]  
 Misalkan *random\_bit* = 1DDDFC22  
 $b[2] = b[1] \text{ AND } random\_bit$   
 $b[2] = A0 02 00 03 \text{ AND }$   
 1D DD FC 22  
 $b[2] = 00 00 00 02$
- 29) Hitung b[3]  
 $b[3] = b[2] \text{ XOR } random\_bit$   
 $b[3] = 00 00 00 02 \text{ XOR }$   
 1D DD FC 22

- b[3] = 1DDD FC20
- 30) Hitung b[4]
- b[4] = NOT b[3]
- b[4] = E2 22 03 DF
- 31) Hitung w[20]
- w[20] = b[4] XOR  
random\_bit
- w[20] = E2 22 03 DF XOR  
1D DD FC 22
- w[20] = FF FF FF FD
- 32) Output w[20]
- w[20] = FF FF FF FD
- 33) Hitung b[1]
- b[1] = w[20] XOR w[13]
- b[1] = FF FF FF FD XOR  
FB DF FE DF
- b[1] = 04 20 01 22
- 34) Hitung b[2]
- Misalkan random\_bit = 2E  
EE FE CE
- b[2] = b[1] AND  
random\_bit
- b[2] = 04 20 01 22 AND 2E  
EE FE CE
- b[2] = 04 20 00 02
- 35) Hitung b[3]
- b[3] = b[2] XOR  
random\_bit
- b[3] = 04 20 00 02 XOR 2E  
EE FE CE
- b[3] = 2A CE FE CC
- 36) Hitung b[4]
- b[4] = NOT b[3]
- b[4] = D5 31 01 33
- 37) Hitung w[21]
- w[21] = b[4] XOR  
random\_bit
- w[21] = D5 31 01 33 XOR  
2E EE FE CE
- w[21] = FB DF FF FD
- 38) Output w[21]
- w[21] = FB DF FF FD
- 39) Hitung b[1]
- b[1] = w[21] XOR w[14]
- b[1] = FB DF FF FD XOR  
FF DF F7 DF
- b[1] = 04 00 08 22
- 40) Hitung b[2]
- Misalkan random\_bit = EC  
01 B2 D1
- b[2] = b[1] AND  
random\_bit
- b[2] = 04 00 08 22 AND  
EC 01 B2 D1
- b[2] = 04 00 00 00
- 41) Hitung b[3]
- b[3] = b[2] XOR  
random\_bit
- b[3] = 04 00 00 00 XOR  
EC 01 B2 D1
- b[3] = E8 01 B2 D1
- 42) Hitung b[4]
- b[4] = NOT b[3]
- b[4] = 17 FE 4D 2E
- 43) Hitung w[22]
- w[22] = b[4] XOR  
random\_bit
- w[22] = 17 FE 4D 2E  
XOR EC 01 B2 D1
- w[22] = FB FF FF FF
- 44) Output w[21]
- w[22] = FB FF FF FF
- 45) Hitung b[1]
- b[1] = w[22] XOR w[15]

$b[1] = FB FF FF FF XOR$

$3D 7D F7 A7$

$b[1] = 3D 7D F7 A7$

46) Hitung  $b[2]$

Misalkan  $random\_bit = 77$

$7A 77 7B$

$b[2] = b[1] AND$   
 $random\_bit$

$b[2] = 3D 7D F7 A7 AND$

$77 7A 77 7B$

$b[2] = CE 87 88 DC$

47) Hitung  $b[3]$

$b[3] = b[2] XOR$   
 $random\_bit$

$b[3] = CE 87 88 DC XOR$

$77 7A 77 7B$

$b[3] = B9 FD FF A7$

48) Hitung  $b[4]$

$b[4] = NOT b[3]$

$b[4] = 46 02 00 58$

49) Hitung  $w[23]$

$w[23] = b[4] XOR$   
 $random\_bit$

$w[23] = 46 02 00 58 XOR$

$77 7A 77 7B$

$w[23] = 31 78 77 23$

50) Output  $w[23]$

$w[23] = 31 78 77 23$

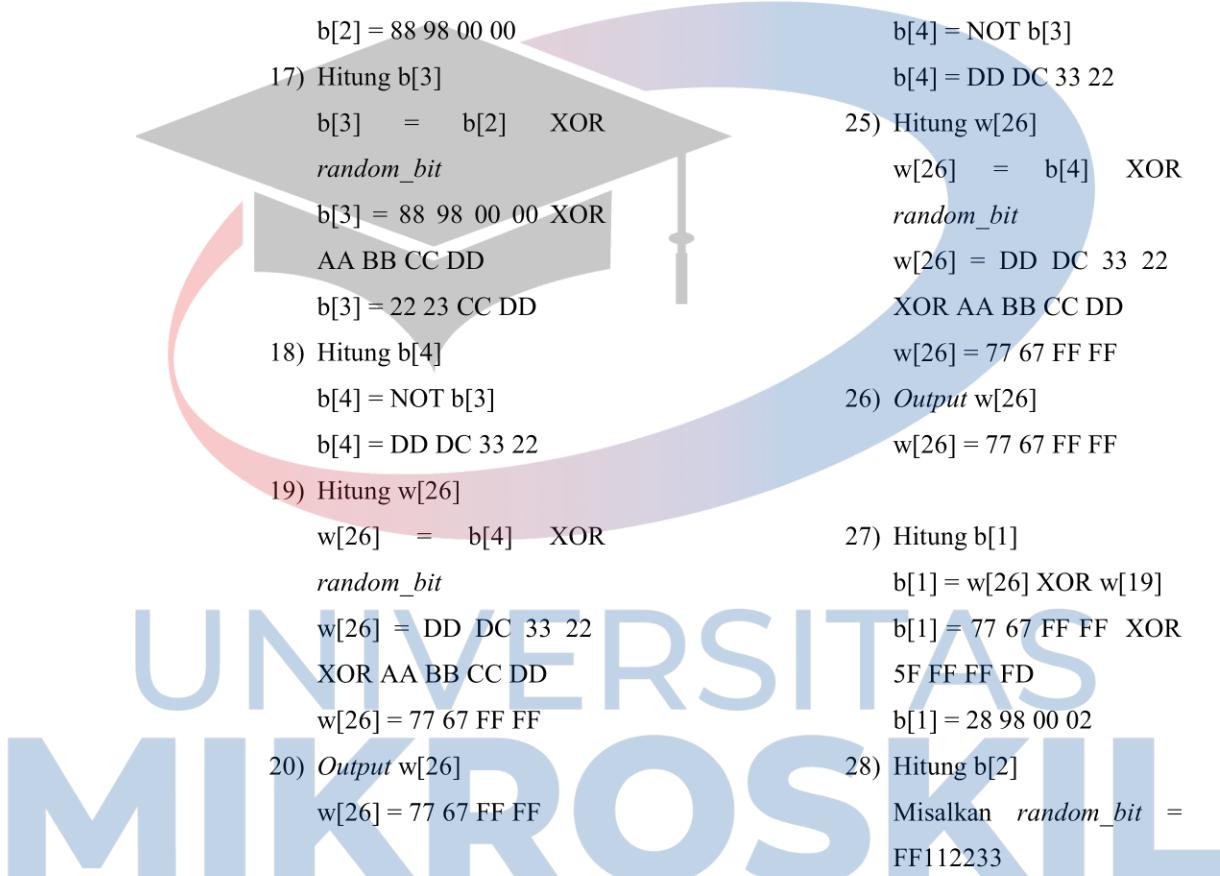


Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- 1) Nilai  $W = w[23] = 31\ 78\ 77\ 23$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = \text{FD F7 36 EF}$
- 2) Pecahkan  $W$  menjadi 4 buah blok  $B$ :  
 $B[0] = 31$   
 $B[1] = 78$   
 $B[2] = 77$   
 $B[3] = 23$   
9) Hitung  $W'$   
 $W' = b[4] \text{ XOR }$   
 $random\_bit$   
 $W' = \text{FD F7 36 EF XOR AB CD EF 12}$   
 $W' = 56\ 3A\ D9\ FD$
- 3) Acak nilai  $B$  menjadi:  
 $B[0] = 78$   
 $B[1] = 77$   
 $B[2] = 23$   
 $B[3] = 31$   
10) Output  $W'$   
 $W' = 56\ 3A\ D9\ FD$
- 4) Eksekusi fungsi S-Box.  
 $B'[0] = \text{SBox}(78) = BC$   
 $B'[1] = \text{SBox}(77) = F5$   
 $B'[2] = \text{SBox}(23) = 26$   
 $B'[3] = \text{SBox}(31) = C7$
- 5) Hitung  $b[1]$   
 $b[1] = RC1 \text{ XOR } B$   
 $b[1] = 01\ 00\ 00\ 00 \text{ XOR }$   
 $BC\ F5\ 26\ C7$   
 $b[1] = BD\ F5\ 26\ C7$   
6) Hitung  $b[2]$   
Misakan  $random\_bit = AB$   
 $CD\ EF\ 12$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = BD\ F5\ 26\ C7 \text{ AND }$   
 $AB\ CD\ EF\ 12$   
 $b[2] = A9\ C5\ 26\ 02$
- 7) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = A9\ C5\ 26\ 02 \text{ XOR }$   
 $AB\ CD\ EF\ 12$   
 $b[3] = 02\ 08\ C9\ 10$
- 8) Hitung  $b[4]$

Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) *Input W' = 56 3A D9 FD*
- 2)  $w[16] = E9 E7 FF EC$   
 $w[17] = FD FF FF FC$   
 $w[18] = FB EF FF FD$   
 $w[19] = 5F FF FF FD$   
 $w[20] = FF FF FF FD$   
 $w[21] = FB DF FF FD$   
 $w[22] = FB FF FF FF$   
 $w[23] = 31 78 77 23$
- 3) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[16]$   
 $b[1] = 56 3A D9 FD \text{ XOR }$   
 $E9 E7 FF EC$   
 $b[1] = BF DD 26 11$
- 4) Hitung  $b[2]$   
Misalkan  $random\_bit = CDABFEFF$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = BF DD 26 11 \text{ AND }$   
 $CD AB FE EF$   
 $b[2] = 8D 89 26 01$
- 5) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 8D 89 26 01 \text{ XOR }$   
 $CD AB FE EF$   
 $b[3] = 40 22 D8 EE$
- 6) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = BF DD 27 11$
- 7) Hitung  $w[24]$   
 $w[24] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[24] = BF DD 27 11$   
XOR CD AB FE EF  
 $w[24] = 72 76 D9 FE$
- 8) *Output w[24]*  
 $w[24] = 72 76 D9 FE$
- 9) Hitung  $b[1]$   
 $b[1] = w[24] \text{ XOR } w[17]$   
 $b[1] = 72 76 D9 FE \text{ XOR }$   
FD FF FF FC  
 $b[1] = 8F 89 26 02$
- 10) Hitung  $b[2]$   
Misalkan  $random\_bit = FDFEFAFC$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 8F 89 26 02 \text{ AND }$   
FD FE FA FC  
 $b[2] = 8D 88 22 00$
- 11) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 8D 88 22 00 \text{ XOR }$   
FD FE FA FC  
 $b[3] = 70 76 D8 FC$
- 12) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 8F 89 27 03$
- 13) Hitung  $w[25]$   
 $w[25] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[25] = 8F 89 27 03 \text{ XOR }$   
FD FE FA FC  
 $w[25] = 72 77 DD FF$
- 14) *Output w[25]*  
 $w[25] = 72 77 DD FF$
- 15) Hitung  $b[1]$   
 $b[1] = w[25] \text{ XOR } w[18]$



- $b[2] = 89\ 98\ 22\ 02\ AND$   
 AA BB CC DD  
 $b[2] = 88\ 98\ 00\ 00$   
 23) Hitung  $b[3]$   
 $b[3] = b[2] XOR$   
 $random\_bit$   
 $b[3] = 88\ 98\ 00\ 00\ XOR$   
 AA BB CC DD  
 $b[3] = 22\ 23\ CC\ DD$   
 24) Hitung  $b[4]$   
 $b[4] = NOT b[3]$   
 $b[4] = DD\ DC\ 33\ 22$   
 25) Hitung  $w[26]$   
 $w[26] = b[4] XOR$   
 $random\_bit$   
 $w[26] = DD\ DC\ 33\ 22$   
 $XOR\ AA\ BB\ CC\ DD$   
 $w[26] = 77\ 67\ FF\ FF$   
 26) Output  $w[26]$   
 $w[26] = 77\ 67\ FF\ FF$   
 27) Hitung  $b[1]$   
 $b[1] = w[26] XOR w[19]$   
 $b[1] = 77\ 67\ FF\ FF\ XOR$   
 $5F\ FF\ FF\ FD$   
 $b[1] = 28\ 98\ 00\ 02$   
 28) Hitung  $b[2]$   
 Misalkan  $random\_bit =$   
 FF112233  
 $b[2] = b[1] AND$   
 $random\_bit$   
 $b[2] = 28\ 98\ 00\ 02\ AND$   
 FF 11 22 33  
 $b[2] = 28\ 10\ 00\ 02$   
 29) Hitung  $b[3]$   
 $b[3] = b[2] XOR$   
 $random\_bit$   
 $b[3] = 28\ 10\ 00\ 02\ XOR\ FF$   
 11 22 33

# UNIVERSITAS MIKROSKIL

- b[3] = D7 01 22 31
- 30) Hitung b[4]  
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 28 \text{ FE DD CE}$
- 31) Hitung w[27]  
 $w[27] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[27] = 28 \text{ FE DD CE}$   
 $\text{XOR FF 11 22 33}$   
 $w[27] = D7 EF FF FD$
- 32) Output w[27]  
 $w[27] = D7 EF FF FD$
- 33) Hitung b[1]  
 $b[1] = w[27] \text{ XOR } w[20]$   
 $b[1] = D7 EF FF FD \text{ XOR }$   
 $FF FF FF FD$   
 $b[1] = 28 10 00 00$
- 34) Hitung b[2]  
 Misalkan  $random\_bit = 44556677$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 28 10 00 00 \text{ AND }$   
 $44 55 66 77$   
 $b[2] = 00 10 00 00$
- 35) Hitung b[3]  
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 00 10 00 00 \text{ XOR }$   
 $44 55 66 77$   
 $b[3] = 44 45 66 77$
- 36) Hitung b[4]  
 $b[4] = \text{NOT } b[3]$   
 $b[4] = BB BA 99 88$
- 37) Hitung w[28]  
 $w[28] = b[4] \text{ XOR }$   
 $random\_bit$
- w[28] = BB BA 99 88  
 $\text{XOR } 44 55 66 77$   
 $w[28] = FF EF FF FF$
- 38) Output w[28]  
 $w[28] = FF EF FF FF$
- 39) Hitung b[1]  
 $b[1] = w[28] \text{ XOR } w[21]$   
 $b[1] = FF EF FF FF \text{ XOR }$   
 $FB DF FF FD$   
 $b[1] = 04 30 00 02$
- 40) Hitung b[2]  
 Misalkan  $random\_bit = 88 99 AA BB$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 04 30 00 02 \text{ AND }$   
 $88 99 AA BB$   
 $b[2] = 00 10 00 02$
- 41) Hitung b[3]  
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 00 10 00 02 \text{ XOR }$   
 $88 99 AA BB$   
 $b[3] = 88 89 AA B9$
- 42) Hitung b[4]  
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 77 76 55 46$
- 43) Hitung w[29]  
 $w[29] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[29] = 77 76 55 46 \text{ XOR }$   
 $88 99 AA BB$   
 $w[29] = FF EF FF FD$
- 44) Output w[29]  
 $w[29] = FF EF FF FD$
- 45) Hitung b[1]  
 $b[1] = w[29] \text{ XOR } w[22]$



Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- 1) Nilai  $W = w[23] = B5\ FB$   
 $FF\ FF$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 4F\ 37\ 9F\ F7$
- 2) Pecahkan  $W$  menjadi 4 buah blok  $B$ :  
 $B[0] = B5$   
 $B[1] = FB$   
 $B[2] = FF$   
 $B[3] = FF$   
 $W' = 4F\ 37\ 9F\ F7\ \text{XOR}\ BA$   
 $DC\ 72\ 19$   
 $W' = F5\ EB\ ED\ EE$
- 3) Acak nilai  $B$  menjadi:  
 $B[0] = FB$   
 $B[1] = FF$   
 $B[2] = FF$   
 $B[3] = B5$   
 $W' = F5\ EB\ ED\ EE$
- 4) Eksekusi fungsi S-Box.  
 $B'[0] = \text{SBox}(FB) = 0F$   
 $B'[1] = \text{SBox}(FF) = 16$   
 $B'[2] = \text{SBox}(FF) = 16$   
 $B'[3] = \text{SBox}(B5) = D5$   
 $W' = F5\ EB\ ED\ EE$
- 5) Hitung  $b[1]$   
 $b[1] = RC1\ \text{XOR}\ B$   
 $b[1] = 01\ 00\ 00\ 00\ \text{XOR}\ 0F$   
 $16\ 16\ D5$   
 $b[1] = 0E\ 16\ 16\ D5$   
 $W' = F5\ EB\ ED\ EE$
- 6) Hitung  $b[2]$   
Misakan  $\text{random\_bit} = BA$   
 $DC\ 72\ 19$   
 $b[2] = b[1]\ \text{AND}$   
 $\text{random\_bit}$   
 $b[2] = 0E\ 16\ 16\ D5\ \text{AND}$   
 $BA\ DC\ 72\ 19$   
 $b[2] = 0A\ 14\ 12\ 11$   
 $W' = F5\ EB\ ED\ EE$
- 7) Hitung  $b[3]$   
 $b[3] = b[2]\ \text{XOR}$   
 $\text{random\_bit}$   
 $b[3] = 0A\ 14\ 12\ 11\ \text{XOR}$   
 $BA\ DC\ 72\ 19$   
 $b[3] = B0\ C8\ 60\ 08$   
 $W' = F5\ EB\ ED\ EE$
- 8) Hitung  $b[4]$   
 $W' = F5\ EB\ ED\ EE$

UNIVERSITAS  
MIKROSKIL

Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) *Input W' = F5 EB ED EE*
- 2)  $w[24] = 72\ 76\ D9\ FE$   
 $w[25] = 72\ 77\ DD\ FF$   
 $w[26] = 77\ 67\ FF\ FF$   
 $w[27] = D7\ EF\ FF\ DD$   
 $w[28] = FF\ EF\ FF\ FF$   
 $w[29] = FF\ EF\ FF\ FD$   
 $w[30] = FF\ EF\ FF\ FD$   
 $w[31] = B5\ FB\ FF\ FF$
- 3) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[24]$   
 $b[1] = F5\ EB\ ED\ EE\ \text{XOR}$   
 $72\ 76\ D9\ FE$   
 $b[1] = 87\ 9D\ 34\ 10$
- 4) Hitung  $b[2]$   
Misalkan  $random\_bit =$   
 $AC\ 12\ 46\ 81$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 87\ 9D\ 34\ 10 \text{ AND }$   
 $AC\ 12\ 46\ 81$   
 $b[2] = 84\ 10\ 04\ 00$
- 5) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 84\ 10\ 04\ 00 \text{ XOR }$   
 $AC\ 12\ 46\ 81$   
 $b[3] = 28\ 02\ 42\ 81$
- 6) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = D7\ FD\ BD\ 7E$
- 7) Hitung  $w[32]$   
 $w[32] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[32] = D7\ FD\ BD\ 7E$   
 $\text{XOR } AC\ 12\ 46\ 81$   
 $w[32] = 7B\ EF\ FB\ FF$
- 8) *Output w[32]*  
 $w[32] = 7B\ EF\ FB\ FF$
- 9) Hitung  $b[1]$   
 $b[1] = w[32] \text{ XOR }$   
 $w[25]$   
 $b[1] = 7B\ EF\ FB\ FF\ \text{XOR}$   
 $72\ 77\ DD\ FF$   
 $b[1] = 09\ 98\ 26\ 00$
- 10) Hitung  $b[2]$   
Misalkan  $random\_bit =$   
 $DA\ 43\ FA\ C1$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 09\ 98\ 26\ 00 \text{ AND }$   
 $DA\ 43\ FA\ C1$   
 $b[2] = 08\ 00\ 22\ 00$
- 11) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 08\ 00\ 22\ 00 \text{ XOR }$   
 $DA\ 43\ FA\ C1$   
 $b[3] = D2\ 43\ D8\ C1$
- 12) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 2D\ BC\ 27\ 3E$
- 13) Hitung  $w[33]$   
 $w[33] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[33] = 2D\ BC\ 27\ 3E$   
 $\text{XOR } DA\ 43\ FA\ C1$   
 $w[33] = F7\ FF\ DD\ FF$
- 14) *Output w[33]*  
 $w[33] = F7\ FF\ DD\ FF$
- 15) Hitung  $b[1]$

- b[1] = w[33] XOR  
w[26]
- b[1] = F7 FF DD FF XOR  
77 67 FF FF
- b[1] = 80 98 22 00
- 16) Hitung b[2]  
Misalkan *random\_bit* =  
CD C2 1F AA
- b[2] = b[1] AND  
*random\_bit*
- b[2] = 80 98 22 00 AND  
CD C2 1F AA
- b[2] = 80 80 02 00
- 17) Hitung b[3]  
b[3] = b[2] XOR  
*random\_bit*
- b[3] = 80 80 02 00 XOR  
CD C2 1F AA
- b[3] = 4D 42 1D AA
- 18) Hitung b[4]  
b[4] = NOT b[3]  
b[4] = B2 BD E2 55
- 19) Hitung w[34]  
w[34] = b[4] XOR  
*random\_bit*
- w[34] = B2 BD E2 55  
XOR CD C2 1F AA
- w[34] = 7F 7F FD FF
- 20) Output w[34]  
w[34] = 7F 7F FD FF
- 21) Hitung b[1]  
b[1] = w[34] XOR  
w[27]
- b[1] = 7F 7F FD FF XOR  
D7 EF FF DD
- b[1] = A8 90 02 22
- 22) Hitung b[2]
- Misalkan *random\_bit* = AF  
C1 DB E7
- b[2] = b[1] AND  
*random\_bit*
- b[2] = A8 90 02 22 AND  
AF C1 DB E7
- b[2] = A8 80 02 22
- 23) Hitung b[3]  
b[3] = b[2] XOR  
*random\_bit*
- b[3] = A8 80 02 22 XOR  
AF C1 DB E7
- b[3] = 07 41 D9 C5
- 24) Hitung b[4]  
b[4] = NOT b[3]  
b[4] = F8 BE 26 3A
- 25) Hitung w[35]  
w[35] = b[4] XOR  
*random\_bit*
- w[35] = F8 BE 26 3A  
XOR AF C1 DB E7
- w[35] = 57 7F FD DD
- 26) Output w[35]  
w[35] = 57 7F FD DD
- 27) Hitung b[1]  
b[1] = w[35] XOR  
w[28]
- b[1] = 57 7F FD DD XOR  
FF EF FF FF
- b[1] = A8 90 02 22
- 28) Hitung b[2]  
Misalkan *random\_bit* = CF  
12 44 7A
- b[2] = b[1] AND  
*random\_bit*
- b[2] = A8 90 02 22 AND  
CF 12 44 7A
- b[2] = 88 10 00 22

- 29) Hitung b[3]
- $$b[3] = b[2] \text{ XOR } random\_bit$$
- $$b[3] = 88\ 10\ 00\ 22 \text{ XOR }$$
- $$\text{CF}\ 12\ 44\ 7A$$
- $$b[3] = 47\ 02\ 44\ 58$$
- 30) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = B8\ FD\ BB\ A7$$
- 31) Hitung w[36]
- $$w[36] = b[4] \text{ XOR } random\_bit$$
- $$w[36] = B8\ FD\ BB\ A7$$
- $$\text{XOR CF}\ 12\ 44\ 7A$$
- $$w[36] = 77\ EF\ FF\ DD$$
- 32) Output w[36]
- $$w[36] = 77\ EF\ FF\ DD$$
- 33) Hitung b[1]
- $$b[1] = w[36] \text{ XOR } w[29]$$
- $$b[1] = 77\ EF\ FF\ DD \text{ XOR }$$
- $$\text{FF}\ EF\ FF\ FD$$
- $$b[1] = 88\ 00\ 00\ 20$$
- 34) Hitung b[2]
- Misalkan  $random\_bit = FC\ 22\ AE\ 81$
- $$b[2] = b[1] \text{ AND } random\_bit$$
- $$b[2] = 88\ 00\ 00\ 20 \text{ AND }$$
- $$\text{FC}\ 22\ AE\ 81$$
- $$b[2] = 88\ 00\ 00\ 00$$
- 35) Hitung b[3]
- $$b[3] = b[2] \text{ XOR } random\_bit$$
- $$b[3] = 88\ 00\ 00\ 00 \text{ XOR }$$
- $$\text{FC}\ 22\ AE\ 81$$
- $$b[3] = 74\ 22\ AE\ 81$$
- 36) Hitung b[4]
- b[4] = NOT b[3]
- $$b[4] = 8B\ DD\ 51\ 7E$$
- 37) Hitung w[37]
- $$w[37] = b[4] \text{ XOR } random\_bit$$
- $$w[37] = 8B\ DD\ 51\ 7E$$
- $$\text{XOR FC}\ 22\ AE\ 81$$
- $$w[37] = 77\ FF\ FF\ FF$$
- 38) Output w[37]
- $$w[37] = 77\ FF\ FF\ FF$$
- 39) Hitung b[1]
- $$b[1] = w[37] \text{ XOR } w[30]$$
- $$b[1] = 77\ FF\ FF\ FF \text{ XOR }$$
- $$\text{FF}\ EF\ FF\ FD$$
- $$b[1] = 88\ 10\ 00\ 02$$
- 40) Hitung b[2]
- Misalkan  $random\_bit = CA\ 1F\ 23\ DD$
- $$b[2] = b[1] \text{ AND } random\_bit$$
- $$b[2] = 88\ 10\ 00\ 02 \text{ AND }$$
- $$\text{CA}\ 1F\ 23\ DD$$
- $$b[2] = 88\ 10\ 00\ 00$$
- 41) Hitung b[3]
- $$b[3] = b[2] \text{ XOR } random\_bit$$
- $$b[3] = 88\ 10\ 00\ 00 \text{ XOR }$$
- $$\text{CA}\ 1F\ 23\ DD$$
- $$b[3] = 42\ 0F\ 23\ DD$$
- 42) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = BD\ F0\ DC\ 22$$
- 43) Hitung w[38]
- $$w[38] = b[4] \text{ XOR } random\_bit$$
- $$w[38] = BD\ F0\ DC\ 22$$
- $$\text{XOR CA}\ 1F\ 23\ DD$$

$w[38] = 77\ EF\ FF\ FF$

44) *Output w[38]*

$w[38] = 77\ EF\ FF\ FF$

45) Hitung  $b[1]$

$b[1] = w[38] \text{ XOR}$

$w[31]$

$b[1] = 77\ EF\ FF\ FF \text{ XOR}$

B5 FB FF FF

$b[1] = C2\ 14\ 00\ 00$

46) Hitung  $b[2]$

Misalkan  $\text{random\_bit} = F1$

DA C1 15

$b[2] = b[1] \text{ AND}$

$\text{random\_bit}$

$b[2] = C2\ 14\ 00\ 00 \text{ AND}$

F1 DA C1 15

$b[2] = C0\ 10\ 00\ 00$

47) Hitung  $b[3]$

$b[3] = b[2] \text{ XOR}$

$\text{random\_bit}$

$b[3] = C0\ 10\ 00\ 00 \text{ XOR}$

F1 DA C1 15

$b[3] = 31\ CA\ C1\ 15$

48) Hitung  $b[4]$

$b[4] = \text{NOT } b[3]$

$b[4] = CE\ 35\ 3E\ EA$

49) Hitung  $w[39]$

$w[39] = b[4] \text{ XOR}$

$\text{random\_bit}$

$w[39] = CE\ 35\ 3E\ EA$

XOR F1 DA C1 15

$w[39] = 3F\ EF\ FF\ FF$

50) *Output w[39]*

$w[39] = 3F\ EF\ FF\ FF$

Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- 1) Nilai  $W = w[39] = 3F\ EF$   
 $FF\ FF$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = FE\ 37\ 57\ F7$
- 2) Pecahkan  $W$  menjadi 4 buah blok  $B$ :  
 $B[0] = 3F$   
 $B[1] = EF$   
 $B[2] = FF$   
 $B[3] = FF$   
 $b[4] = \text{random\_bit}$   
 $W' = FE\ 37\ 57\ F7\ XOR\ DD\ DA\ AA\ 19$   
 $W' = 23\ ED\ FD\ EE$
- 3) Acak nilai  $B$  menjadi:  
 $B[0] = EF$   
 $B[1] = FF$   
 $B[2] = FF$   
 $B[3] = 3F$   
 $W' = 23\ ED\ FD\ EE$
- 4) Eksekusi fungsi S-Box.  
 $B'[0] = \text{SBox}(EF) = DF$   
 $B'[1] = \text{SBox}(FF) = 16$   
 $B'[2] = \text{SBox}(FF) = 16$   
 $B'[3] = \text{SBox}(3F) = 75$   
 $W' = 23\ ED\ FD\ EE$
- 5) Hitung  $b[1]$   
 $b[1] = RC1\ XOR\ B$   
 $b[1] = 01\ 00\ 00\ 00\ XOR\ DF\ 16\ 16\ 75$   
 $b[1] = DE\ 16\ 16\ 75$   
 $W' = 23\ ED\ FD\ EE$
- 6) Hitung  $b[2]$   
Misakan  $\text{random\_bit} = DD$   
 $DA\ AA\ 19$   
 $b[2] = b[1]\ AND\ \text{random\_bit}$   
 $b[2] = DE\ 16\ 16\ 75\ AND\ DD\ DA\ AA\ 19$   
 $b[2] = DC\ 12\ 02\ 11$   
 $W' = 23\ ED\ FD\ EE$
- 7) Hitung  $b[3]$   
 $b[3] = b[2]\ XOR\ \text{random\_bit}$   
 $b[3] = DC\ 12\ 02\ 11\ XOR\ DD\ DA\ AA\ 19$   
 $b[3] = 01\ C8\ A8\ 08$   
 $W' = 23\ ED\ FD\ EE$
- 8) Hitung  $b[4]$   
 $W' = 23\ ED\ FD\ EE$

UNIVERSITAS  
MIKROSKIL

Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) *Input W' = 23 ED FD EE*
- 2)  $w[32] = 7B\ EF\ FB\ FF$   
 $w[33] = F7\ FF\ DD\ FF$   
 $w[34] = 7F\ 7F\ FD\ FF$   
 $w[35] = 57\ 7F\ FD\ DD$   
 $w[36] = 77\ EF\ FF\ DD$   
 $w[37] = 77\ FF\ FF\ FF$   
 $w[38] = 77\ EF\ FF\ FF$   
 $w[39] = 3F\ EF\ FF\ FF$
- 3) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[32]$   
 $b[1] = 23\ ED\ FD\ EE\ \text{XOR}$   
 $7B\ EF\ FB\ FF$   
 $b[1] = 58\ 02\ 06\ 11$
- 4) Hitung  $b[2]$   
Misalkan  $\text{random\_bit} = 77$   
 $7A\ CF\ 12$   
 $b[2] = b[1] \text{ AND }$   
 $\text{random\_bit}$   
 $b[2] = 58\ 02\ 06\ 11 \text{ AND } 77$   
 $7A\ CF\ 12$   
 $b[2] = 50\ 02\ 06\ 10$
- 5) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $\text{random\_bit}$   
 $b[3] = 50\ 02\ 06\ 10 \text{ XOR } 77$   
 $7A\ CF\ 12$   
 $b[3] = 27\ 78\ C9\ 02$
- 6) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = D8\ 87\ 36\ FD$
- 7) Hitung  $w[40]$   
 $w[40] = b[4] \text{ XOR }$   
 $\text{random\_bit}$   
 $w[40] = D8\ 87\ 36\ FD$   
 $\text{XOR } 77\ 7A\ CF\ 12$
- 8) *Output w[40]*  
 $w[40] = AF\ FD\ F9\ EF$
- 9) Hitung  $b[1]$   
 $b[1] = w[40] \text{ XOR } w[33]$   
 $b[1] = AF\ FD\ F9\ EF \text{ XOR }$   
 $F7\ FF\ DD\ FF$   
 $b[1] = 58\ 02\ 24\ 10$
- 10) Hitung  $b[2]$   
Misalkan  $\text{random\_bit} = 61$   
 $2B\ 43\ 9C$   
 $b[2] = b[1] \text{ AND }$   
 $\text{random\_bit}$   
 $b[2] = 58\ 02\ 24\ 10 \text{ AND } 61$   
 $2B\ 43\ 9C$   
 $b[2] = 40\ 02\ 00\ 10$
- 11) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $\text{random\_bit}$   
 $b[3] = 40\ 02\ 00\ 10 \text{ XOR } 61$   
 $2B\ 43\ 9C$   
 $b[3] = 21\ 29\ 43\ 8C$
- 12) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = DE\ D6\ BC\ 73$
- 13) Hitung  $w[41]$   
 $w[41] = b[4] \text{ XOR }$   
 $\text{random\_bit}$   
 $w[41] = DE\ D6\ BC\ 73$   
 $\text{XOR } 61\ 2B\ 43\ 9C$   
 $w[41] = BF\ FD\ FF\ EF$
- 14) *Output w[41]*  
 $w[41] = BF\ FD\ FF\ EF$
- 15) Hitung  $b[1]$   
 $b[1] = w[41] \text{ XOR } w[34]$



- b[1] = BF FD FF EF XOR  
7F 7F FD FF
- b[1] = C0 82 02 10
- 16) Hitung b[2]  
Misalkan *random\_bit* =  
A5612BCA  
b[2] = b[1] AND  
*random\_bit*  
b[2] = C0 82 02 10 AND  
A5 61 2B CA  
b[2] = 80 00 02 00
- 17) Hitung b[3]  
b[3] = b[2] XOR  
*random\_bit*  
b[3] = 80 00 02 00 XOR  
A5 61 2B CA  
b[3] = 25 61 29 CA
- 18) Hitung b[4]  
b[4] = NOT b[3]  
b[4] = DA 9E D6 35
- 19) Hitung w[42]  
w[42] = b[4] XOR  
*random\_bit*  
w[42] = DA 9E D6 35  
XOR A5 61 2B CA  
w[42] = 7F FF FD FF
- 20) Output w[42]  
w[42] = 7F FF FD FF
- 21) Hitung b[1]  
b[1] = w[42] XOR w[35]  
b[1] = 7F FF FD FF XOR  
57 7F FD DD  
b[1] = 28 80 00 22
- 22) Hitung b[2]  
Misalkan *random\_bit* =  
90AA0CF0  
b[2] = b[1] AND  
*random\_bit*
- 23) Hitung b[3]  
b[3] = b[2] XOR  
*random\_bit*  
b[3] = 00 80 00 20 XOR 90  
AA 0C F0  
b[3] = 90 2A 0C D0
- 24) Hitung b[4]  
b[4] = NOT b[3]  
b[4] = 6F D5 F3 2F
- 25) Hitung w[43]  
w[43] = b[4] XOR  
*random\_bit*  
w[43] = 6F D5 F3 2F XOR  
90 AA 0C F0  
w[43] = FF 7F FF DF
- 26) Output w[43]  
w[43] = FF 7F FF DF
- 27) Hitung b[1]  
b[1] = w[43] XOR w[36]  
b[1] = FF 7F FF DF XOR  
77 EF FF DD  
b[1] = 88 90 00 02
- 28) Hitung b[2]  
Misalkan *random\_bit* =  
C67190CD  
b[2] = b[1] AND  
*random\_bit*  
b[2] = 88 90 00 02 AND  
C6 71 90 CD  
b[2] = 80 10 00 00
- 29) Hitung b[3]  
b[3] = b[2] XOR  
*random\_bit*  
b[3] = 80 10 00 00 XOR C6  
71 90 CD

# UNIVERSITAS MIKROSKIL

- b[3] = 46 61 90 CD
- 30) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = B9 9E 6F 32$$
- 31) Hitung w[44]
- $$w[44] = b[4] \text{ XOR }$$
- $$\text{random\_bit}$$
- $$w[44] = B9 9E 6F 32 \text{ XOR }$$
- $$C6 71 90 CD$$
- $$w[44] = 7F EF FF FF$$
- 32) Output w[44]
- $$w[44] = 7F EF FF FF$$
- 33) Hitung b[1]
- $$b[1] = w[44] \text{ XOR } w[37]$$
- $$b[1] = 7F EF FF FF \text{ XOR }$$
- $$77 FF FF FF$$
- $$b[1] = 08 10 00 00$$
- 34) Hitung b[2]
- Misalkan random\_bit = 4E010F0B
- $$b[2] = b[1] \text{ AND }$$
- $$\text{random\_bit}$$
- $$b[2] = 0810 0000 \text{ AND } 4E$$
- $$01 0F 0B$$
- $$b[2] = 08 00 00 00$$
- 35) Hitung b[3]
- $$b[3] = b[2] \text{ XOR }$$
- $$\text{random\_bit}$$
- $$b[3] = 08 00 00 00 \text{ XOR } 4E$$
- $$01 0F 0B$$
- $$b[3] = 46 01 0F 0B$$
- 36) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = B9 FE F0 F4$$
- 37) Hitung w[45]
- $$w[45] = b[4] \text{ XOR }$$
- $$\text{random\_bit}$$
- w[45] = B9 FE F0 F4 XOR
- 4E 01 0F 0B
- w[45] = F7 FF FF FF
- 38) Output w[45]
- $$w[45] = F7 FF FF FF$$
- 39) Hitung b[1]
- $$b[1] = w[45] \text{ XOR } w[38]$$
- $$b[1] = F7 FF FF FF \text{ XOR }$$
- $$77 EF FF FF$$
- $$b[1] = 80 10 00 00$$
- 40) Hitung b[2]
- Misalkan random\_bit = C7
- $$AC 01 20$$
- $$b[2] = b[1] \text{ AND }$$
- $$\text{random\_bit}$$
- $$b[2] = 80 10 00 00 \text{ AND }$$
- $$C7 AC 01 20$$
- $$b[2] = 80 00 00 00$$
- 41) Hitung b[3]
- $$b[3] = b[2] \text{ XOR }$$
- $$\text{random\_bit}$$
- $$b[3] = 08 00 00 00 \text{ XOR } C7$$
- $$AC 01 20$$
- $$b[3] = 47 AC 01 20$$
- 42) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = B8 53 FE DF$$
- 43) Hitung w[46]
- $$w[46] = b[4] \text{ XOR }$$
- $$\text{random\_bit}$$
- $$w[46] = B8 53 FE DF \text{ XOR }$$
- $$C7 AC 01 20$$
- $$w[46] = 7F FF FF FF$$
- 44) Output w[46]
- $$w[46] = 7F FF FF FF$$
- 45) Hitung b[1]
- $$b[1] = w[46] \text{ XOR } w[39]$$

$b[1] = 7F FF FF FF$  XOR

3F EF FF FF

$b[1] = 40\ 10\ 00\ 00$

46) Hitung  $b[2]$

Misalkan  $random\_bit =$

CA CF A9 CC

$b[2] = b[1] \text{ AND}$

$random\_bit$

$b[2] = 40\ 10\ 00\ 00$  AND

CA CF A9 CC

$b[2] = 40\ 00\ 00\ 00$

47) Hitung  $b[3]$

$b[3] = b[2] \text{ XOR}$

$random\_bit$

$b[3] = 40\ 00\ 00\ 00$  XOR

CA CF A9 CC

$b[3] = 8A\ CF\ A9\ CC$

48) Hitung  $b[4]$

$b[4] = \text{NOT } b[3]$

$b[4] = 75\ 30\ 56\ 33$

49) Hitung  $w[47]$

$w[47] = b[4] \text{ XOR}$

$random\_bit$

$w[47] = 75\ 30\ 56\ 33$  XOR

CA CF A9 CC

$w[47] = BF\ FF\ FF\ FF$

50) Output  $w[47]$

$w[47] = BF\ FF\ FF\ FF$

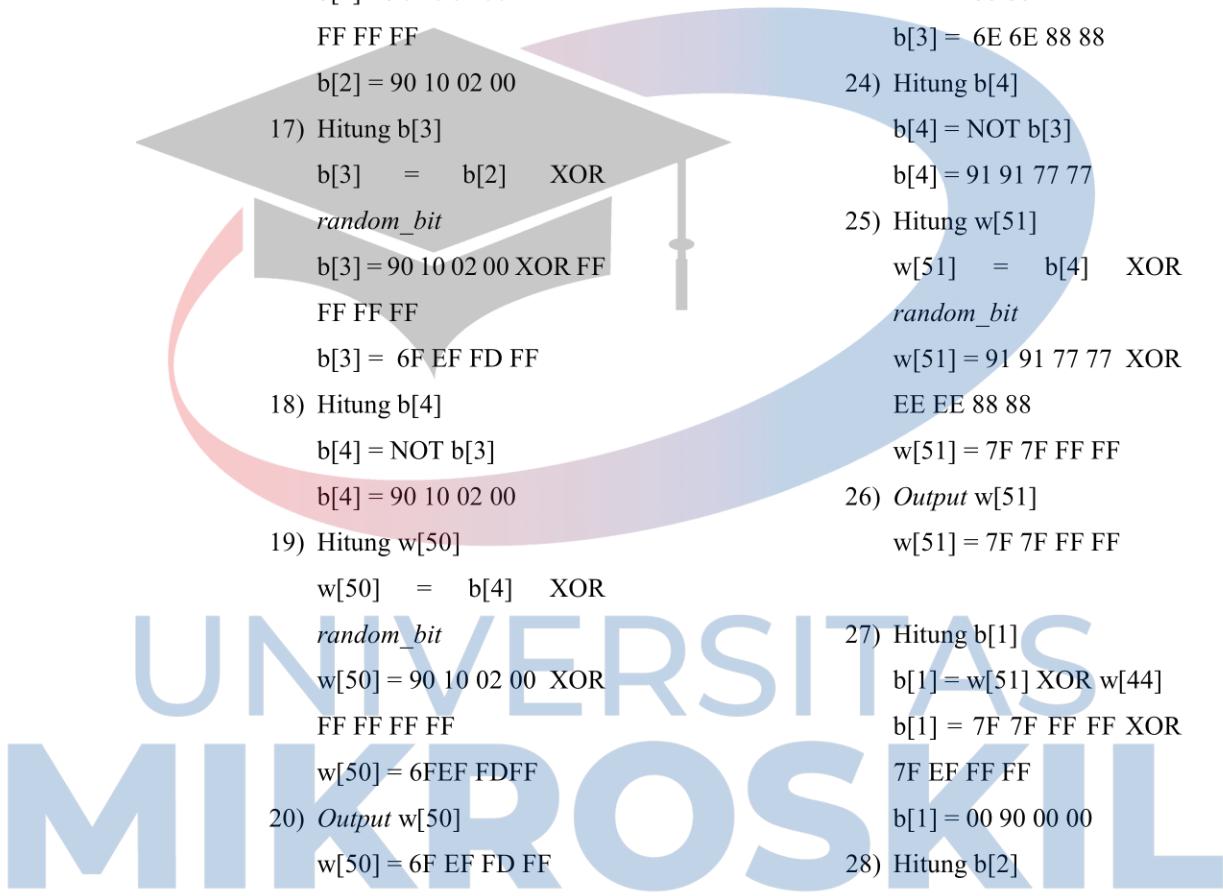


Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

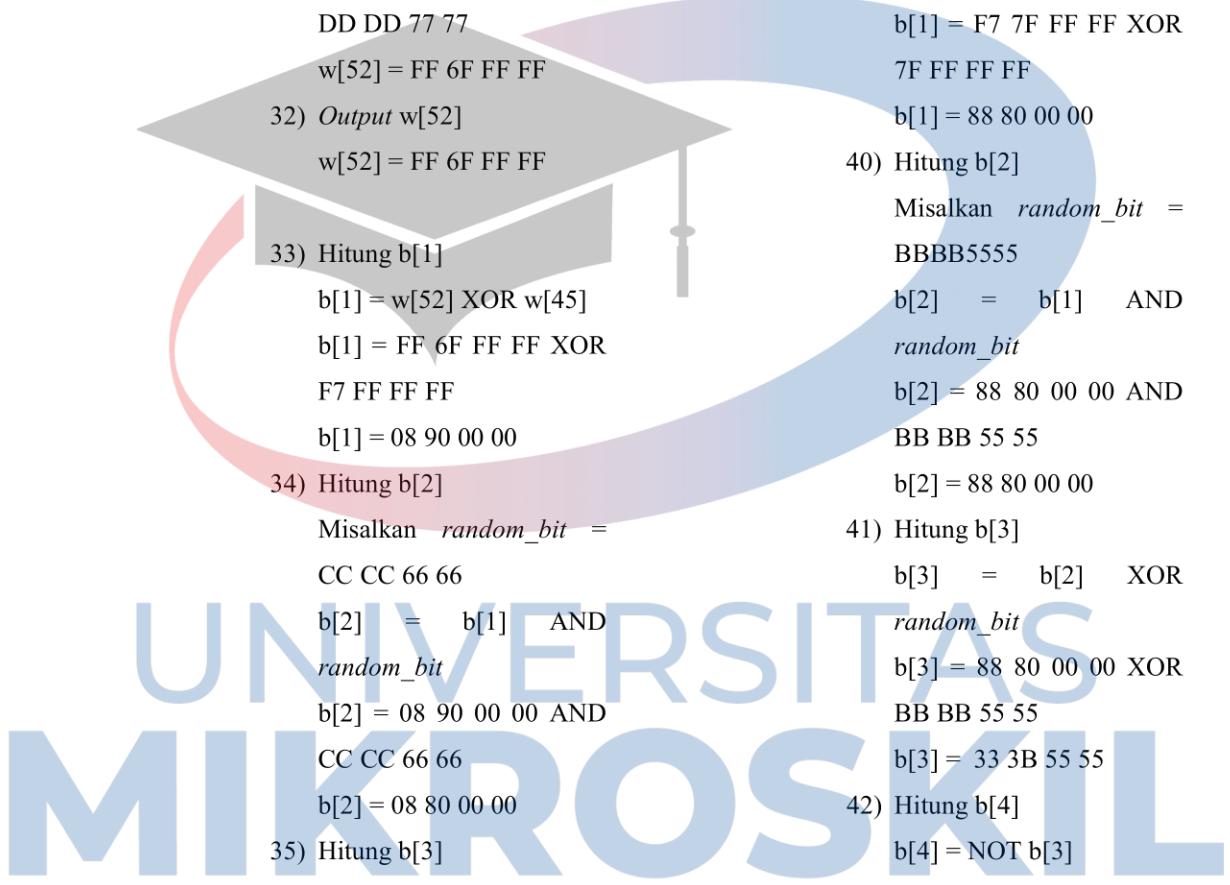
- 
- 1) Nilai  $W = w[47] = BF\ FF$   
 $FF\ FF$   
 $b[4] = NOT\ b[3]$   
 $b[4] = 5F\ 37\ 36\ EE$
  - 2) Pecahkan  $W$  menjadi 4 buah blok  $B$ :  
 $B[0] = BF$   
 $B[1] = FF$   
 $B[2] = FF$   
 $B[3] = FF$   
 $9) \text{ Hitung } W'$   
 $W' = b[4] \text{ XOR }$   
 $random\_bit$   
 $W' = 5F\ 37\ 36\ EE \text{ XOR } A0$   
 $DA\ C9\ 11$   
 $W' = FF\ ED\ FF\ FF$
  - 3) Acak nilai  $B$  menjadi:  
 $B[0] = FF$   
 $B[1] = FF$   
 $B[2] = FF$   
 $B[3] = BF$   
 $10) \text{ Output } W'$   
 $W' = FF\ ED\ FF\ FF$
  - 4) Eksekusi fungsi S-Box.  
 $B'[0] = SBox(FF) = 16$   
 $B'[1] = SBox(FF) = 16$   
 $B'[2] = SBox(FF) = 16$   
 $B'[3] = SBox(BF) = 08$
  - 5) Hitung  $b[1]$   
 $b[1] = RC1 \text{ XOR } B$   
 $b[1] = 01\ 00\ 00\ 00 \text{ XOR } 16$   
 $16\ 16\ 08$   
 $b[1] = 17\ 16\ 16\ 08$   
 $6) \text{ Hitung } b[2]$   
Misakan  $random\_bit = A0$   
 $DA\ C9\ 11$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 17\ 16\ 16\ 08 \text{ AND }$   
 $A0\ DA\ C9\ 11$   
 $b[2] = 00\ 12\ 00\ 00$
  - 7) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 00\ 12\ 00\ 00 \text{ XOR }$   
 $A0\ DA\ C9\ 11$   
 $b[3] = A0\ C8\ C9\ 11$
  - 8) Hitung  $b[4]$

Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) Input  $W' = FF\ ED\ FF\ FF$
- 2)  $w[40] = AF\ FD\ F9\ EF$   
 $w[41] = BF\ FD\ FF\ EF$   
 $w[42] = 7F\ FF\ FD\ FF$   
 $w[43] = FF\ 7F\ FD\ FF$   
 $w[44] = 7F\ EF\ FF\ FF$   
 $w[45] = F7\ FF\ FF\ FF$   
 $w[46] = 7F\ FF\ FF\ FF$   
 $w[47] = BF\ FF\ FF\ FF$
- 3) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[40]$   
 $b[1] = FF\ ED\ FF\ FF\ XOR$   
 $AF\ FD\ F9\ EF$   
 $b[1] = 50\ 10\ 06\ 10$
- 4) Hitung  $b[2]$   
Misalkan  $random\_bit = F891AC9D$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 50\ 10\ 06\ 10 \text{ AND } F8$   
 $91\ AC\ 9D$   
 $b[2] = 50\ 10\ 04\ 10$
- 5) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 50\ 10\ 04\ 10 \text{ XOR } F8$   
 $91\ AC\ 9D$   
 $b[3] = A8\ 81\ A8\ 8D$
- 6) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 57\ 7E\ 57\ 72$
- 7) Hitung  $w[48]$   
 $w[48] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[48] = 57\ 7E\ 57\ 72 \text{ XOR }$   
 $F8\ 91\ AC\ 9D$
- 8) Output  $w[48]$   
 $w[48] = AF\ EF\ FB\ EF$
- 9) Hitung  $b[1]$   
 $b[1] = w[48] \text{ XOR } w[41]$   
 $b[1] = AF\ EF\ FB\ EF\ XOR$   
 $BF\ FD\ FF\ EF$   
 $b[1] = 10\ 12\ 04\ 00$
- 10) Hitung  $b[2]$   
Misalkan  $random\_bit = 99$   
 $99\ AA\ AA$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 10\ 12\ 04\ 00 \text{ AND } 99$   
 $99\ AA\ AA$   
 $b[2] = 10\ 10\ 00\ 00$
- 11) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 10\ 10\ 00\ 00 \text{ XOR } 99$   
 $99\ AA\ AA$   
 $b[3] = 89\ 89\ AA\ AA$
- 12) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 76\ 76\ 55\ 55$
- 13) Hitung  $w[49]$   
 $w[49] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[49] = 76\ 76\ 55\ 55 \text{ XOR }$   
 $99\ 99\ AA\ AA$   
 $w[49] = EF\ EF\ FF\ FF$
- 14) Output  $w[49]$   
 $w[49] = EF\ EF\ FF\ FF$
- 15) Hitung  $b[1]$



- 21) Hitung  $b[1]$   
 $b[1] = w[50] \text{ XOR } w[43]$   
 $b[1] = 6F\ EF\ FD\ FF\ XOR$   
 $FF\ 7F\ FF\ DF$   
 $b[1] = 90\ 90\ 02\ 20$   
 22) Hitung  $b[2]$   
 Misalkan  $random\_bit = EE$   
 $EE\ 88\ 88$   
 23) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 80\ 80\ 00\ 00 \text{ XOR }$   
 $EE\ EE\ 88\ 88$   
 $b[3] = 6E\ 6E\ 88\ 88$   
 24) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 91\ 91\ 77\ 77$   
 25) Hitung  $w[51]$   
 $w[51] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[51] = 91\ 91\ 77\ 77 \text{ XOR }$   
 $EE\ EE\ 88\ 88$   
 $w[51] = 7F\ 7F\ FF\ FF$   
 26) Output  $w[51]$   
 $w[51] = 7F\ 7F\ FF\ FF$   
 27) Hitung  $b[1]$   
 $b[1] = w[51] \text{ XOR } w[44]$   
 $b[1] = 7F\ 7F\ FF\ FF\ XOR$   
 $7F\ EF\ FF\ FF$   
 $b[1] = 00\ 90\ 00\ 00$   
 28) Hitung  $b[2]$   
 Misalkan  $random\_bit =$   
 $DD\ DD\ 77\ 77$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 00\ 90\ 00\ 00 \text{ AND }$   
 $DD\ DD\ 77\ 77$   
 $b[2] = 00\ 90\ 00\ 00$   
 29) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$



- 33) Hitung b[1]  
 $b[1] = w[52] \text{ XOR } w[46]$   
 $b[1] = F7\ 7F\ FF\ FF\ XOR$   
 $7F\ FF\ FF\ FF$   
 $b[1] = 88\ 80\ 00\ 00$
- 34) Hitung b[2]  
 Misalkan  $random\_bit =$   
 $BBBB5555$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 88\ 80\ 00\ 00 \text{ AND }$   
 $BB\ BB\ 55\ 55$   
 $b[2] = 88\ 80\ 00\ 00$
- 35) Hitung b[3]  
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 88\ 80\ 00\ 00 \text{ XOR }$   
 $BB\ BB\ 55\ 55$   
 $b[3] = 33\ 3B\ 55\ 55$
- 36) Hitung b[4]  
 $b[4] = NOT b[3]$   
 $b[4] = CC\ C4\ AA\ AA$
- 37) Hitung w[54]  
 $w[54] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[54] = CC\ C4\ AA\ AA$   
 $XOR\ BB\ BB\ 55\ 55$   
 $w[54] = 77\ 7F\ FF\ FF$
- 38) Output w[53]  
 $w[53] = F7\ 7F\ FF\ FF$
- 39) Hitung b[1]  
 $b[1] = w[53] \text{ XOR } w[46]$   
 $b[1] = F7\ 7F\ FF\ FF\ XOR$   
 $7F\ FF\ FF\ FF$   
 $b[1] = 88\ 80\ 00\ 00$
- 40) Hitung b[2]  
 Misalkan  $random\_bit =$   
 $BBBB5555$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 88\ 80\ 00\ 00 \text{ AND }$   
 $BB\ BB\ 55\ 55$   
 $b[2] = 88\ 80\ 00\ 00$
- 41) Hitung b[3]  
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 88\ 80\ 00\ 00 \text{ XOR }$   
 $BB\ BB\ 55\ 55$   
 $b[3] = 33\ 3B\ 55\ 55$
- 42) Hitung b[4]  
 $b[4] = NOT b[3]$   
 $b[4] = CC\ C4\ AA\ AA$
- 43) Hitung w[54]  
 $w[54] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[54] = CC\ C4\ AA\ AA$   
 $XOR\ BB\ BB\ 55\ 55$   
 $w[54] = 77\ 7F\ FF\ FF$
- 44) Output w[54]  
 $w[54] = 77\ 7F\ FF\ FF$

- 45) Hitung b[1]
- $$b[1] = w[53] \text{ XOR } w[46]$$
- $$b[1] = F7 7F FF FF \text{ XOR}$$
- $$7F FF FF FF$$
- $$b[1] = 88 80 00 00$$
- 46) Hitung b[2]
- Misalkan  $random\_bit = BBBB5555$
- $$b[2] = b[1] \text{ AND }$$
- $$random\_bit$$
- $$b[2] = 88 80 00 00 \text{ AND }$$
- $$BB BB 55 55$$
- $$b[2] = 88 80 00 00$$
- 47) Hitung b[3]
- $$b[3] = b[2] \text{ XOR }$$
- $$random\_bit$$
- $$b[3] = 88 80 00 00 \text{ XOR }$$
- $$BB BB 55 55$$
- $$b[3] = 33 3B 55 55$$
- 48) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = CC C4 AA AA$$
- 49) Hitung w[54]
- $$w[54] = b[4] \text{ XOR }$$
- $$random\_bit$$
- $$w[54] = CC C4 AA AA$$
- $$\text{XOR } BB BB 55 55$$
- $$w[54] = 77 7F FF FF$$
- 50) Output w[54]
- $$w[54] = 77 7F FF FF$$
- 51) Hitung b[1]
- $$b[1] = w[54] \text{ XOR } w[47]$$
- $$b[1] = 77 7F FF FF \text{ XOR}$$
- $$BF FF FF FF$$
- $$b[1] = C8 80 00 00$$
- 52) Hitung b[2]
- Misalkan  $random\_bit = 5A$
- $$BA 18 DC$$
- 53) Hitung b[3]
- $$b[3] = b[2] \text{ XOR }$$
- $$random\_bit$$
- $$b[3] = 48 80 00 00 \text{ XOR }$$
- $$5A BA 18 DC$$
- $$b[3] = 12 3A 18 DC$$
- 54) Hitung b[4]
- $$b[4] = \text{NOT } b[3]$$
- $$b[4] = ED C5 E7 23$$
- 55) Hitung w[55]
- $$w[55] = b[4] \text{ XOR }$$
- $$random\_bit$$
- $$w[55] = ED C5 E7 23 \text{ XOR }$$
- $$5A BA 18 DC$$
- $$w[55] = B7 7F FF FF$$
- 56) Output w[55]
- $$w[55] = B7 7F FF FF$$

# UNIVERSITAS MIKROSKIL

Langkah kerja dari fungsi  $g$  dapat dirincikan sebagai berikut:

- Nilai  $W = w[55] = B7\ 7F\ FF\ FF$
- 1) Pecahkan  $W$  menjadi 4 buah blok  $B$ :
    - $B[0] = B7$
    - $B[1] = 7F$
    - $B[2] = FF$
    - $B[3] = FF$  - 2) Acak nilai  $B$  menjadi:
    - $B[0] = 7F$
    - $B[1] = FF$
    - $B[2] = FF$
    - $B[3] = B7$  - 3) Eksekusi fungsi S-Box.
    - $B'[0] = \text{SBox}(7F) = D2$
    - $B'[1] = \text{SBox}(FF) = 16$
    - $B'[2] = \text{SBox}(FF) = 16$
    - $B'[3] = \text{SBox}(B7) = A9$  - 4) Hitung  $b[1]$
    - $b[1] = RC1 \text{ XOR } B$
    - $b[1] = 01\ 00\ 00\ 00 \text{ XOR }$
    - $D2\ 16\ 16\ A9$
    - $b[1] = D3\ 16\ 16\ A9$  - 5) Hitung  $b[2]$   
Misakan  $\text{random\_bit} = 77$
    - $b[2] = b[1] \text{ AND }$
    - $\text{random\_bit}$
    - $b[2] = D3\ 16\ 16\ A9 \text{ AND }$
    - $77\ 77\ 77\ 77$
    - $b[2] = 53\ 16\ 16\ 21$  - 6) Hitung  $b[3]$
    - $b[3] = b[2] \text{ XOR }$
    - $\text{random\_bit}$
    - $b[3] = 53\ 16\ 16\ 21 \text{ XOR } 77$
    - $77\ 77\ 77$
    - $b[3] = 24\ 61\ 61\ 56$  - 7) Hitung  $b[4]$
    - $b[4] = \text{NOT } b[3]$

$$b[4] = DB\ 9E\ 9E\ A9$$

$$W' = b[4] \text{ XOR }$$

$$random\_bit$$

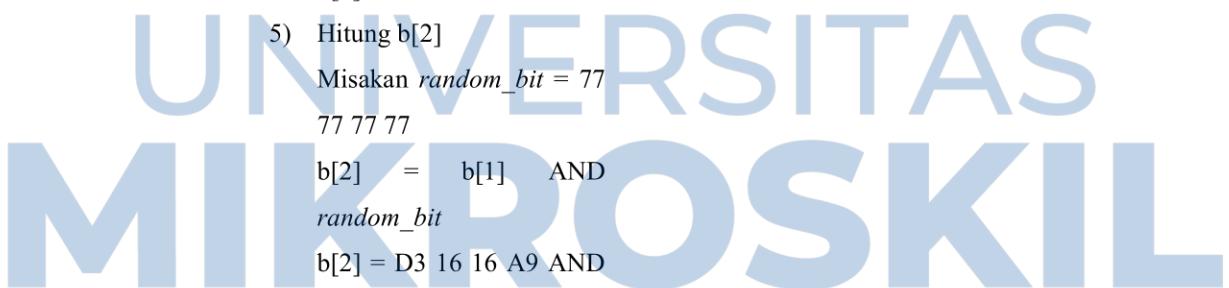
$$W' = DB\ 9E\ 9E\ A9 \text{ XOR }$$

$$77\ 77\ 77\ 77$$

$$W' = AC\ E9\ E9\ DE$$

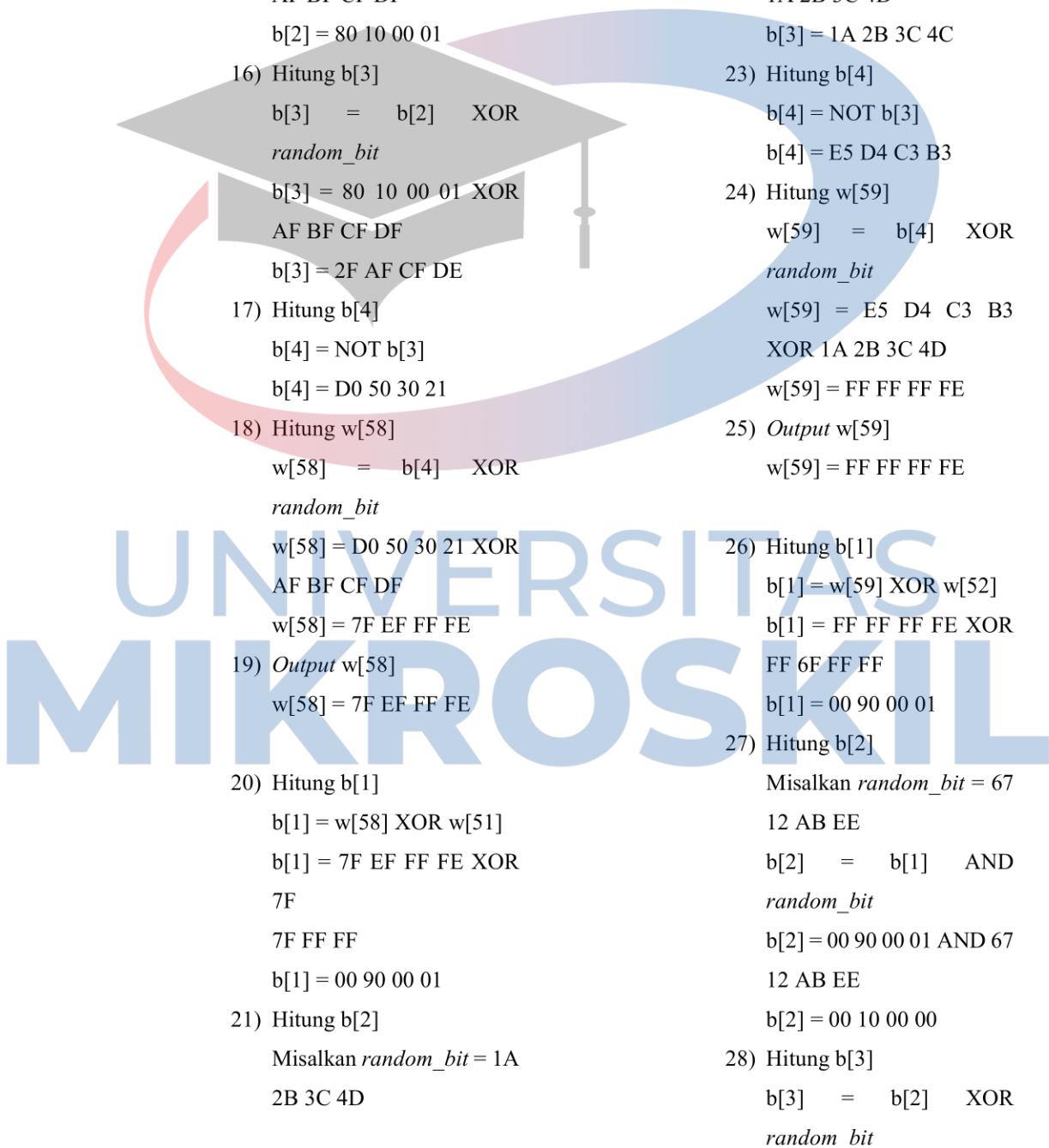
$$9) Output W'$$

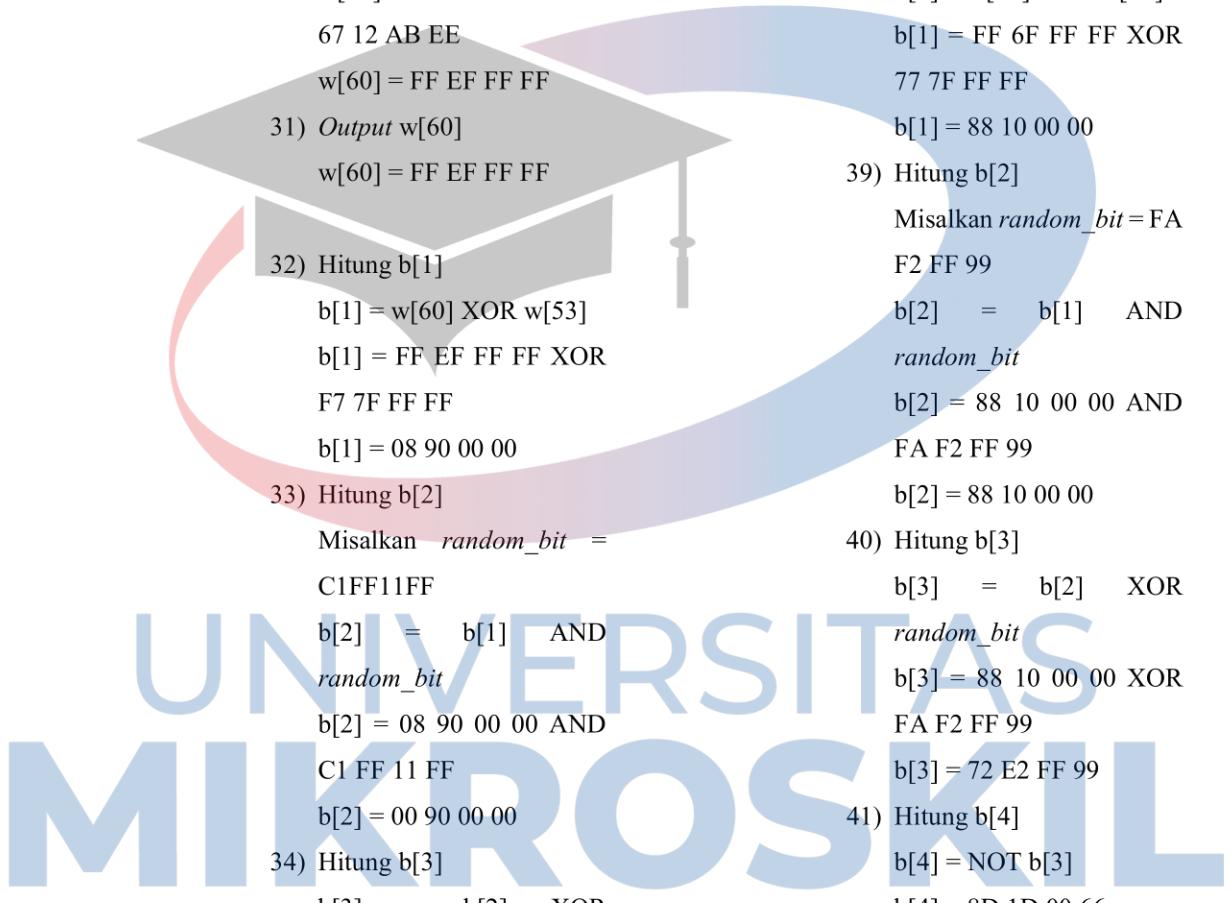
$$W' = AC\ E9\ E9\ DE$$



Langkah kerja dari fungsi SRFG tanpa penambahan RCi dapat dijabarkan sebagai berikut:

- 1) *Input W' = AC E9 E9 DE*  
 $w[48] = AF EF FB EF$   
 $w[49] = EF EF FF FF$   
 $w[50] = 6F EF FD FF$   
 $w[51] = 7F 7F FF FF$   
 $w[52] = FF 6F FF FF$   
 $w[53] = F7 7F FF FF$   
 $w[54] = 77 7F FF FF$   
 $w[55] = B7 7F FF FF$
- 2) Hitung  $b[1]$   
 $b[1] = W' \text{ XOR } w[48]$   
 $b[1] = AC E9 E9 DE \text{ XOR }$   
 $AF EF FB EF$   
 $b[1] = 03 06 12 31$
- 3) Hitung  $b[2]$   
Misalkan  $random\_bit = 11111111$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 03 06 12 31 \text{ AND } 11$   
 $11 11 11$   
 $b[2] = 01 00 10 11$
- 4) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 01 00 10 11 \text{ XOR } 11$   
 $11 11 11$   
 $b[3] = 10 11 01 00$
- 5) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = EF EE FE FF$
- 6) Hitung  $w[56]$   
 $w[56] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[56] = EF EE FE FF$   
 $\text{XOR } 11 11 11 11$   
 $w[56] = FE FF EF EE$
- 7) *Output w[56]*  
 $w[56] = FE FF EF EE$
- 8) Hitung  $b[1]$   
 $b[1] = w[56] \text{ XOR } w[49]$   
 $b[1] = FE FF EF EE \text{ XOR }$   
 $EF EF FF FF$   
 $b[1] = 11 10 10 11$
- 9) Hitung  $b[2]$   
Misalkan  $random\_bit = CACBCCCD$   
 $b[2] = b[1] \text{ AND }$   
 $random\_bit$   
 $b[2] = 11 10 10 11 \text{ AND }$   
 $CA CB CC CD$   
 $b[2] = 00 00 00 01$
- 10) Hitung  $b[3]$   
 $b[3] = b[2] \text{ XOR }$   
 $random\_bit$   
 $b[3] = 00 00 00 01 \text{ XOR }$   
 $CA CB CC CD$   
 $b[3] = CA CB CC CC$
- 11) Hitung  $b[4]$   
 $b[4] = \text{NOT } b[3]$   
 $b[4] = 35 34 33 33$
- 12) Hitung  $w[57]$   
 $w[57] = b[4] \text{ XOR }$   
 $random\_bit$   
 $w[57] = 35 34 33 33 \text{ XOR }$   
 $CA CB CC CD$   
 $w[57] = FF FF FF FE$
- 13) *Output w[57]*  
 $w[57] = FF FF FF FE$
- 14) Hitung  $b[1]$   
 $b[1] = w[57] \text{ XOR } w[50]$





w[61] = b[4] XOR

*random\_bit*

w[61] = 3E 90 EE 00 XOR

C1 FF 11 FF

w[61] = FF 6F FF FF

37) Output w[61]

w[61] = FF 6F FF FF

38) Hitung b[1]

b[1] = w[61] XOR w[54]

b[1] = FF 6F FF FF XOR

77 7F FF FF

b[1] = 88 10 00 00

39) Hitung b[2]

Misalkan *random\_bit* = FA

F2 FF 99

b[2] = b[1] AND  
*random\_bit*

b[2] = 88 10 00 00 AND

FA F2 FF 99

b[2] = 88 10 00 00

40) Hitung b[3]

b[3] = b[2] XOR

*random\_bit*

b[3] = 88 10 00 00 XOR

FA F2 FF 99

b[3] = 72 E2 FF 99

41) Hitung b[4]

b[4] = NOT b[3]

b[4] = 8D 1D 00 66

42) Hitung w[62]

w[62] = b[4] XOR

*random\_bit*

w[62] = 8D 1D 00 66 XOR

FA F2 FF 99

w[62] = 77 EF FF FF

43) Output w[62]

w[62] = 77 EF FF FF

44) Hitung b[1]

$$b[1] = w[62] \text{ XOR } w[55]$$

$$b[1] = 77 \text{ EF FF FF XOR}$$

$$\text{B7 7F FF FF}$$

$$b[1] = C0 90 00 00$$

45) Hitung b[2]

$$\text{Misalkan } random\_bit = 11$$

$$\text{FF 22 EE}$$

$$b[2] = b[1] \text{ AND }$$

$$random\_bit$$

$$b[2] = C0 90 00 00 \text{ AND }$$

$$11 \text{ FF 22 EE}$$

$$b[2] = 00 90 00 00$$

46) Hitung b[3]

$$b[3] = b[2] \text{ XOR }$$

$$random\_bit$$

$$b[3] = 00 90 00 00 \text{ XOR } 11$$

$$\text{FF 22 EE}$$

$$b[3] = 11 6F 22 EE$$

47) Hitung b[4]

$$b[4] = \text{NOT } b[3]$$

$$b[4] = EE 90 DD 11$$

48) Hitung w[63]

$$w[63] = b[4] \text{ XOR }$$

$$random\_bit$$

$$w[63] = EE 90 DD 11$$

$$\text{XOR } 11 \text{ FF 22 EE}$$

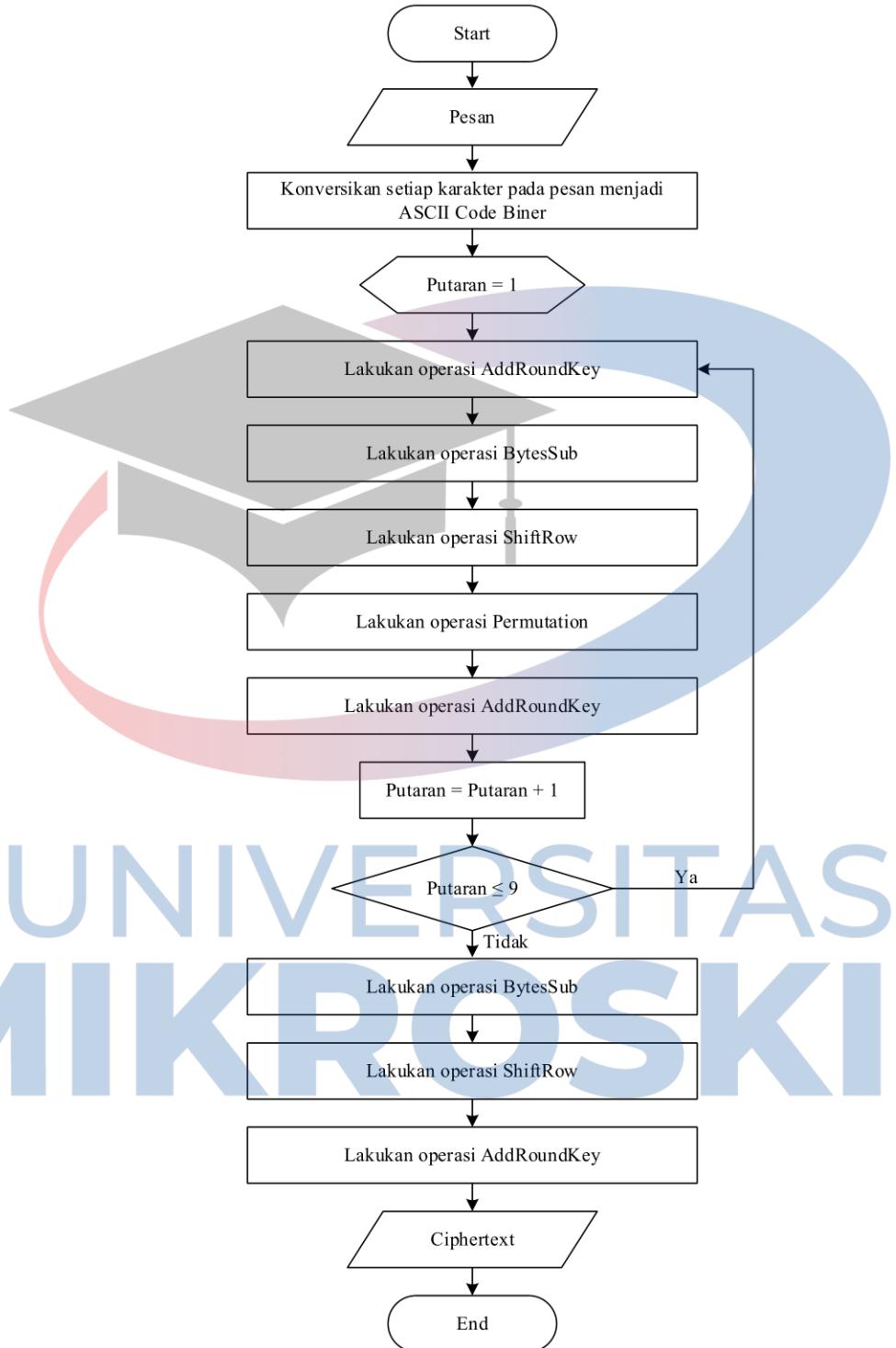
$$w[63] = FF 6F FF FF$$

49) Output w[63] = FF 6F FF

$$\text{FF}$$



3. Enkripsi pesan rahasia dengan menggunakan metode MAES. Proses enkripsi pesan rahasia dengan menggunakan metode MAES dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut:



Gambar 3.6 Flowchart Proses Enkripsi Pesan Rahasia dengan Metode MAES

Langkah kerja dari proses ekripsi pesan rahasia dengan metode MAES adalah sebagai berikut:

a. Pesan *input*:

01010000 01100001 01110011 01110011 01110111 01101111 01110010 01100100  
00100000 00111101 00100000 01100001 01100010 01100011 00110001 00110010

b. Ubah ke heksadesimal.

50 61 73 73 77 6F 72

64 20 3D 20 61 62 63

31 32

c. Putaran = 1.

1) Lakukan operasi AddRoundKey , Masukkan *plaintext* ke dalam *state*:

*State*

50 61 73 73

77 6F 72 64

20 3D 20 61

62 63 31 32

2) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(0) yaitu:

State di-XOR-kan dengan Subkunci(0)

State	$\oplus$	Subkunci(0)	=	AddRoundkey
50 61 73 73		B5 EF B7 EF		E5 8E C4 9C
77 6F 72 64		DE DF 9F FB		A9 B0 ED 9F
20 3D 20 61		F5 EE A9 EB		D5 D3 89 8A
62 63 31 32		5F DF DD D0		3D BC EC E2

3) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
E5 8E C4 9C		D9 19 1C DE
A9 B0 ED 9F		D3 E7 55 DB
D5 D3 89 8A		03 66 A7 7E
3D BC EC E2		27 65 CE 98

Substitusi dilakukan dengan cara:

E5 => S-Box (Baris E, Kolom 5) = D9

8E => S-Box (Baris 8, Kolom E) = 19

C4 => S-Box (Baris C, Kolom 4) = 1C

9C => S-Box (Baris 9, Kolom C) = DE

A9 => S-Box (Baris A, Kolom 9) = D3

B0 => S-Box (Baris B, Kolom 0) = E7

ED => S-Box (Baris E, Kolom D) = 55

9F => S-Box (Baris 9, Kolom F) = DB

D5 => S-Box (Baris D, Kolom 5) = 03

D3 => S-Box (Baris D, Kolom 3) = 66  
89 => S-Box (Baris 8, Kolom 9) = A7  
8A => S-Box (Baris 8, Kolom A) = 7E  
3D => S-Box (Baris 3, Kolom D) = 27  
BC => S-Box (Baris B, Kolom C) = 65  
EC => S-Box (Baris E, Kolom C) = CE  
E2 => S-Box (Baris E, Kolom 2) = 98

- 4) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

=> Baris pertama tetap  
=> Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte  
=> Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte  
=> Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte

D9 19 1C DE  
D3 E7 55 DB  
03 66 A7 7E  
27 65 CE 98

Hasil ShiftRow

D9 19 1C DE  
E7 55 DB D3  
A7 7E 03 66  
98 27 65 CE

- 5) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

D9 19 1C DE  
E7 55 DB D3  
A7 7E 03 66

Hasil ShiftRow (biner)

1101 1001 0001 1001 0001 1100 1101 1110  
1110 0111 0101 0101 1101 1011 1101 0011  
1010 0111 0111 1110 0000 0011 0110 0110  
1001 1000 0010 0111 0110 0101 1100 1110

Hasil Permutation:

1111 1001 1110 1111 0011 1100 1111 0011  
1101 1001 0001 0000 0100 1111 1101 1000  
1100 1010 0001 0010 1110 1011 0110 0101  
1001 0001 0110 1011 1001 0010 1010 1111

Hasil Permutation (heksa)

F9 EF 3C F3  
D9 10 4F D8  
CA 12 EB 65  
91 6B 92 AF

- d. Putaran = 2.
- 6) Lakukan operasi AddRoundKey , Masukkan hasil *Permutation* ke dalam *state*:
- State*
- |             |
|-------------|
| F9 EF 3C F3 |
| D9 10 4F D8 |
| CA 12 EB 65 |
| 91 6B 92 AF |

- 7) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(1) yaitu:

State di-XOR-kan dengan Subkunci(1)

State	$\oplus$	Subkunci(1)	=	AddRoundkey
F9 EF 3C F3		ED FD EF FD		14 12 D3 0E
D9 10 4F D8		7B DB DA DE		A2 CB 95 06
CA 12 EB 65		D8 F7 F6 7F		12 E5 1D 1A
91 6B 92 AF		1D 7D BF AE		8C 16 2D 01

- 8) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
14 12 D3 0E		FA C9 66 AB
A2 CB 95 06		3A 1F 2A 6F
12 35 1D 1A		C9 96 A4 A2
8C 16 2D 01		54 47 D8 7C

Substitusi dilakukan dengan cara:

14 => S-Box (Baris 1, Kolom 4) = FA

12 => S-Box (Baris 1, Kolom 2) = C9

D3 => S-Box (Baris D, Kolom 3) = 66

0E => S-Box (Baris 0, Kolom E) = AB

A2 => S-Box (Baris A, Kolom 2) = 3A

CB => S-Box (Baris C, Kolom B) = 1F

95 => S-Box (Baris 9, Kolom 5) = 2A

06 => S-Box (Baris 0, Kolom 6) = 6F

12 => S-Box (Baris 1, Kolom 2) = C9

35 => S-Box (Baris 3, Kolom 5) = 96

1D => S-Box (Baris 1, Kolom D) = A4

1A => S-Box (Baris 1, Kolom A) = A2

8C => S-Box (Baris 8, Kolom C) = 54

16 => S-Box (Baris 1, Kolom 6) = 47

2D => S-Box (Baris 2, Kolom D) = D8

01 => S-Box (Baris 0, Kolom 1) = 7C

- 9) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte	Hasil ShiftRow
FA C9 66 AB	FA C9 66 AB
3A 1F 2A 6F	1F 2A 6F 3A
C9 96 A4 A2	A4 A2 C9 96
54 47 D8 7C	7C 54 47 D8

10) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)  
FA C9 66 AB  
1F 2A 6F 3A  
A4 A2 C9 96  
7C 54 47 D8  
Hasil ShiftRow (biner)  
1111 1010 1100 1001 0110 0110 1010 1011  
0001 1111 0010 1010 0110 1111 0011 1010  
1010 0100 1010 0010 1100 1001 1001 0110  
0111 1100 0101 0100 0100 0111 1101 1000

Hasil Permutation:  
0100 0111 1001 0001 0101 0100 0101 1010  
0000 1011 1110 1101 1111 1011 1111 1101  
1111 0100 1011 1000 0111 1001 0100 0100  
1000 1111 0001 0011 1001 0100 0100 1010

Hasil Permutation (heksa)  
47 91 54 5A  
0B ED FB FD  
F4 B8 79 44  
8F 13 94 4A

e. Putaran = 3.

11) Lakukan operasi AddRoundKey , Masukkan hasil Permutation ke dalam state:

*State*

47 91 54 5A  
0B ED FB FD  
F4 B8 79 44  
8F 13 94 4A

12) Lakukan transformasi AddRoundkey pada State dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(2)

State	$\oplus$	Subkunci(2)	=	AddRoundkey
47 91 54 5A		F3 EF F7 EB		B4 7E A3 B1
0B ED FB FD		D3 DF D7 EF		D8 32 2C 12
F4 B8 79 44		FB EF FF FB		0F 57 96 BF
8F 13 94 4A		5F FF FF FE		D0 EC 6B B4

13) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
B4 7E A3 B1		8D F3 0A C8
D8 32 2C 12		61 23 71 C9
0F 57 96 BF		76 5B 90 08
D0 EC 6B B4		70 CE 7F 8D

Substitusi dilakukan dengan cara:

- B4 => S-Box (Baris B, Kolom 4) = 8D
- 7E => S-Box (Baris 7, Kolom E) = F3
- A3 => S-Box (Baris A, Kolom 3) = 0A
- B1 => S-Box (Baris B, Kolom 1) = C8
- D8 => S-Box (Baris D, Kolom 8) = 61
- 32 => S-Box (Baris 3, Kolom 2) = 23
- 2C => S-Box (Baris 2, Kolom C) = 71
- 12 => S-Box (Baris 1, Kolom 2) = C9
- 0F => S-Box (Baris 0, Kolom F) = 76
- 57 => S-Box (Baris 5, Kolom 7) = 5B
- 96 => S-Box (Baris 9, Kolom 6) = 90
- BF => S-Box (Baris B, Kolom F) = 08
- D0 => S-Box (Baris D, Kolom 0) = 70
- EC=> S-Box (Baris E, Kolom C) = CE
- 6B => S-Box (Baris 6, Kolom B) = 7F
- B4 => S-Box (Baris B, Kolom 4) = 8D

14) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte	Hasil ShiftRow
8D F3 0A C8	8D F3 0A C8
61 23 71 C9	23 71 C9 61
76 5B 90 08	90 08 76 5B
70 CE 7F 8D	8D 70 CE 7F

15) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

8D F3 0A C8

23 71 C9 61

90 08 76 5B

8D 70 CE 7F

Hasil ShiftRow (biner)

1000 1101 1111 0011 0000 1010 1100 1000

0010 0011 0111 0001 1100 1001 0110 0001

1001 0000 0000 1000 0111 0110 0101 1011

1000 1101 0111 0000 1100 1110 0111 1111

Hasil Permutation:

1110 1010 0010 0010 0000 0001 1111 0011

0100 1011 1011 0010 0100 1101 0001 0110

1110 1100 1010 1101 1101 0100 1001 1000

0101 0001 1010 0100 1101 1010 1100 1100

Hasil Permutation (heksa)

EA 22 01 F3

4B B2 4D 16

EC AD D4 98

51 A4 DA CC

f. Putaran = 4.

16) Lakukan operasi AddRoundKey , Masukkan hasil Permutation ke dalam state:

State

EA 22 01 F3

4B B2 4D 16

EC AD D4 98

51 A4 DA CC

17) Lakukan transformasi AddRoundkey pada State dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(3)

State	$\oplus$	Subkunci(3)	=	AddRoundkey
EA 22 01 F3		FF FD FF FE		15 DF FE 0D
4B B2 4D 16		FB DF FE DF		B0 6D B3 C9
EC AD D4 98		FF DF F7 DF		13 72 23 47
51 A4 DA CC		3D 7D F7 A7		6C D9 2D 6B

18) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey  $\rightarrow$  Hasil SubByte

15 DF FE 0D  $\rightarrow$  59 9E BB D7

B0 6D B3 C9  $\rightarrow$  E7 3C 6D DD

13 72 23 47	7D 40 26 A0
6C D9 2D 6B	50 35 D8 7F

Substitusi dilakukan dengan cara:

15 => S-Box (Baris 1, Kolom 5) = 59  
DF => S-Box (Baris D, Kolom F) = 9E  
FE => S-Box (Baris F, Kolom E) = BB  
0D => S-Box (Baris 0, Kolom D) = D7  
B0 => S-Box (Baris B, Kolom 0) = E7  
6D => S-Box (Baris 6, Kolom D) = 3C  
B3 => S-Box (Baris B, Kolom 3) = 6D  
C9 => S-Box (Baris C, Kolom 9) = DD  
13 => S-Box (Baris 1, Kolom 3) = 7D  
72 => S-Box (Baris 7, Kolom 2) = 40  
23 => S-Box (Baris 2, Kolom 3) = 26  
47 => S-Box (Baris 4, Kolom 7) = A0  
6C => S-Box (Baris 6, Kolom C) = 50  
D9 => S-Box (Baris D, Kolom 9) = 35  
2D => S-Box (Baris 2, Kolom D) = D8  
6B => S-Box (Baris 6, Kolom B) = 7F

19) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

=> Baris pertama tetap  
=> Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte  
=> Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte  
=> Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte  
59 9E BB D7  
E7 3C 6D DD  
7D 40 26 A0  
50 35 D8 7F

Hasil ShiftRow  
59 9E BB D7  
3C 6D DD E7  
26 A0 7D 40  
7F 50 35 D8

20) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

59 9E BB D7  
3C 6D DD E7  
26 A0 7D 40  
7F 50 35 D8

Hasil ShiftRow (biner)

0101 1001 1001 1110 1011 1011 1101 0111  
0011 1100 0110 1101 1101 1101 1110 0111

0010 0110 1010 0000 0111 1101 0100 0000  
0111 1111 0101 0000 0011 0101 1101 1000

Hasil Permutation:

1110 1001 0101 1111 1111 1010 1110 1101  
1100 1110 1011 0100 0111 0111 1000 1110  
1011 1100 1111 0100 0101 0101 0101 0100  
1000 0010 0101 0111 1001 0100 0001 0001

Hasil Permutation (heksa)

E9 5F FA ED

CE B4 77 8E

BC F4 55 54

82 57 94 11

g. Putaran = 5.

21) Lakukan operasi AddRoundKey , Masukkan hasil *Permutation* ke dalam *state*:

State

E9 5F FA ED

CE B4 77 8E

BC F4 55 54

82 57 94 11

22) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(4)

State	$\oplus$	Subkunci(4)	=	AddRoundkey
E9 5F FA ED		E9 E7 FF EC		00 B8 05 01
CE B4 77 8E		FD FF FF FC		33 4B 88 72
BC F4 55 54		FB EF FF FD		47 1B AA A9
82 57 94 11		5F FF FF FD		DD A8 6B EC

23) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
00 B8 05 01		63 6C 6B 7C
33 4B 88 72		C3 B3 C4 40
47 1B AA A9		A0 AF AC D3
DD A8 6B EC		C1 C2 7F CE

Substitusi dilakukan dengan cara:

00 => S-Box (Baris 0, Kolom 0) = 63

B8 => S-Box (Baris B, Kolom 8) = 6C

05 => S-Box (Baris 0, Kolom 5) = 6B

01 => S-Box (Baris 0, Kolom 1) = 7C

33 => S-Box (Baris 3, Kolom 3) = C3

4B => S-Box (Baris 4, Kolom B) = B3

88 => S-Box (Baris 8, Kolom 8) = C4  
72 => S-Box (Baris 7, Kolom 2) = 40  
47 => S-Box (Baris 4, Kolom 7) = A0  
1B => S-Box (Baris 1, Kolom B) = AF  
AA => S-Box (Baris A, Kolom A) = AC  
A9 => S-Box (Baris A, Kolom 9) = D3  
DD => S-Box (Baris D, Kolom D) = C1  
A8 => S-Box (Baris A, Kolom 8) = C2  
6B => S-Box (Baris 6, Kolom B) = 7F  
EC => S-Box (Baris E, Kolom C) = CE

24) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte  
63 6C 6B 7C  
C3 B3 C4 40  
A0 AF AC D3  
C1 C2 7F CE

Hasil ShiftRow  
63 6C 6B 7C  
B3 C4 40 C3  
AC D3 A0 AF  
CE C1 C2 7F

25) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

63 6C 6B 7C  
B3 C4 40 C3  
AC D3 A0 AF  
CE C1 C2 7F

Hasil ShiftRow (biner)

0110 0011 0110 1100 0110 1011 0111 1100  
1011 0011 1100 0100 0100 0000 1100 0011  
1010 1100 1101 0011 1010 0000 1010 1111  
1100 1110 1100 0001 1100 0010 0111 1111

Hasil Permutation:

1110 1111 0001 1000 0010 1010 1001 0101  
1011 0000 0001 1111 0000 1110 1001 0101  
1111 0010 1000 0010 1001 1001 1010 1010  
0111 1111 1000 1101 1001 1001 1101 1010

Hasil Permutation (heksa)

EF 18 2A 95

B0 1F 0E 95  
 F2 82 99 AA  
 7F 8D 99 DA

h. Putaran = 6.

26) Lakukan operasi AddRoundKey , Masukkan hasil *Permutation* ke dalam *state*:

*State*

EF 18 2A 95  
 B0 1F 0E 95  
 F2 82 99 AA  
 7F 8D 99 DA

27) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(5)

State	$\oplus$	Subkunci(5)	=	AddRoundkey
EF 18 2A 95		FF FF FF FD		10 E7 D5 68
B0 1F 0E 95		FB DF FF FD		4B C0 F1 68
F2 82 99 AA		FB FF FF FF		09 7D 66 55
7F 8D 99 DA		31 78 77 23		4E F5 EE F9

28) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
10 E7 D5 68		CA 94 03 45
4B C0 F1 68		B3 BA A1 45
09 7D 66 55		01 FF 33 FC
4E F5 EE F9		2F E6 28 99

Substitusi dilakukan dengan cara:

10 => S-Box (Baris 1, Kolom 0) = CA

E7 => S-Box (Baris E, Kolom 7) = 94

D5 => S-Box (Baris D, Kolom 5) = 03

68 => S-Box (Baris 6, Kolom 8) = 45

4B => S-Box (Baris 4, Kolom B) = B3

C0 => S-Box (Baris C, Kolom 0) = BA

F1 => S-Box (Baris F, Kolom 1) = A1

68 => S-Box (Baris 6, Kolom 8) = 45

09 => S-Box (Baris 0, Kolom 9) = 01

7D => S-Box (Baris 7, Kolom D) = FF

66 => S-Box (Baris 6, Kolom 6) = 33

55 => S-Box (Baris 5, Kolom 5) = FC

4E => S-Box (Baris 4, Kolom E) = 2F

F5 => S-Box (Baris F, Kolom 5) = E6

EE => S-Box (Baris E, Kolom E) = 28

F9 => S-Box (Baris F, Kolom 9) = 99

29) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte

CA 94 03 45

B3 BA A1 45

01 FF 33 FC

2F E6 28 99

Hasil ShiftRow

CA 94 03 45

BA A1 45 B3

33 FC 01 FF

99 2F E6 28

30) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

CA 94 03 45

BA A1 45 B3

33 FC 01 FF

99 2F E6 28

Hasil ShiftRow (biner)

1100 1010 1001 0100 0000 0011 0100 0101

1011 1010 1010 0001 0100 0101 1011 0011

0011 0011 1111 1100 0000 0001 1111 1111

1001 1001 0010 1111 1110 0110 0010 1000

Hasil Permutation:

0100 1001 1001 0010 0100 1010 1110 1100

1011 0011 1011 0000 0001 0001 1001 0101

0100 1010 0001 1011 0110 1010 0011 1101

0101 1010 1110 1011 1011 1010 0110 1001

Hasil Permutation (heksa)

49 92 4A EC

B3 B0 11 95

4A 1B 6A 3D

5A EB BA 69

i. Putaran = 7.

31) Lakukan operasi AddRoundKey , Masukkan hasil Permutation ke dalam state:

State

49 92 4A EC

B3 B0 11 95

4A 1B 6A 3D

5A EB BA 69

32) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(6)

State	$\oplus$	Subkunci(6)	=	AddRoundkey
49 92 4A EC		72 76 D9 FE		3B E4 93 12
B3 B0 11 95		72 77 DD FF		C1 C7 CC 6A
4A 1B 6A 3D		77 67 FF FF		3D 7C 95 C2
5A EB BA 69		D7 EF FF DD		8D 04 45 B4

33) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
3B 34 93 12		E2 18 DC C9
C1 C7 CC 6A		78 C6 4B 02
3D 7C 95 C2		27 10 2A 25
8D 04 45 B4		5D F2 6E 8D

Substitusi dilakukan dengan cara:

3B => S-Box (Baris 3, Kolom B) = E2

34 => S-Box (Baris 3, Kolom 4) = 18

93 => S-Box (Baris 9, Kolom 3) = DC

12 => S-Box (Baris 1, Kolom 2) = C9

C1 => S-Box (Baris C, Kolom 1) = 78

C7 => S-Box (Baris C, Kolom 7) = C6

CC => S-Box (Baris C, Kolom C) = 4B

6A => S-Box (Baris 6, Kolom A) = 02

3D => S-Box (Baris 3, Kolom D) = 27

7C => S-Box (Baris 7, Kolom C) = 10

95 => S-Box (Baris 9, Kolom 5) = 2A

C2 => S-Box (Baris C, Kolom 2) = 25

8D => S-Box (Baris 8, Kolom D) = 5D

04 => S-Box (Baris 0, Kolom 4) = F2

45 => S-Box (Baris 4, Kolom 5) = 6E

B4 => S-Box (Baris B, Kolom 4) = 8D

34) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte	Hasil ShiftRow
E2 18 DC C9	E2 18 DC C9
78 C6 4B 02	C6 4B 02 78
27 10 2A 25	2A 25 27 10
5D F2 6E 8D	8D 5D F2 6E

35) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

E2 18 DC C9

C6 4B 02 78

2A 25 27 10

8D 5D F2 6E

Hasil ShiftRow (biner)

1110 0010 0001 1000 1101 1100 1100 1001

1100 0110 0100 1011 0000 0010 0111 1000

0010 1010 0010 0101 0010 0111 0001 0000

1000 1101 0101 1101 1111 0010 0110 1110

Hasil Permutation:

1011 1101 1000 0110 0001 0100 0010 1000

0001 1101 1000 0001 1010 1110 0111 0001

1110 0000 0110 1000 1011 0110 0011 0110

0101 0000 1100 0111 1011 0001 1100 0101

Hasil Permutation (heksa)

BD 86 14 28

1D 81 AE 71

E0 68 B6 36

50 C7 B1 C5

j. Putaran = 8.

36) Lakukan operasi AddRoundKey , Masukkan hasil Permutation ke dalam state:

State

BD 86 14 28

1D 81 AE 71

E0 68 B6 36

50 C7 B1 C5

37) Lakukan transformasi AddRoundkey pada State dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(7)

State  $\oplus$  Subkunci(7) = AddRoundkey

BD 86 14 28 FF EF FF FF 42 69 EB D7

1D 81 AE 71	FF EF FF FD	E2 6E 51 8C
E0 68 B6 36	FF EF FF FD	1F 87 49 CB
50 C7 B1 C5	B5 FB FF FF	E5 3C 4E 3A

38) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	→	Hasil SubByte
42 69 EB D7		2C F9 E9 0E
E2 6E 51 8C		98 9F D1 64
1F 87 49 CB		C0 17 3B 1F
E5 3C 4E 3A		D9 EB 2F 80

Substitusi dilakukan dengan cara:

42 => S-Box (Baris 4, Kolom 2) = 2C  
 69 => S-Box (Baris 6, Kolom 9) = F9  
 EB => S-Box (Baris E, Kolom B) = E9  
 D7 => S-Box (Baris D, Kolom 7) = 0E  
 E2 => S-Box (Baris E, Kolom 2) = 98  
 6E => S-Box (Baris 6, Kolom E) = 9F  
 51 => S-Box (Baris 5, Kolom 1) = D1  
 8C => S-Box (Baris 8, Kolom C) = 64  
 1F => S-Box (Baris 1, Kolom F) = C0  
 87 => S-Box (Baris 8, Kolom 7) = 17  
 49 => S-Box (Baris 4, Kolom 9) = 3B  
 CB => S-Box (Baris C, Kolom B) = 1F  
 E5 => S-Box (Baris E, Kolom 5) = D9  
 3C => S-Box (Baris 3, Kolom C) = EB  
 4E => S-Box (Baris 4, Kolom E) = 2F  
 3A => S-Box (Baris 3, Kolom A) = 80

39) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte	Hasil ShiftRow
2C F9 E9 0E	2C F9 E9 0E
98 9F D1 64	9F D1 64 98
C0 17 3B 1F	3B 1F C0 17
D9 EB 2F 80	80 D9 EB 2F

40) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

2C F9 E9 0E  
 9F D1 64 98  
 3B 1F C0 17  
 80 D9 EB 2F  
 Hasil ShiftRow (biner)  
 0010 1100 1111 1001 1110 1001 0000 1110  
 1001 1111 1101 0001 0110 0100 1001 1000  
 0011 1011 0001 1111 1100 0000 0001 0111  
 1000 0000 1101 1001 1110 1011 0010 1111

Hasil Permutation:

0110 0110 1011 0010 0101 1001 0011 0110  
 1011 0110 0100 0111 1001 1111 0001 1000  
 0110 0100 0010 1011 1000 1010 1110 1011  
 0111 0100 1100 0001 1110 0011 1100 1011

Hasil Permutation (heksa)

66 B2 59 36  
 B6 47 9F 18  
 64 2B 8A EB  
 74 C1 E3 CB

k. Putaran = 9.

41) Lakukan operasi AddRoundKey , Masukkan hasil *Permutation* ke dalam *state*:

*State*

66 B2 59 36  
 B6 47 9F 18  
 64 2B 8A EB  
 74 C1 E3 CB

42) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(2) yaitu:

State di-XOR-kan dengan Subkunci(8)

State	$\oplus$	Subkunci(8)	=	AddRoundkey
66 B2 59 36		7B EF FB FF		1D 5D A2 C9
B6 47 9F 18		F7 FF DD FF		41 B8 42 E7
64 2B 8A EB		7F 7F FD FF		1B 54 77 14
74 C1 E3 CB		57 7F FD DD		23 BE 1E 16

43) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
1D 5D A2 C9		A4 4C 3A DD
41 B8 42 E7		83 6C 2C 94
1B 54 77 14		AF 20 F5 FA
23 BE 1E 16		26 AE 72 47

Substitusi dilakukan dengan cara:

1D => S-Box (Baris 1, Kolom D) = A4  
5D => S-Box (Baris 5, Kolom D) = 4C  
A2 => S-Box (Baris A, Kolom 2) = 3A  
C9 => S-Box (Baris C, Kolom 9) = DD  
41 => S-Box (Baris 4, Kolom 1) = 83  
B8 => S-Box (Baris B, Kolom 8) = 6C  
42 => S-Box (Baris 4, Kolom 2) = 2C  
E7 => S-Box (Baris E, Kolom 7) = 94  
1B => S-Box (Baris 1, Kolom B) = AF  
54 => S-Box (Baris 5, Kolom 4) = 20  
77 => S-Box (Baris 7, Kolom 7) = F5  
14 => S-Box (Baris 1, Kolom 4) = FA  
23 => S-Box (Baris 2, Kolom 3) = 26  
BE => S-Box (Baris B, Kolom E) = AE  
1E => S-Box (Baris 1, Kolom E) = 72  
16 => S-Box (Baris 1, Kolom 6) = 47

44) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

=> Baris pertama tetap  
=> Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte  
=> Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte  
=> Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte

A4 4C 3A DD  
83 6C 2C 94  
AF 20 F5 FA  
26 AE 72 47

Hasil ShiftRow

A4 4C 3A DD  
6C 2C 94 83  
F5 FA AF 20  
47 26 AE 72

45) Lakukan operasi Permutation:

Hasil ShiftRow (heksa)

A4 4C 3A DD  
6C 2C 94 83  
F5 FA AF 20  
47 26 AE 72

Hasil ShiftRow (biner)

1010 0100 0100 1100 0011 1010 1101 1101  
0110 1100 0010 1100 1001 0100 1000 0011  
1111 0101 1111 1010 1010 1111 0010 0000  
0100 0111 0010 0110 1010 1110 0111 0010

Hasil Permutation:

0001 1010 0100 1100 0111 1011 1000 1000  
1100 1001 0011 0101 0011 1110 1000 0100  
1001 0011 1000 0011 0111 0101 0001 0101  
0100 0111 1110 1111 0100 0110 1111 0110

Hasil Permutation (heksa)

1A 4C 7B 88  
C9 35 3E 84  
93 83 75 15  
47 EF 46 F6

1. Putaran = 10.

46) Lakukan operasi AddRoundKey , Masukkan hasil *Permutation* ke dalam *state*:

State  
1A 4C 7B 88  
C9 35 3E 84  
93 83 75 15  
47 EF 46 F6

47) Lakukan transformasi AddRoundkey pada *State* dengan Subkunci(9) yaitu:

State di-XOR-kan dengan Subkunci(9)

State	$\oplus$	Subkunci(9)	=	AddRoundkey
1A 4C 7B 88		77 EF FF DD		6D A3 84 55
C9 35 3E 84		77 FF FF FF		BE CA C1 7B
93 83 75 15		77 EF FF FF		E4 6C 8A EA
47 EF 46 F6		3F EF FF FF		78 00 B9 09

48) Lakukan transformasi SubBytes disubstitusikan dengan S-Box:

Hasil AddRoundkey	$\rightarrow$	Hasil SubByte
6D A3 84 55		3C 0A 5F FC
BE CA C1 7B		E4 74 78 21
E4 6C 8A EA		69 50 7E 87
78 00 B9 09		BC 63 56 01

Substitusi dilakukan dengan cara:

6D => S-Box (Baris 6, Kolom D) = 3C

A3 => S-Box (Baris A, Kolom 3) = 0A

84 => S-Box (Baris 8, Kolom 4) = 5F

55 => S-Box (Baris 5, Kolom 5) = FC

BE => S-Box (Baris B, Kolom E) = E4

CA => S-Box (Baris C, Kolom A) = 74

C1 => S-Box (Baris C, Kolom 1) = 78

7B => S-Box (Baris 7, Kolom B) = 21  
 E4 => S-Box (Baris E, Kolom 4) = 69  
 6C => S-Box (Baris 6, Kolom C) = 50  
 8A => S-Box (Baris 8, Kolom A) = 7E  
 EA => S-Box (Baris E, Kolom A) = 87  
 78 => S-Box (Baris 7, Kolom 8) = BC  
 00 => S-Box (Baris 0, Kolom 0) = 63  
 B9 => S-Box (Baris B, Kolom 9) = 56  
 09 => S-Box (Baris 0, Kolom 9) = 01

49) Lakukan operasi transformasi ShiftRow yaitu:

Hasil SubByte dilakukan transformasi ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kiri sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kiri sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kiri sebanyak 3 byte

Hasil SubByte

3C 0A 5F FC
E4 74 78 21
69 50 7E 87
BC 63 56 01

Hasil ShiftRow

3C 0A 5F FC
74 78 21 E4
7E 87 69 50
01 BC 63 56

50) Lakukan operasi AddRoundKey , Masukkan hasil *ShiftRow* ke dalam *state*:

State	$\oplus$	Subkunci(10)	$\rightarrow$	Hasil AddRoundkey
3C 0A 5F FC		AF FD F9 EF		93 F7 A6 13
74 78 21 E4		BF FD FF EF		CB 85 DE 0B
7E 87 69 50		7F FF FD FF		01 78 94 AF
01 BC 63 56		FF 7F FF DF		FE C3 9C 89

#### 4. *Input* Citra Sampul

User memasukkan citra sampul yang akan digunakan sebagai tempat penyisipan pesan rahasia. Citra ini berupa citra berwarna. Misalkan citra *input* berukuran 100 x 100 berupa gambar kosong berwarna putih semua.

#### 5. Cek apakah pesan dapat ditampung dalam citra sampul.

Ukuran citra sampul :  $100 * 100 = 10.000$  piksel

Satu piksel dapat menyimpan 6 bit, jadi total bit yang dapat disimpan:

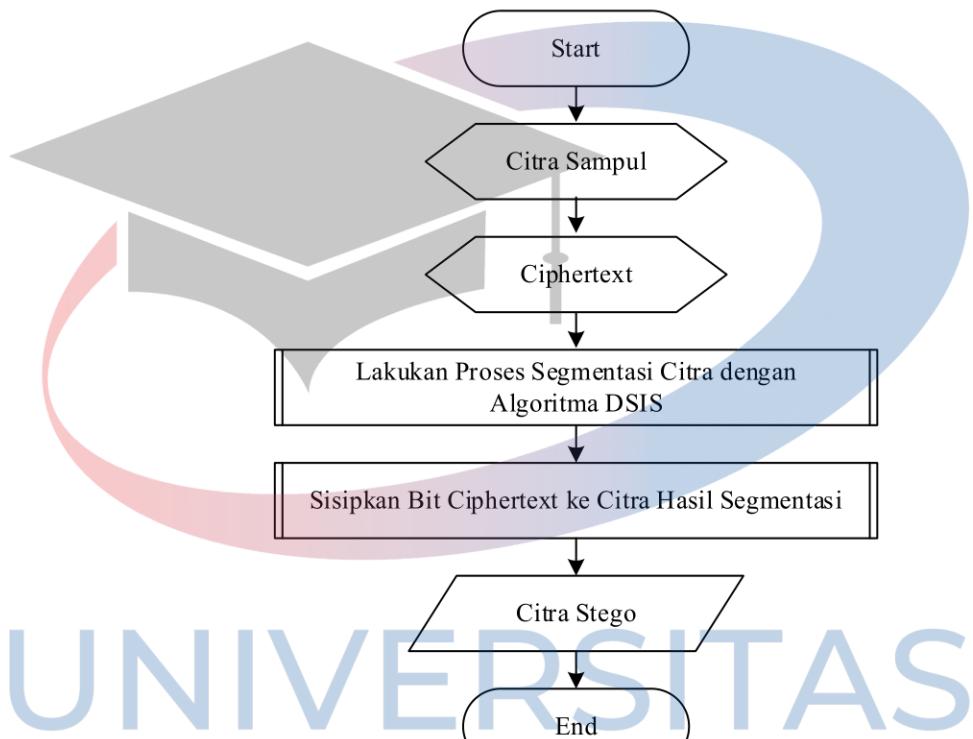
$$10.000 * 6 = 60.000 \text{ bit.}$$

Ukuran pesan : 16 karakter \* 8 bit = 128 bit.

Karena total bit yang dapat disimpan dalam citra sampul {60.000 bit} > ukuran bit pesan {128 bit}, maka proses dilanjutkan.

6. Proses penyisipan hasil enkripsi ke citra sampul dengan metode steganografi MLSB.

Langkah kerja dari proses penyisipan dengan metode steganografi MLSB dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:

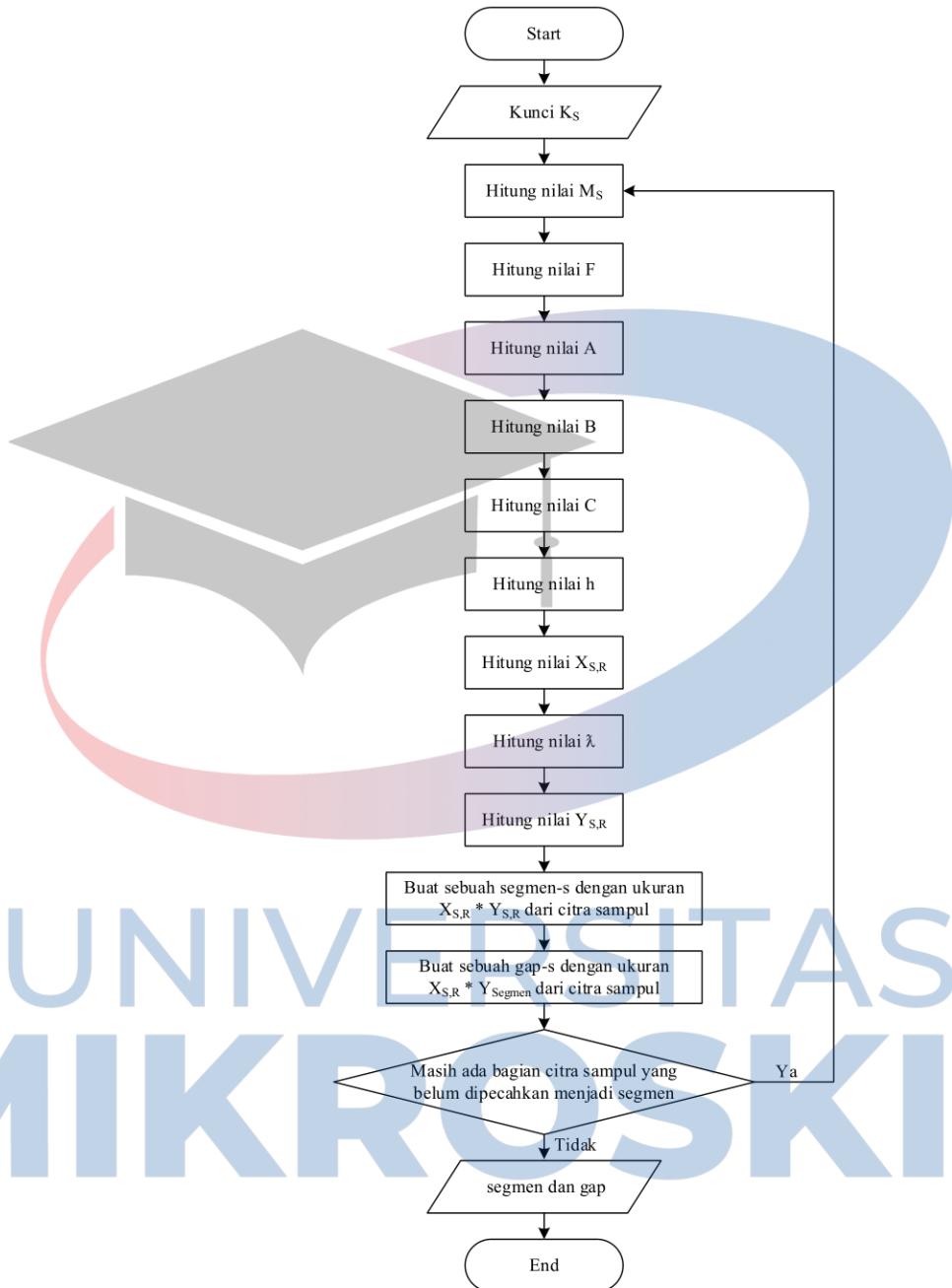


Gambar 3.7 *Flowchart* Proses Penyisipan dengan Metode Steganografi MLSB

Langkah kerja dari proses penyisipan dengan metode steganografi MLSB dapat dijabarkan sebagai berikut:

- a) *Input* Citra Sampul.  
Rincian warna RGB dari citra sampul.
- b) *Input ciphertext.*  
*Ciphertext* yang dihasilkan oleh metode MAES adalah  
93 F7 A6 13 CB 85 DE 0B 01 78 94 AF FE C3 9C 89
- c) Lakukan proses segmentasi citra dengan algoritma DSIS.

Proses segmentasi citra dengan algoritma DSIS dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:

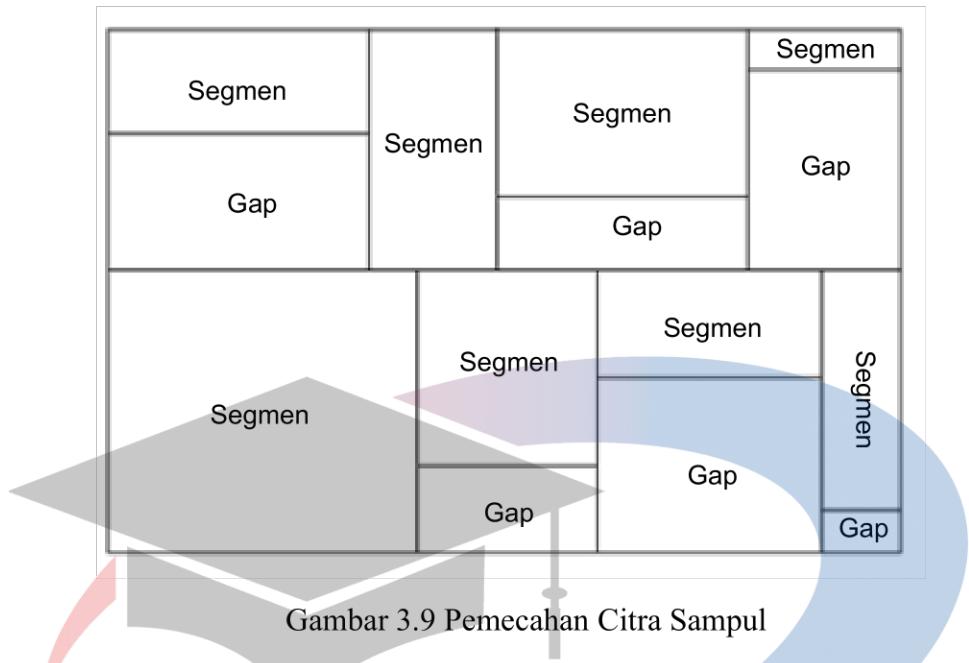


Gambar 3.8 *Flowchart* Proses Segmentasi Citra dengan algoritma DSIS

Langkah kerja dari proses segmentasi citra dengan algoritma DSIS dapat dirincikan sebagai berikut:

- 1) *Input* kunci  $K_S$ . Misalkan  $K_S = \text{'putra123'}$ .
- 2) Hitung nilai  $M_S$ .

Citra sampul yang digunakan akan dipecahkan menjadi segmen dan *gap* seperti terlihat pada gambar berikut:



$$M_1 = 'p' = \text{ASCII Code : } 112$$

$$M_2 = 'u' = \text{ASCII Code : } 117$$

$$M_3 = 't' = \text{ASCII Code : } 116$$

$$M_4 = 'r' = \text{ASCII Code : } 114$$

$$M_5 = 'a' = \text{ASCII Code : } 97$$

$$M_6 = '1' = \text{ASCII Code : } 49$$

$$M_7 = '2' = \text{ASCII Code : } 50$$

$$M_8 = '3' = \text{ASCII Code : } 51$$

- 3) Perhitungan untuk raster 1; segmen 1, hitung nilai F.

$$F = 300 - 51 = 249$$

- 4) Hitung nilai A.

$$A = \lfloor F/100 \rfloor = \lfloor 249/100 \rfloor = 2$$

- 5) Hitung nilai B.

$$B = \lfloor (F - A * 100)/10 \rfloor$$

$$B = \lfloor (249 - 2 * 100)/10 \rfloor$$

$$B = \lfloor 49/10 \rfloor = 4$$

- 6) Hitung nilai C.

$$C = (F - (A * 100) - (B * 10))$$

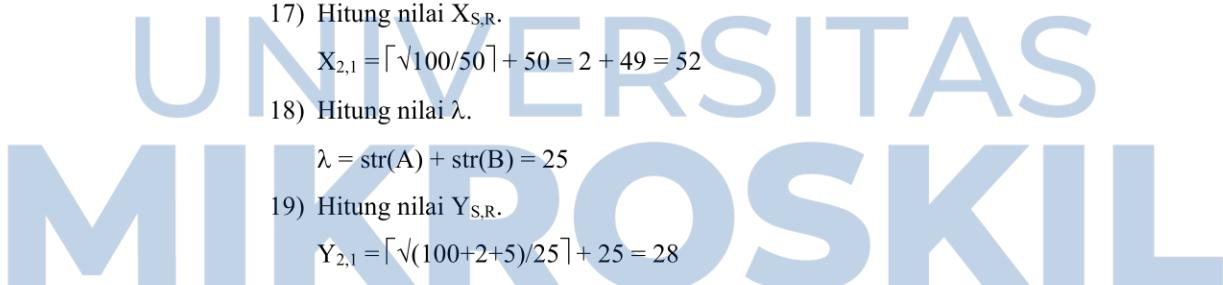
$$C = (249 - (2 * 100) - (4 * 10))$$

$$C = 9$$

- 7) Hitung nilai h.

$$h = \text{str}(B) + \text{str}(C) = 49$$

- 8) Hitung nilai  $X_{S,R}$ .  
 $X_{1,1} = \lceil \sqrt{100/49} \rceil + 49 = 2 + 49 = 51$
- 9) Hitung nilai  $\lambda$ .  
 $\lambda = \text{str}(A) + \text{str}(B) = 24$
- 10) Hitung nilai  $Y_{S,R}$ .  
 $Y_{1,1} = \lceil \sqrt{(100+2+4)/24} \rceil + 24 = 27$
- 11) Buat sebuah segmen  $s = 1$  pada raster 1 dengan ukuran segmen pertama =  $51 * 27$ .
- 12) Perhitungan untuk raster 1; segmen kedua, hitung nilai F.  
 $F = 300 - 50 = 250$
- 13) Hitung nilai A.  
 $A = \lfloor F/100 \rfloor = \lfloor 250/100 \rfloor = 2$
- 14) Hitung nilai B.  
 $B = \lfloor (F - A * 100)/10 \rfloor$   
 $B = \lfloor (250 - 2 * 100)/10 \rfloor$   
 $B = \lfloor 50/10 \rfloor = 5$
- 15) Hitung nilai C.  
 $C = (F - (A * 100) - (B * 10))$   
 $C = (250 - (2 * 100) - (5 * 10))$   
 $C = 0$
- 16) Hitung nilai h.  
 $h = \text{str}(B) + \text{str}(C) = 50$



- 17) Hitung nilai  $X_{S,R}$ .  
 $X_{2,1} = \lceil \sqrt{100/50} \rceil + 50 = 2 + 50 = 52$
- 18) Hitung nilai  $\lambda$ .  
 $\lambda = \text{str}(A) + \text{str}(B) = 25$
- 19) Hitung nilai  $Y_{S,R}$ .  
 $Y_{2,1} = \lceil \sqrt{(100+2+5)/25} \rceil + 25 = 28$
- 20) Buat sebuah segmen  $s = 2$  pada raster 1 dengan ukuran segmen kedua =  $52 * 28$ .

Karena ukuran citra sampul adalah  $100 * 100$ , maka ukuran segmen kedua adalah  $49 * 28$ .

- 21) Buat sebuah gap-1 dengan ukuran  $51 * 1$ .
- 22) Perhitungan untuk raster-2; segmen 2, hitung nilai F.

$$F = 300 - 49 = 251$$

- 23) Hitung nilai A.  
 $A = \lfloor F/100 \rfloor = \lfloor 251/100 \rfloor = 2$
- 24) Hitung nilai B.  
 $B = \lfloor (F - A * 100)/10 \rfloor$

$$B = \lfloor (251 - 2 * 100) / 10 \rfloor$$

$$B = \lfloor 51 / 10 \rfloor = 5$$

25) Hitung nilai C.

$$C = (F - (A * 100) - (B * 10))$$

$$C = (251 - (2 * 100) - (5 * 10))$$

$$C = 1$$

26) Hitung nilai h.

$$h = str(B) + str(C) = 51$$

27) Hitung nilai  $X_{S,R}$ .

$$X_{1,2} = \lceil \sqrt{100/51} \rceil + 51 = 2 + 51 = 53$$

28) Hitung nilai  $Y_{S,R}$ .

$$Y_{1,2} = 100 - 28 = 72$$

29) Buat sebuah segmen 1 pada raster-2 dengan ukuran:  $53 * 72$ .

30) Perhitungan untuk segmen 2 pada raster 2, hitung nilai F:

$$F = 300 - 97 = 203$$

31) Hitung nilai A.

$$A = \lfloor F / 100 \rfloor = \lfloor 203 / 100 \rfloor = 2$$

32) Hitung nilai B.

$$B = \lfloor (F - A * 100) / 10 \rfloor$$

$$B = \lfloor (203 - 2 * 100) / 10 \rfloor$$

$$B = \lfloor 3 / 10 \rfloor = 0$$

33) Hitung nilai  $\lambda$ .

$$\lambda = str(A) + str(B) = 20$$

34) Hitung nilai  $Y_{S,R}$ .

$$Y_{2,2} = \lceil \sqrt{(100+2+0)/20} \rceil + 20 = 23$$

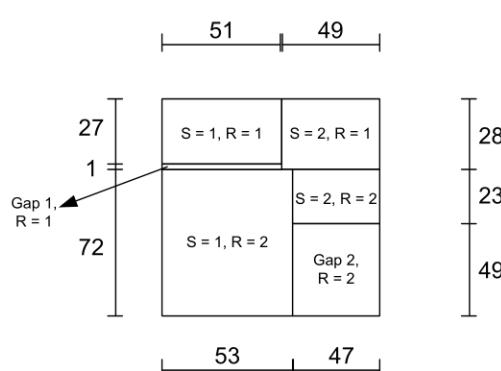
35) Hitung nilai  $X_{S,R}$

$$X_{2,2} = 100 - 53 = 47$$

36) Buat sebuah segmen  $s = 2$  pada raster 2 dengan ukuran segmen kedua =  $47 * 23$ .

37) Buat sebuah gap pada raster 2 dengan ukuran :  $47 * 49$ .

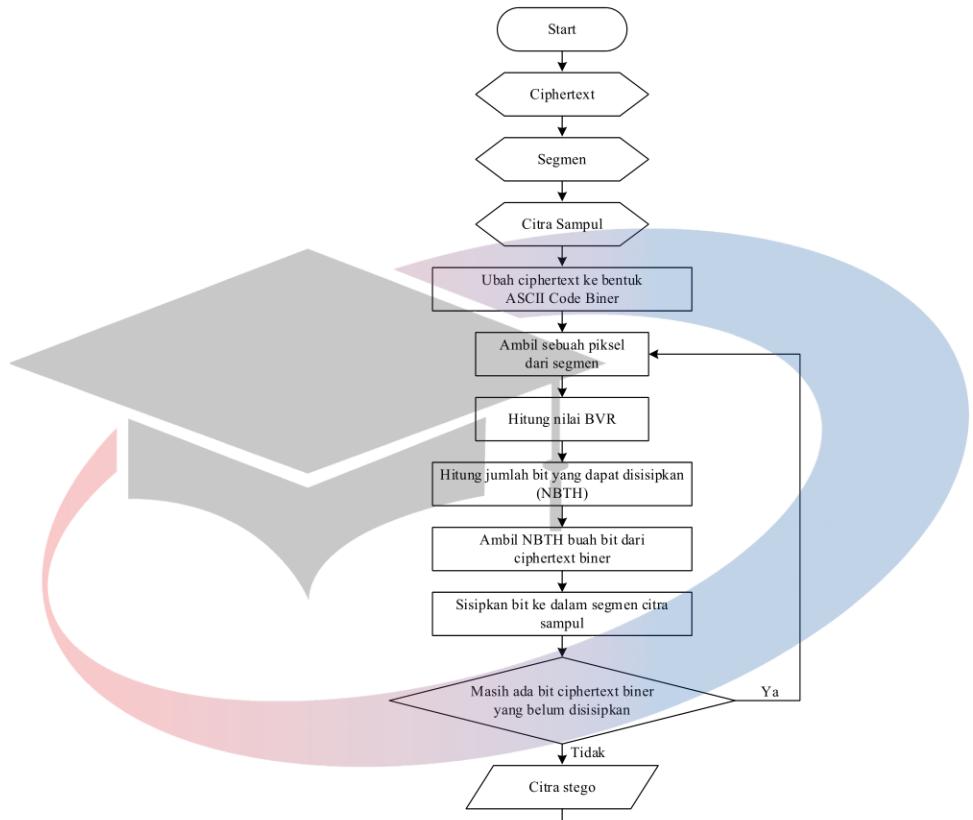
Hasil pembagian segmen dan gap dapat dilihat pada gambar berikut:



Gambar 3.10 Hasil Pembagian Segmen dan *Gap* pada Citra Sampul

- d) Sisipkan bit *ciphertext* ke citra hasil segmentasi.

Proses penyisipan bit *ciphertext* ke citra hasil segmentasi dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 3.11 *Flowchart* Proses Penyisipan Bit *Ciphertext*

Langkah kerja dari proses penyisipan bit *ciphertext* dapat dirincikan sebagai berikut:

- Input ciphertext.*  
*Ciphertext* yang dihasilkan oleh metode MAES adalah  
93 F7 A6 13 CB 85 DE 0B 01 78 94 AF FE C3 9C 89
- Input segmen* dapat dilihat pada gambar 3.11.
- Input citra sampul.*
- Ubah ciphertext* ke bentuk biner:  
1001 0011 1111 0111 1010 0110 0001 0011 1100 1011 1000 0101 1101 1110 0000  
1011 0000 0001 0111 1000 1001 0100 1010 1111 1111 1110 1100 0011 1001 1100  
1000 1001
- Proses dimulai dari segmen 1, raster 1. Ambil sebuah piksel dari segmen tersebut.
- Hitung nilai BVR.*

Segmen 1 *raster 1*: Ukuran  $51 * 27$  dengan elemen piksel berwarna putih semua. Jadi warna elemen RGB dari setiap piksel adalah (255, 255, 255). Setiap elemen warna tersebut akan diubah dari range (0 – 255) menjadi (1 – 16) dengan menggunakan rumusan BVR berikut:

$$BVR = \lfloor (255/16)+1 \rfloor = \lfloor (255/16)+1 \rfloor = 16$$

Jadi nilai BVR dari setiap piksel menjadi (16, 16, 16).

- f. Hitung nilai NBTH untuk menentukan jumlah bit yang dapat disisipkan ke dalam setiap piksel.

Varians ( $\sigma^2$ ) =

$$\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1}$$

Karena semua nilai piksel sama, maka nilai rata-rata  $\bar{x} = 16$ , sehingga nilai varians  $\sigma^2 = 0$ .

$$\begin{aligned} NBTH &= \lceil (\text{EXP}(0 * 16) / 16) * 0.6 \rceil^{1.7} = \lceil (\text{EXP}(0) * 0.6) \rceil^{1.7} \\ &= \lceil (1 * 0.6) \rceil^{1.7} = \lceil 0.419 \rceil = 1 \end{aligned}$$

Jadi setiap piksel dapat disisipkan 3 bit, masing-masing satu bit pada elemen R, G dan B.

- g. Sisipkan bit ke dalam piksel.

Segmen 1, raster 1:

Piksel 0, 0:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0, 1:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,2:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,3:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,4:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,5:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,6:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,7:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,8:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,9:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,10:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,11:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,12:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,13:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,14:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$



B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,15:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,16:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,17:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,18:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,19:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,20:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,21:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,22:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,23:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,24:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,25:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,26:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,27:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,28:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,29:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

Piksel 0,30:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = \mathbf{254}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,31:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,32:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

Piksel 0,33:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = \mathbf{255}$



Piksel 0,34:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,35:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,36:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,37:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,38:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

Piksel 0,39:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,40:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,41:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

Piksel 0,42:

R:  $255 = 1111\ 1111 \rightarrow 1111\ 1110 = 254$

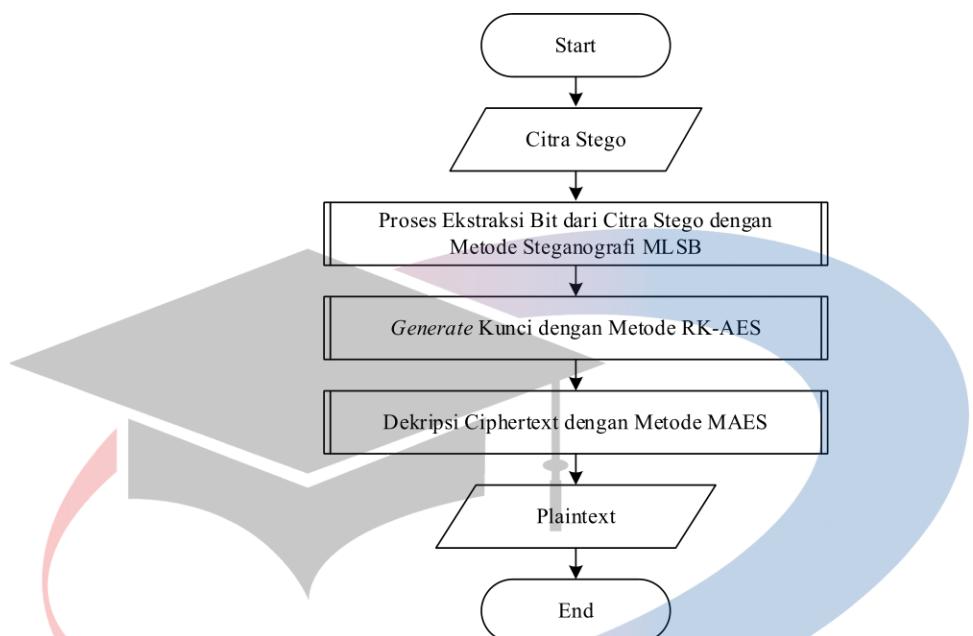
G:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

B:  $255 = 1111\ 1111 \rightarrow 1111\ 1111 = 255$

## 7. Output citra stego.

### 3.1.1.2 Analisis Proses Ekstraksi

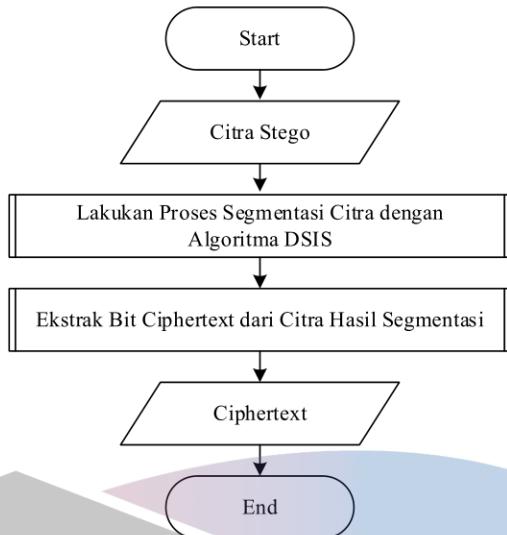
Prosedur kerja dari proses ekstraksi dengan menggunakan penggabungan dari algoritma kriptografi RK-MAES dan metode steganografi MLSB ini dapat dideskripsikan seperti terlihat pada gambar berikut:



Gambar 3.12 *Flowchart* Proses Ekstraksi dan Dekripsi dari Penggabungan Algoritma Kriptografi RK-MAES dan Metode Steganografi MLSB

Langkah kerja proses ekstraksi dari penggabungan algoritma kriptografi RK-MAES dan metode steganografi MLSB dapat dijabarkan sebagai berikut:

- 1) *Input* Citra Stego
- 2) Proses ekstraksi bit dari citra stego dengan metode steganografi MLSB.



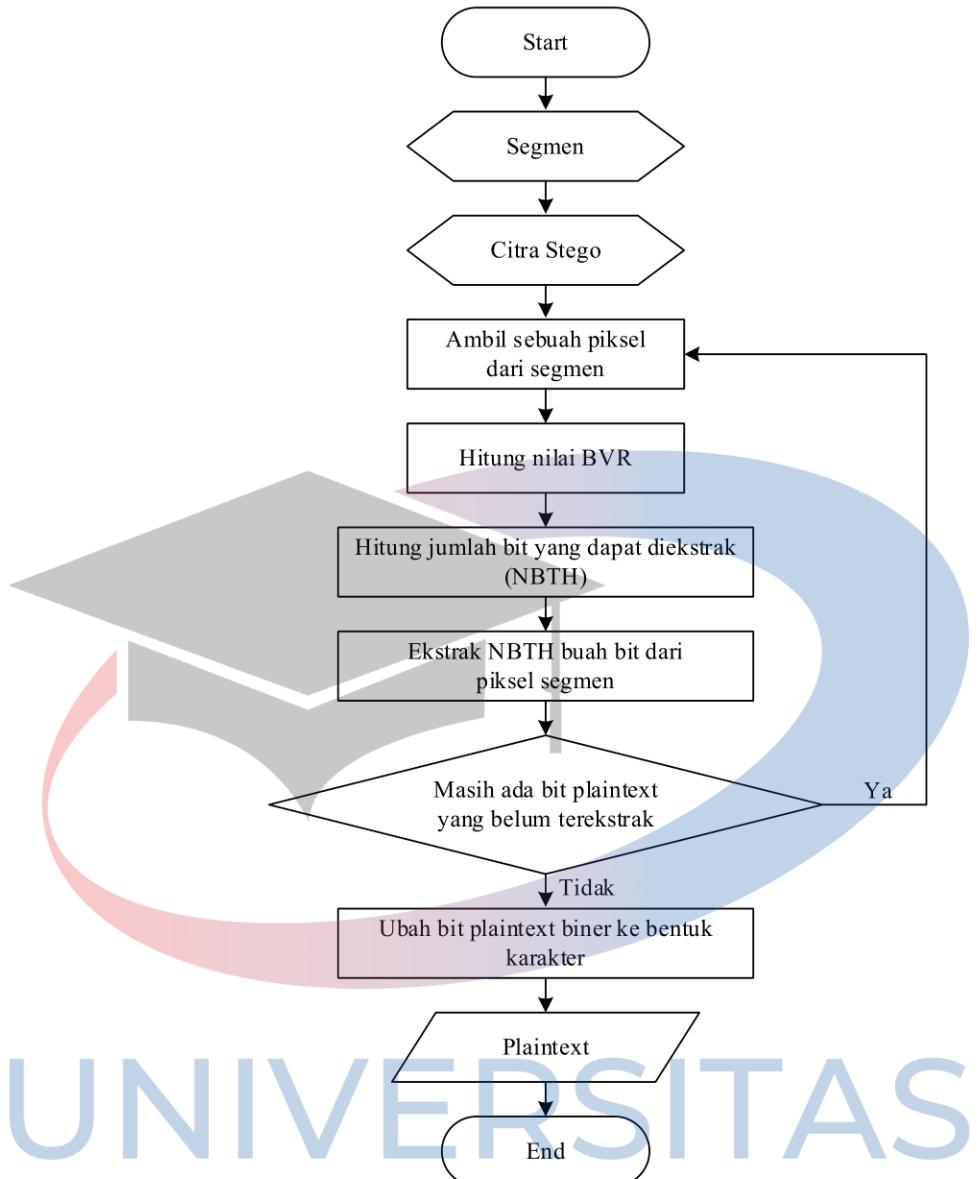
Gambar 3.13 *Flowchart* Proses Ekstraksi Bit dengan Metode Steganografi MLSB

Langkah kerja dari proses ekstraksi dengan metode steganografi MLSB dapat dijabarkan sebagai berikut:

- Input* Citra Stego.
- Lakukan proses segmentasi citra dengan algoritma DSIS.
- Ekstraksi *bit ciphertext* dari citra hasil segmentasi.

Proses segmentasi citra dengan algoritma DSIS sama seperti proses segmentasi citra pada prosedur penyisipan diatas. Hasil segmentasi yang diperoleh dapat dilihat pada gambar 3.11.

# UNIVERSITAS MIKROSKIL



Gambar 3.14 Flowchart Proses Ekstraksi Bit Ciphertext

Proses perhitungan dari ekstraksi bit *ciphertext* hampir sama dengan proses penyisipan diatas yaitu dalam hal perhitungan nilai BVR dan NBTH. Sesuai dengan perhitungan diatas, maka diketahui jumlah piksel yang dapat disisipkan adalah sebesar 3 bit per piksel, sehingga hasil ekstraksi bit dapat dirincikan sebagai berikut:

Segmen 1, raster 1:

Piksel 0, 0:

R:  $255 = 1111\ 1111 \rightarrow$  bit yang terekstrak = 1

G:  $254 = 1111\ 1110 \rightarrow$  bit yang terekstrak = 0

B:  $254 = 1111\ 1110 \rightarrow$  bit yang terekstrak = 0

Piksel 0, 1:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 2:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 3:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 4:

R: **255** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 5:

R: **254** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 6:

R: **254** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 7:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 8:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 9:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 10:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 11:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 12:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 13:

R: **254** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 1

B: **255** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 14:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 15:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 16:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 17:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 18:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 19:

R: **255** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 20:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 21:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 22:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 23:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 24:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 25:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 26:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 27:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **255** = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 28:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 29:

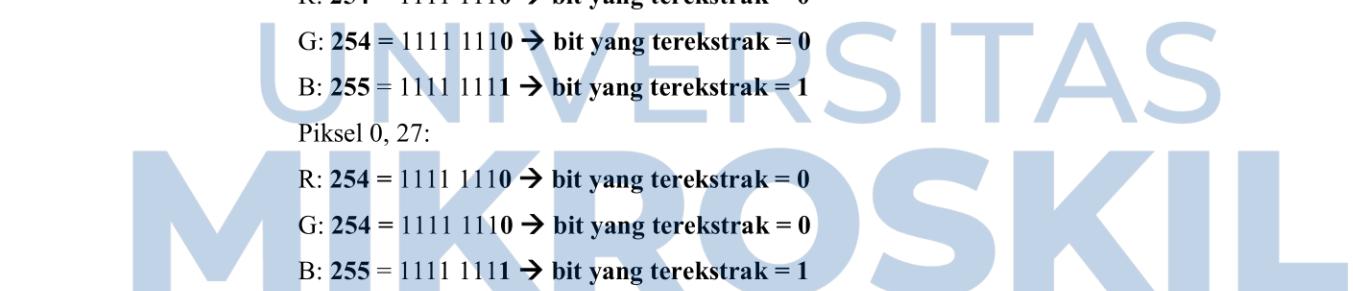
R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 30:

R: **255** = 1111 1111 → bit yang terekstrak = 1



G: 254 = 1111 1110 → bit yang terekstrak = 0

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 31:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 255 = 1111 1111 → bit yang terekstrak = 1

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 32:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 255 = 1111 1111 → bit yang terekstrak = 1

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 33:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 255 = 1111 1111 → bit yang terekstrak = 1

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 34:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 254 = 1111 1110 → bit yang terekstrak = 0

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 35:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 254 = 1111 1110 → bit yang terekstrak = 0

B: 254 = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 36:

R: 254 = 1111 1110 → bit yang terekstrak = 0

G: 254 = 1111 1110 → bit yang terekstrak = 0

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 37:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 255 = 1111 1111 → bit yang terekstrak = 1

B: 254 = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 38:

R: 254 = 1111 1110 → bit yang terekstrak = 0

G: 255 = 1111 1111 → bit yang terekstrak = 1

B: 255 = 1111 1111 → bit yang terekstrak = 1

Piksel 0, 39:

R: 255 = 1111 1111 → bit yang terekstrak = 1

G: 254 = 1111 1110 → bit yang terekstrak = 0

B: 254 = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 40:

R: **255** = 1111 1111 → bit yang terekstrak = 1

G: **254** = 1111 1110 → bit yang terekstrak = 0

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 41:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **254** = 1111 1110 → bit yang terekstrak = 0

Piksel 0, 42:

R: **254** = 1111 1110 → bit yang terekstrak = 0

G: **255** = 1111 1111 → bit yang terekstrak = 1

B: **255** = 1111 1111 → bit yang terekstrak = 1

Jadi, *bit ciphertext* yang terekstrak adalah:

1001 0011 1111 0111 1010 0110 0001 0011 1100 1011 1000 0101 1101 1110 0000 1011 0000

0001 0111 1000 1001 0100 1010 1111 1111 1110 1100 0011 1001 1100 1000 1001

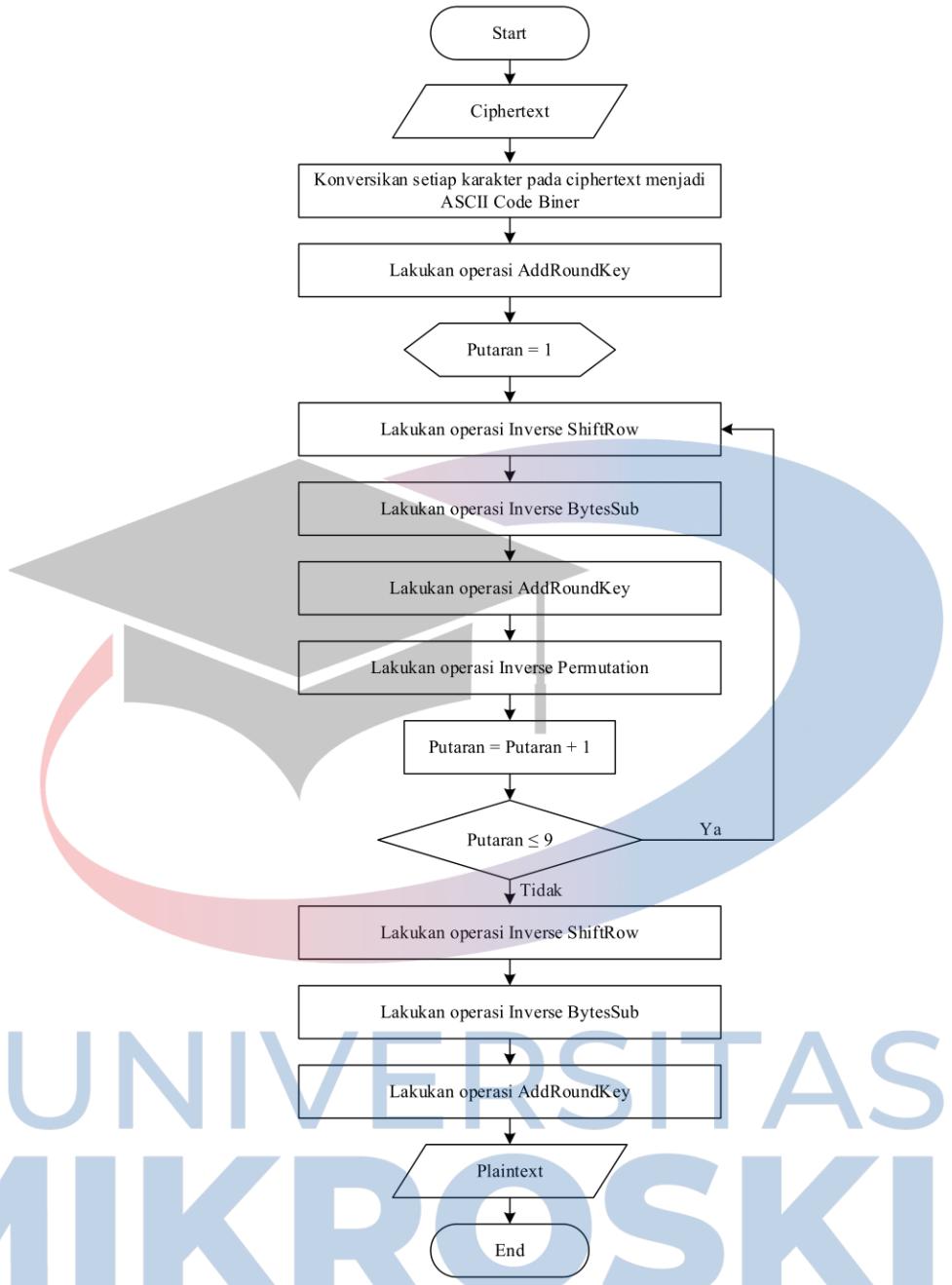
- 3) Proses *Generate* kunci dengan metode RK-AES.

Langkah kerja dari proses *Generate* kunci dengan metode RK-AES ini sama dengan proses *Generate* kunci pada proses enkripsi.

- 4) Proses dekripsi *ciphertext* dengan metode MAES.

Proses dekripsi *ciphertext* dengan menggunakan metode MAES dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:

# UNIVERSITAS MIKROSKIL



Gambar 3.15 Flowchart Proses Dekripsi *Ciphertext* dengan Metode MAES

Langkah kerja dari proses dekripsi *ciphertext* dengan metode MAES dapat dirincikan sebagai berikut:

1. Bit *ciphertext*:

(1001 0011 1111 0111 1010 0110 0001 0011 1100 1011 1000 0101 1101 1110 0000 1011  
0000 0001 0111 1000 1001 0100 1010 1111 1111 1110 1100 0011 1001 1100 1000 1001)<sub>2</sub>  
(93 F7 A6 13 CB 85 DE 0B 01 78 94 AF FE C3 9C 89)<sub>16</sub>

2. Lakukan operasi *AddRoundKey*:

State	$\oplus$	Subkunci(10)	$\rightarrow$	Hasil <i>AddRoundkey</i>
93 F7 A6 13		AF FD F9 EF		3C 0A 5F FC
CB 85 DE 0B		BF FD FF EF		74 78 21 E4
01 78 94 AF		7F FF FD FF		7E 87 69 50
FE C3 9C 89		FF 7F FF DF		01 BC 63 56

3. Putaran = 1.  
4. Lakukan operasi *Inverse Shift Row*.

Hasil *AddRoundKey* dilakukan transformasi *Inverse ShiftRow* yaitu:

=> Baris pertama tetap  
=> Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte  
=> Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte  
=> Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil *AddRoundKey*

3C 0A 5F FC  
74 78 21 E4  
7E 87 69 50  
01 BC 63 56

Hasil *Inverse ShiftRow*

3C 0A 5F FC  
E4 74 78 21  
69 50 7E 87  
BC 63 56 01

5. Lakukan operasi *Inverse SubBytes*.

Hasil *Inverse ShiftRow*

3C 0A 5F FC  
E4 74 78 21  
69 50 7E 87  
BC 63 56 01

Hasil *Inverse SubByte*

6D A3 84 55  
BE CA C1 7B  
E4 6C 8A EA  
78 00 B9 09

6. Lakukan proses transformasi *AddRoundKey* dengan SubKunci(9)

State	$\oplus$	Subkunci(9)	=	AddRoundkey
6D A3 84 55		77 EF FF DD		1A 4C 7B 88
BE CA C1 7B		77 FF FF FF		C9 35 3E 84
E4 6C 8A EA		77 EF FF FF		93 83 75 15
78 00 B9 09		3F EF FF FF		47 EF 46 F6

7. Lakukan operasi *Inverse Permutation*:

State:

A4 4C 3A DD  
6C 2C 94 83  
F5 FA AF 20  
47 26 AE 72

8. Putaran = 2.  
9. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation	Hasil Inverse ShiftRow
A4 4C 3A DD	A4 4C 3A DD
6C 2C 94 83	83 6C 2C 94
F5 FA AF 20	AF 20 F5 FA
47 26 AE 72	26 AE 72 47

10. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow	→	Hasil Inverse SubByte
A4 4C 3A DD		1D 5D A2 C9
83 6C 2C 94		41 B8 42 E7
AF 20 F5 FA		1B 54 77 14
26 AE 72 47		23 BE 1E 16

11. Lakukan proses transformasi AddRoundKey dengan SubKunci(8)

State	⊕	Subkunci(8)	=	AddRoundkey
1D 5D A2 C9		7B EF FB FF		66 B2 59 36
41 B8 42 E7		F7 FF DD FF		B6 47 9F 18
1B 54 77 14		7F 7F FD FF		64 2B 8A EB
23 BE 1E 16		57 7F FD DD		74 C1 E3 CB

12. Lakukan operasi Inverse Permutation:

State:

2C F9 E9 0E  
9F D1 64 98  
3B 1F C0 17  
80 D9 EB 2F

13. Putaran = 3.

14. Lakukan operasi Inverse Shift Row.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation	Hasil Inverse ShiftRow
2C F9 E9 0E	2C F9 E9 0E
9F D1 64 98	98 9F D1 64
3B 1F C0 17	C0 17 3B 1F

80 D9 EB 2F

D9 EB 2F 80

15. Lakukan operasi Inverse SubBytes.

Hasil *Inverse ShiftRow*



Hasil *Inverse SubByte*

2C F9 E9 0E

42 69 EB D7

98 9F D1 64

E2 6E 51 8C

C0 17 3B 1F

1F 87 49 CB

D9 EB 2F 80

E5 3C 4E 3A

16. Lakukan proses transformasi AddRoundKey dengan SubKunci(7)

State ⊕ Subkunci(7) = AddRoundkey

42 69 EB D7	FF EF FF FF	BD 86 14 28
E2 6E 51 8C	FF EF FF FD	1D 81 AE 71
1F 87 49 CB	FF EF FF FD	E0 68 B6 36
E5 3C 4E 3A	B5 FB FF FF	50 C7 B1 C5

17. Lakukan operasi *Inverse Permutation*:

State:

E2 18 DC C9  
C6 4B 02 78  
2A 25 27 10  
8D 5D F2 6E

18. Putaran = 4.

19. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation

E2 18 DC C9  
C6 4B 02 78  
2A 25 27 10  
8D 5D F2 6E

Hasil Inverse ShiftRow

E2 18 DC C9  
78 C6 4B 02  
27 10 2A 25  
5D F2 6E 8D

20. Lakukan operasi Inverse SubBytes.

Hasil *Inverse ShiftRow*



Hasil *Inverse SubByte*

E2 18 DC C9

3B 34 93 12

78 C6 4B 02

C1 C7 CC 6A

27 10 2A 25

3D 7C 95 C2

5D F2 6E 8D

8D 04 45 B4

21. Lakukan proses transformasi AddRoundKey dengan SubKunci(6)

State ⊕ Subkunci(6) = AddRoundkey

3B 34 93 12	72 76 D9 FE	49 92 4A EC
C1 C7 CC 6A	72 77 DD FF	B3 B0 11 95
3D 7C 95 C2	77 67 FF FF	4A 1B 6A 3D
8D 04 45 B4	D7 EF FF DD	5A EB BA 69

22. Lakukan operasi *Inverse Permutation*:

*State:*

CA 94 03 45  
BA A1 45 B3  
33 FC 01 FF  
99 2F E6 28

23. Putaran = 5.

24. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation	Hasil Inverse ShiftRow
CA 94 03 45	CA 94 03 45
BA A1 45 B3	B3 BA A1 45
33 FC 01 FF	01 FF 33 FC
99 2F E6 28	2F E6 E8 99

25. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow	→	Hasil Inverse SubByte
CA 94 03 45		10 E7 D5 68
B3 BA A1 45		4B C0 F1 68
01 FF 33 FC		09 7D 66 55
2F E6 E8 99		4E F5 EE F9

26. Lakukan proses transformasi AddRoundKey dengan SubKunci(5)

State	⊕	Subkunci(5)	=	AddRoundkey
10 E7 D5 68		FF FF FF FD		EF 18 2A 95
4B C0 F1 68		FB DF FF FD		B0 1F 0E 95
09 7D 66 55		FB FF FF FF		F2 B2 99 AA
4E F5 EE F9		31 78 77 23		7F 8D 99 DA

27. Lakukan operasi *Inverse Permutation*:

*State:*

63 6C 6B 7C  
B3 C4 40 C3

AC D3 A0 AF

CE C1 C2 7F

28. Putaran = 6.

29. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation

63 6C 6B 7C

B3 C4 40 C3

AC D3 A0 AF

CE C1 C2 7F

Hasil Inverse ShiftRow

63 6C 6B 7C

C3 B3 C4 40

A0 AF AC D3

C1 C2 7F CE

30. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow →

63 6C 6B 7C

C3 B3 C4 40

A0 AF AC D3

C1 C2 7F CE

Hasil Inverse SubByte

00 B8 05 01

33 4B 88 72

47 1B AA A9

DD A8 6B EC

31. Lakukan proses transformasi AddRoundKey dengan SubKunci(4)

State	⊕	Subkunci(4)	=	AddRoundkey
00 B8 05 01	E9 E7 FF EC			E9 5F FA ED
33 4B 88 72	FD FF FF FC			CE B4 77 8E
47 1B AA A9	FB EF FF FD			BC F4 55 54
DD A8 6B EC	5F FF FF FD			82 57 94 11

32. Lakukan operasi *Inverse Permutation*:

State:

59 9E BB D7

3C 6D DD E7

26 A0 7D 40

7F 50 35 D8

33. Putaran = 7.

34. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation	Hasil Inverse ShiftRow
59 9E BB D7	59 9E BB D7
3C 6D DD E7	E7 3C 6D DD
26 A0 7D 40	7D 40 26 A0
7F 50 35 D8	50 35 D8 7F

35. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow	→	Hasil Inverse SubByte
59 9E BB D7		15 DF FE 0D
E7 3C 6D DD		B0 6D B3 C9
7D 40 26 A0		13 72 23 47
50 35 D8 7F		6C D9 2D 6B

36. Lakukan proses transformasi AddRoundKey dengan SubKunci(3)

State	⊕	Subkunci(3)	=	AddRoundkey
15 DF FE 0D		FF FD FF FE		EA 22 01 F3
B0 6D B3 C9		FB DF FE DF		4B B2 4D 16
13 72 23 47		FF DF F7 DF		EC AD D4 98
6C D9 2D 6B		3D 7D F7 A7		51 A4 DA CC

37. Lakukan operasi Inverse Permutation:

State:

8D F3 0A C8  
23 71 C9 61  
90 08 76 5B  
8D 70 CE 7F

38. Putaran = 8.

39. Lakukan operasi Inverse Shift Row.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation	Hasil Inverse ShiftRow
8D F3 0A C8	8D F3 0A C8
23 71 C9 61	61 23 71 C9
90 08 76 5B	76 5B 90 08
8D 70 CE 7F	70 CE 7F 8D

40. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow	→	Hasil Inverse SubByte
8D F3 0A C8		B4 7E A3 B1

61 23 71 C9	D8 32 2C 12
76 5B 90 08	0F 57 96 BF
70 CE 7F 8D	D0 EC 6B B4

41. Lakukan proses transformasi AddRoundKey dengan SubKunci(2)

State	$\oplus$	Subkunci(2)	=	AddRoundkey
B4 7E A3 B1	F3 EF F7 EB			47 91 54 5A
D8 32 2C 12	D3 DF D7 EF			0B ED FB FD
0F 57 96 BF	FB EF EF FB			F4 B8 79 44
D0 EC 6B B4	5F FF FF FE			8F 13 94 4A

42. Lakukan operasi *Inverse Permutation*:

State:

FA C9 66 AB
1F 2A 6F 3A
A4 A2 C9 96
7C 54 47 D8

43. Putaran = 9.

44. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

- => Baris pertama tetap
- => Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte
- => Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte
- => Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation  
FA C9 66 AB  
1F 2A 6F 3A  
A4 A2 C9 96  
7C 54 47 D8

Hasil Inverse ShiftRow  
FA C9 66 AB  
3A 1F 2A 6F  
C9 96 A4 A2  
54 47 D8 7C

45. Lakukan operasi Inverse SubBytes.

Hasil Inverse ShiftRow	$\rightarrow$	Hasil Inverse SubByte
FA C9 66 AB		14 12 D3 0E
3A 1F 2A 6F		A2 CB 95 06
C9 96 A4 A2		12 35 1D 1A
54 47 D8 7C		8C 16 2D 01

46. Lakukan proses transformasi AddRoundKey dengan SubKunci(1)

State	$\oplus$	Subkunci(1)	=	AddRoundkey
14 12 D3 0E	ED FD EF FD			F9 EF 3C F3
A2 CB 95 06	7B DB DA DE			D9 10 4F D8
12 E5 1D 1A	D8 F7 F6 7F			CA 12 EB 65

8C 16 2D 01

1D 7D BF AE

91 6B 92 AF

47. Lakukan operasi *Inverse Permutation*:

*State:*

D9 19 1C DE

E7 55 DB D3

A7 7E 03 66

98 27 65 CE

48. Putaran = 10.

49. Lakukan operasi *Inverse Shift Row*.

Hasil AddRoundKey dilakukan transformasi Inverse ShiftRow yaitu:

=> Baris pertama tetap

=> Baris Ke-2 dirotasikan ke kanan sebanyak 1 byte

=> Baris Ke-3 dirotasikan ke kanan sebanyak 2 byte

=> Baris Ke-4 dirotasikan ke kanan sebanyak 3 byte

Hasil Inverse Permutation

D9 19 1C DE

E7 55 DB D3

A7 7E 03 66

98 27 65 CE

Hasil Inverse ShiftRow

D9 19 1C DE

D3 E7 55 DB

03 66 A7 7E

27 65 CE 98

50. Lakukan operasi *Inverse SubBytes*.

Hasil *Inverse ShiftRow*

D9 19 1C DE

D3 E7 55 DB

03 66 A7 7E

27 65 CE 98

Hasil *Inverse SubByte*

E5 8E C4 9C

A9 B0 ED 9F

D5 D3 89 8A

3D BC EC E2

51. Lakukan proses transformasi AddRoundKey dengan SubKunci(0)

State	$\oplus$	Subkunci(0)	=	AddRoundkey
E5 8E C4 9C		B5 EF B7 EF		50 61 73 73
A9 B0 ED 9F		DE DF 9F FB		77 6F 72 64
D5 D3 89 8A		F5 EE A9 EB		20 3D 20 61
3D BC EC E2		5F DF DD D0		62 63 31 32

52. Lakukan operasi *Inverse Permutation*:

*State:*

50 61 73 73

77 6F 72 64

20 3D 20 61

62 63 31 32

Hasil dekripsi yang diperoleh:

**01010000 01100001 01110011 01110011 01110111 01101111 01110010**  
**01100100 00100000 00111101 00100000 01100001 01100010 01100011**  
**00110001 00110010**

Konversikan ke bentuk karakter

<b>01010000</b>	:	80	:	P
<b>01100001</b>	:	97	:	a
<b>01110011</b>	:	115	:	s
<b>01110011</b>	:	115	:	s
<b>01110111</b>	:	119	:	w
<b>01101111</b>	:	111	:	o
<b>01110010</b>	:	114	:	r
<b>01100100</b>	:	100	:	d
<b>00100000</b>	:	32	:	
<b>00111101</b>	:	62	:	=
<b>00100000</b>	:	32	:	
<b>01100001</b>	:	97	:	a
<b>01100010</b>	:	98	:	b
<b>01100011</b>	:	99	:	c
<b>00110001</b>	:	49	:	1
<b>00110010</b>	:	50	:	2

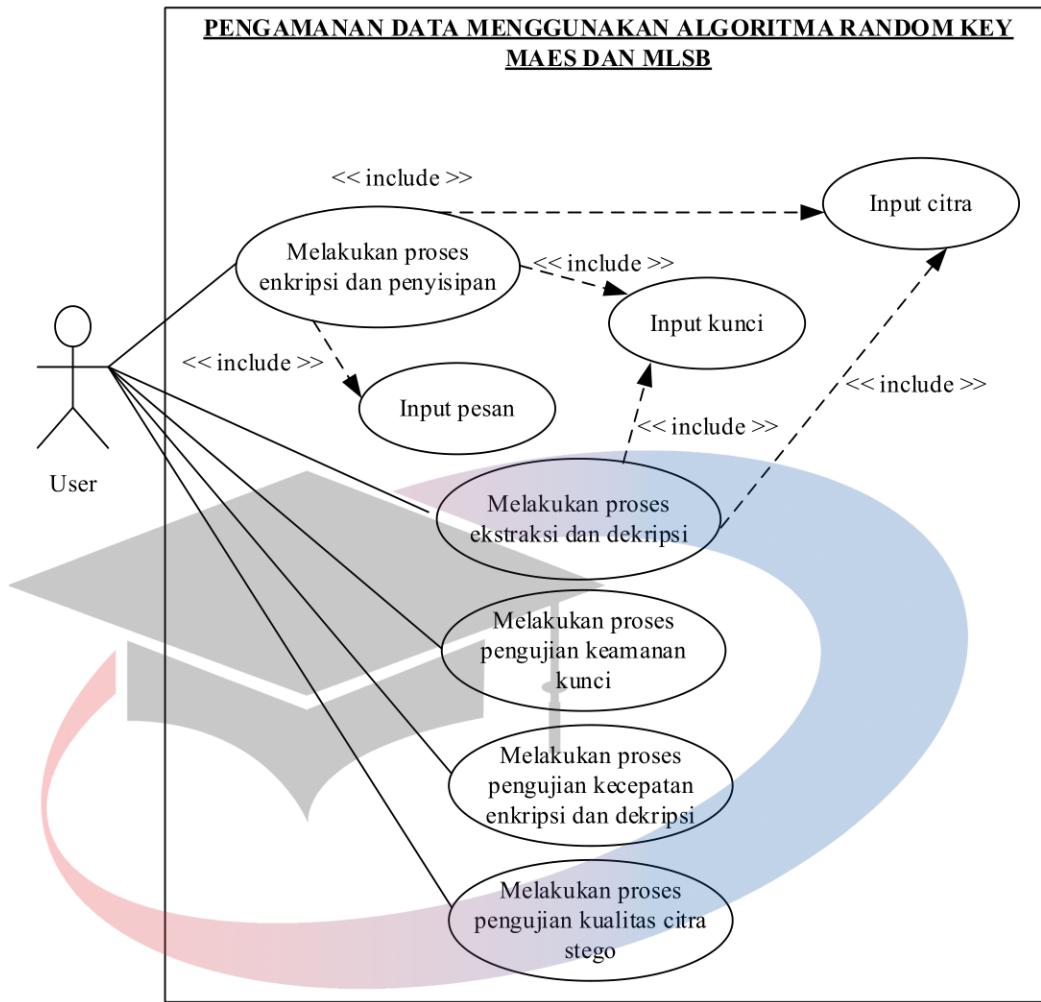
Pesan = “Password = abc12”

### 3.1.2 Analisis Kebutuhan

Analisis kebutuhan terhadap sistem yang akan dirancang mencakup Analisis fungsional yang mendeskripsikan fungsionalitas-fungsionalitas yang harus dipenuhi oleh perangkat lunak dan Analisis non fungsional yang mendeskripsikan persyaratan non fungsional yang berhubungan dengan kualitas sistem.

#### 3.1.2.1 Analisis Fungsional

Adapun beberapa persyaratan fungsional pada sistem pengamanan data menggunakan penggabungan dari algoritma kriptografi RK-MAES dan metode steganografi MLSB digambarkan dengan menggunakan *use case diagram* yang ditunjukkan pada gambar 3.16 sebagai berikut:



Gambar 3.16 *Use Case Diagram* dari Sistem Pengamanan Data

Narasi dari *use case* sistem pengamanan data diatas dapat dijabarkan sebagai berikut:

Tabel 3.1 *Use Case Input Citra*

Nama <i>use case</i>	<i>Input Citra Sampul</i>	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk membuka <i>file</i> citra sampul	
Prakondisi	-	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin membuka <i>file</i> citra sampul	
	<b>Aksi Aktor</b>	<b>Respons Sistem</b>

Bidang khas suatu event	1. <i>User</i> mengklik <i>link</i> ‘Browse’.  3. <i>User</i> memilih <i>file</i> yang diinginkan.	2. Sistem membuka kotak dialog open untuk pemilihan citra <i>input</i> .  4. Sistem membaca isi <i>file</i> .  5. Sistem menampilkan gambar yang dipilih.
Bidang alternatif	Alt Lgkh 2: Sistem menampilkan pesan kesalahan bahwa citra sampul <i>input</i> belum dimasukkan.	
Kesimpulan	<i>Use case</i> ini digunakan untuk membuka <i>file</i> citra sampul	
Postkondisi	Citra sampul telah ditampilkan	

Tabel 3.2 *Use Case Input* Pesan

Nama <i>use case</i>	<i>Input</i> Pesan	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk memasukkan pesan rahasia	
Prakondisi	-	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin memasukkan pesan rahasia	
	<b>Aksi Aktor</b>	<b>Respons Sistem</b>
Bidang khas suatu event	1. <i>User</i> mengklik <i>link</i> ‘Browse’.  3. <i>User</i> memilih <i>file</i> yang diinginkan.	2. Sistem membuka kotak dialog open untuk memilih <i>file</i> teks.  4. Sistem membaca isi <i>file</i> .  5. Sistem menampilkan isi <i>file</i> teks.

Bidang alternatif	Alt Lgkh 1: <i>User</i> memasukkan pesan rahasia Alt Lgkh 2: Sistem menampilkan pesan kesalahan bahwa <i>file input</i> belum dimasukkan.
Kesimpulan	<i>Use case</i> ini digunakan untuk memasukkan pesan rahasia
Postkondisi	Pesan rahasia telah dimasukkan

Tabel 3.3 *Use Case Input Kunci*

Nama <i>use case</i>	<i>Input Kunci</i>	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk memasukkan kunci	
Prakondisi	-	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin memasukkan kunci	
	<b>Aksi Aktor</b>	<b>Respons Sistem</b>
Bidang khas suatu event	1. <i>User</i> memasukkan kunci rahasia. 2. <i>User</i> memasukkan kunci Ks.	
Bidang alternatif	-	
Kesimpulan	<i>Use case</i> ini digunakan untuk memasukkan kunci	
Postkondisi	Kunci telah dimasukkan	

Tabel 3.4 *Use Case Melakukan Proses Enkripsi dan Penyisipan*

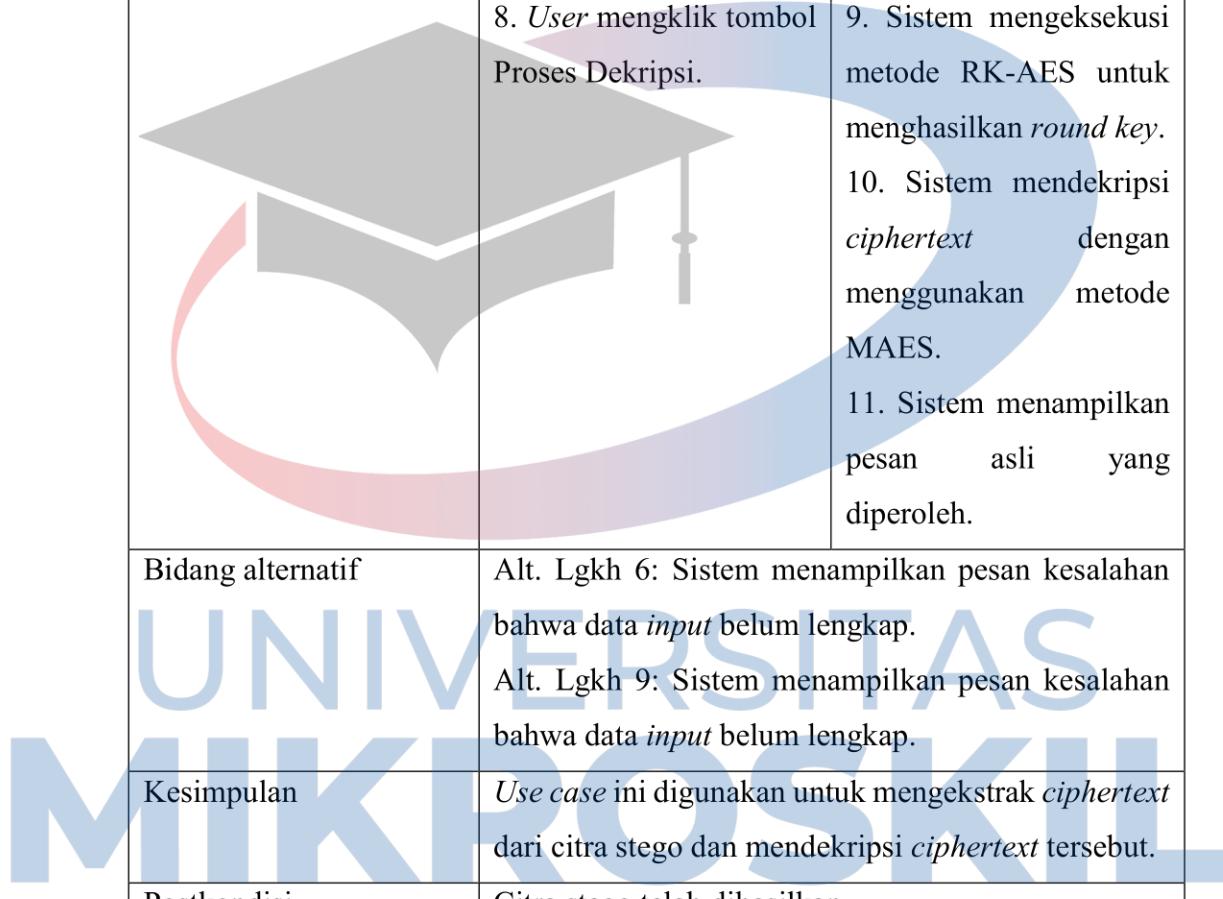
Nama <i>use case</i>	Melakukan Proses Enkripsi dan Penyisipan
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses enkripsi dan penyisipan
Prakondisi	Citra stego belum dihasilkan

Sasaran	<i>Use case ini diawali saat user ingin mengenkripsi pesan rahasia dan menyisipkan ciphertext ke citra sampul</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	<p>1. <i>User</i> mengklik tombol Enkripsi dan Penyisipan.</p> <p>3. <i>User</i> memasukkan pesan rahasia.</p> <p>4. <i>User</i> memasukkan kunci rahasia.</p> <p>5. <i>User</i> mengklik tombol Proses Enkripsi.</p> <p>9. <i>User</i> memasukkan citra sampul.</p> <p>10. <i>User</i> memasukkan kunci Ks.</p> <p>11. <i>User</i> mengklik tombol Proses Penyisipan.</p>	<p>2. Sistem menampilkan form Enkripsi dan Penyisipan.</p> <p>6. Sistem mengeksekusi metode RK-AES untuk menghasilkan <i>round key</i>.</p> <p>7. Sistem mengenkripsi pesan rahasia dengan menggunakan metode MAES.</p> <p>8. Sistem menampilkan ciphertext yang diperoleh.</p> <p>12. Sistem menyisipkan ciphertext ke dalam citra sampul dengan menggunakan metode MLSB.</p>

	<p>14. <i>User</i> mengklik tombol Simpan.</p> <p>16. <i>User</i> memilih lokasi penyimpanan dan memasukkan nama <i>file</i>.</p>	<p>13. Sistem menampilkan citra stego yang diperoleh.</p> <p>15. Sistem membuka kotak dialog Simpan.</p> <p>17. Sistem menyimpan citra stego yang diperoleh ke dalam <i>file</i>.</p>
Bidang alternatif	<p>Alt. Lgkh 6: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.</p> <p>Alt. Lgkh 12: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.</p>	
Kesimpulan	<i>Use case</i> ini digunakan untuk mengenkripsi pesan rahasia dan menyisipkan <i>ciphertext</i> ke dalam citra sampul.	
Postkondisi	Citra stego telah dihasilkan.	

Tabel 3.5 *Use Case* Melakukan Proses Ekstraksi dan Dekripsi

Nama <i>use case</i>	Melakukan Proses Ekstraksi dan Dekripsi	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses ekstraksi dan dekripsi	
Prakondisi	Pesan rahasia belum diperoleh	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin mengekstrak <i>ciphertext</i> dari citra stego dan mendekripsi <i>ciphertext</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	<p>1. <i>User</i> mengklik tombol Ekstraksi dan Dekripsi.</p> <p>3. <i>User</i> memasukkan citra stego.</p>	<p>2. Sistem menampilkan form Ekstraksi dan Dekripsi.</p>



	<p>4. <i>User</i> memasukkan kunci Ks.</p> <p>5. <i>User</i> mengklik tombol Proses Ekstraksi.</p> <p>7. <i>User</i> memasukkan kunci rahasia.</p> <p>8. <i>User</i> mengklik tombol Proses Dekripsi.</p>	<p>6. Sistem mengekstrak <i>ciphertext</i> dari citra stego dengan menggunakan metode MLSB.</p> <p>9. Sistem mengeksekusi metode RK-AES untuk menghasilkan <i>round key</i>.</p> <p>10. Sistem mendekripsi <i>ciphertext</i> dengan menggunakan metode MAES.</p> <p>11. Sistem menampilkan pesan asli yang diperoleh.</p>
Bidang alternatif	Alt. Lgkh 6: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap. Alt. Lgkh 9: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.	
Kesimpulan	<i>Use case</i> ini digunakan untuk mengekstrak <i>ciphertext</i> dari citra stego dan mendekripsi <i>ciphertext</i> tersebut.	
Postkondisi	Citra stego telah dihasilkan.	

Tabel 3.6 *Use Case* Melakukan Proses Pengujian Keamanan Kunci

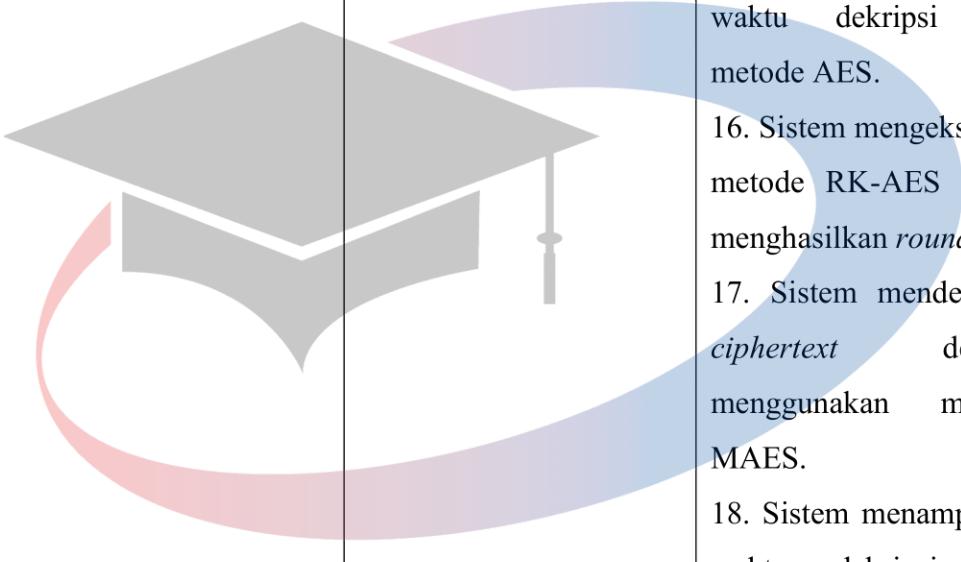
Nama <i>use case</i>	Melakukan Proses Pengujian Keamanan Kunci
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses pengujian keamanan kunci

Prakondisi	Hasil pengujian keamanan kunci belum diperoleh	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menguji keamanan kunci dari metode RK-AES	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	<p>1. <i>User</i> mengklik tombol Pengujian Keamanan Kunci dengan Metode RK-AES.</p> <p>3. <i>User</i> memasukkan kunci.</p> <p>4. <i>User</i> mengklik tombol Proses.</p>	<p>2. Sistem menampilkan form Pengujian Keamanan Kunci.</p> <p>5. Sistem mengeksekusi metode AES dan metode RK-AES untuk menghasilkan <i>round key</i>.</p> <p>6. Sistem menampilkan sub kunci yang dihasilkan.</p>
Bidang alternatif	Alt. Lgkh 5: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.	
Kesimpulan	<i>Use case</i> ini digunakan untuk menguji keamanan kunci dari metode RK-AES.	
Postkondisi	Hasil pengujian keamanan kunci telah diperoleh.	

Tabel 3.7 *Use Case* Melakukan Proses Pengujian Kecepatan Enkripsi dan Dekripsi

Nama <i>use case</i>	Melakukan Proses Pengujian Kecepatan Enkripsi dan Dekripsi
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses pengujian kecepatan enkripsi dan dekripsi
Prakondisi	Hasil pengujian kecepatan enkripsi dan dekripsi belum diperoleh

Sasaran	<i>Use case ini diawali saat user ingin menguji kecepatan enkripsi dan dekripsi dari metode MAES</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	<p>1. <i>User</i> mengklik tombol Pengujian Kecepatan Enkripsi dan Dekripsi dengan Metode RK-AES.</p> <p>3. <i>User</i> memasukkan pesan rahasia.</p> <p>4. <i>User</i> memasukkan kunci.</p> <p>5. <i>User</i> mengklik tombol Proses Enkripsi.</p>	<p>2. Sistem menampilkan <i>form</i> Pengujian Enkripsi dan Dekripsi.</p> <p>6. Sistem mengeksekusi metode AES untuk menghasilkan <i>round key</i>.</p> <p>7. Sistem mengenkripsi pesan rahasia dengan menggunakan metode AES.</p> <p>8. Sistem menampilkan waktu enkripsi dari metode AES.</p> <p>9. Sistem mengeksekusi metode RK-AES untuk menghasilkan <i>round key</i>.</p> <p>10. Sistem mengenkripsi pesan rahasia dengan menggunakan metode MAES.</p> <p>11. Sistem menampilkan waktu enkripsi dari metode MAES.</p>

	<p>12. User mengklik tombol Proses Dekripsi.</p> 	<p>13. Sistem mengeksekusi metode AES untuk menghasilkan <i>round key</i>.</p> <p>14. Sistem mendekripsi <i>ciphertext</i> dengan menggunakan metode AES.</p> <p>15. Sistem menampilkan waktu dekripsi dari metode AES.</p> <p>16. Sistem mengeksekusi metode RK-AES untuk menghasilkan <i>round key</i>.</p> <p>17. Sistem mendekripsi <i>ciphertext</i> dengan menggunakan metode MAES.</p> <p>18. Sistem menampilkan waktu dekripsi dari metode MAES.</p>
Bidang alternatif	<p>Alt. Lgkh 6: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.</p> <p>Alt. Lgkh 13: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.</p>	
Kesimpulan	<p><i>Use case</i> ini digunakan untuk menguji kecepatan enkripsi dan dekripsi dari metode MAES.</p>	
Postkondisi	Hasil pengujian kecepatan enkripsi dan dekripsi telah diperoleh.	

Tabel 3.8 *Use Case* Melakukan Proses Pengujian Kualitas Citra Stego

Nama <i>use case</i>	Melakukan Proses Pengujian Kualitas Citra Stego
----------------------	---

Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses pengujian kualitas citra stego	
Prakondisi	Hasil pengujian kualitas citra stego belum diperoleh	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menguji kualitas citra stego dari metode MLSB	
	<b>Aksi Aktor</b>	<b>Respons Sistem</b>
Bidang khas suatu event	<p>1. <i>User</i> mengklik tombol Pengujian Kualitas Citra Stego dengan Metode MLSB.</p> <p>3. <i>User</i> memasukkan citra sampul.</p> <p>5. <i>User</i> memasukkan citra stego.</p> <p>7. <i>User</i> mengklik tombol Proses.</p>	<p>2. Sistem menampilkan <i>form</i> Pengujian Kualitas Citra Stego.</p> <p>4. Sistem menampilkan citra sampul yang dimasukkan.</p> <p>6. Sistem menampilkan citra stego yang dimasukkan.</p> <p>8. Sistem membandingkan kedua citra dengan menggunakan metode MSE.</p> <p>9. Sistem menghitung nilai PSNR berdasarkan nilai MSE yang dihasilkan.</p> <p>10. Sistem menampilkan nilai MSE dan PSNR yang diperoleh.</p>
Bidang alternatif	Alt. Lgkh 8: Sistem menampilkan pesan kesalahan bahwa data <i>input</i> belum lengkap.	

Kesimpulan	<i>Use case</i> ini digunakan untuk menguji kualitas citra stego dari metode MLSB.
Postkondisi	Hasil pengujian kualitas citra stego telah diperoleh.

### 3.1.2.2 Analisis Non Fungsional

Untuk merumuskan persyaratan non-fungsional dari sistem, maka harus dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan customer. Panduan ini dikenal dengan analisis PIECES (*Performance, Information, Economic, Control, Efficiency, dan Services*).

#### 1. *Performance*

Perangkat lunak memiliki waktu proses kerja yang relatif lebih cepat dikarenakan kompleksitas perhitungan yang sudah dipermudah.

#### 2. *Information*

Perangkat lunak menyediakan informasi yang menjelaskan setiap fitur, tombol dan link yang terdapat dalam sistem.

#### 3. *Economics*

Perangkat lunak tidak memerlukan perangkat dukung lainnya dalam proses eksekusinya. Aplikasi yang dibutuhkan hanya Microsoft Visual C# 2013.

#### 4. *Control*

Perangkat lunak akan menampilkan pesan kesalahan apabila terdapat kesalahan atau kegagalan sistem pada input ataupun prosesnya.

#### 5. *Efficiency*

Perangkat lunak melakukan proses perhitungan yang relatif lebih cepat dikarenakan memori yang dipakai oleh metode perhitungan enkripsi lebih sedikit.

#### 6. *Service*

Perangkat lunak mampu melakukan proses terhadap citra dengan ukuran berapapun tetapi dengan minimal ukuran citra 100 x 100.

### 3.2 Perancangan

Aplikasi pengamanan data ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual C#* dengan menggunakan beberapa objek dasar seperti:

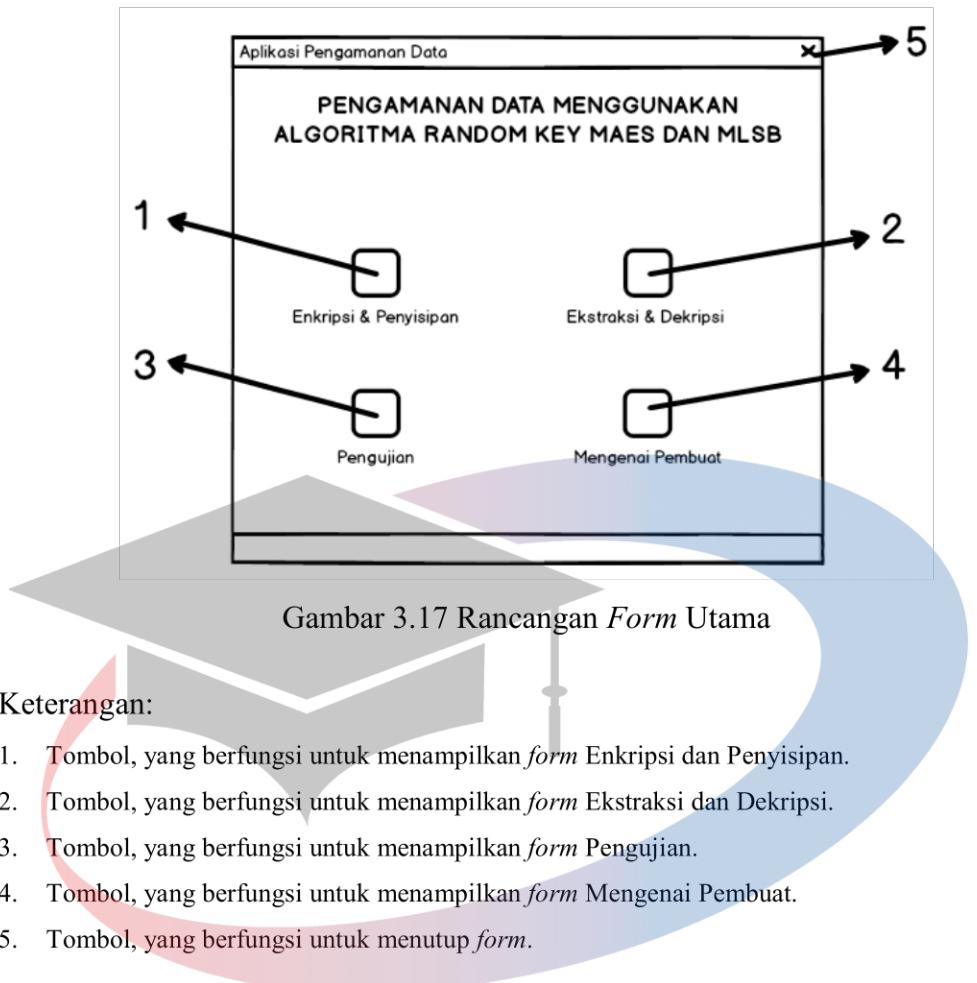
1. Button, yang dipakai sebagai tombol eksekusi.
2. Label, yang digunakan untuk penamaan disetiap objek.
3. Textbox, yang digunakan sebagai tempat penginputan nilai.
4. OpenFileDialog, yang digunakan untuk menampilkan dialog open.
5. SaveFileDialog, yang digunakan untuk menampilkan dialog save.
6. Picture box, yang digunakan untuk menampilkan gambar.
7. Check box, yang digunakan untuk pilihan menampilkan laporan atau tidak.
8. Group box, yang digunakan untuk menggabungkan objek.

Aplikasi pengamanan data ini memiliki beberapa *form* berikut:

1. *Form* Utama.
2. *Form* Enkripsi.
3. *Form* Dekripsi.
4. *Form* Pengujian.
5. *Form* Pengujian Keamanan Kunci.
6. *Form* Pengujian Kecepatan Enkripsi dan Dekripsi.
7. *Form* Pengujian Kualitas Citra Stego.

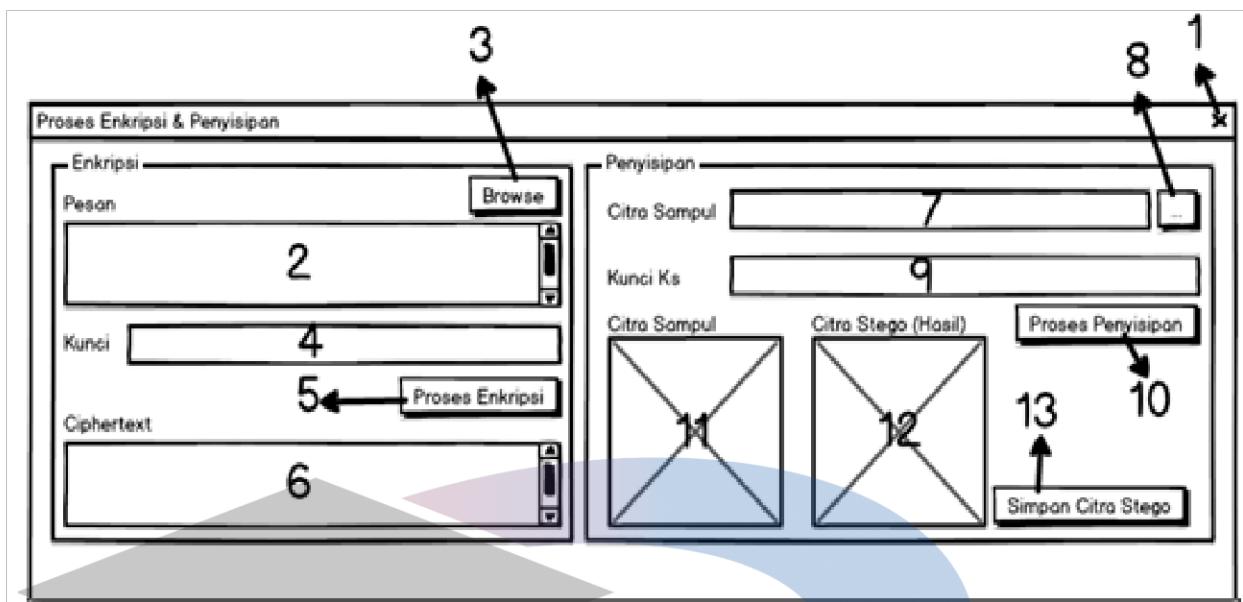
#### 3.2.1 *Form* Utama

*Form* Utama merupakan tampilan yang pertama kali muncul pada saat menjalankan aplikasi pengamanan data. Pada *form* Utama terdapat beberapa tombol yang dapat digunakan untuk mengakses *form-form* lainnya yang terdapat pada sistem. Rancangan *form* Utama dapat dilihat pada gambar berikut:



### 3.2.2 *Form* Enkripsi dan Penyisipan

*Form* Enkripsi dan Penyisipan berfungsi untuk melakukan proses enkripsi terhadap pesan rahasia dan penyisipan *ciphertext* yang dihasilkan ke dalam citra sampul yang dimasukkan. Rancangan *form* Enkripsi dan Penyisipan dapat dilihat pada gambar berikut:



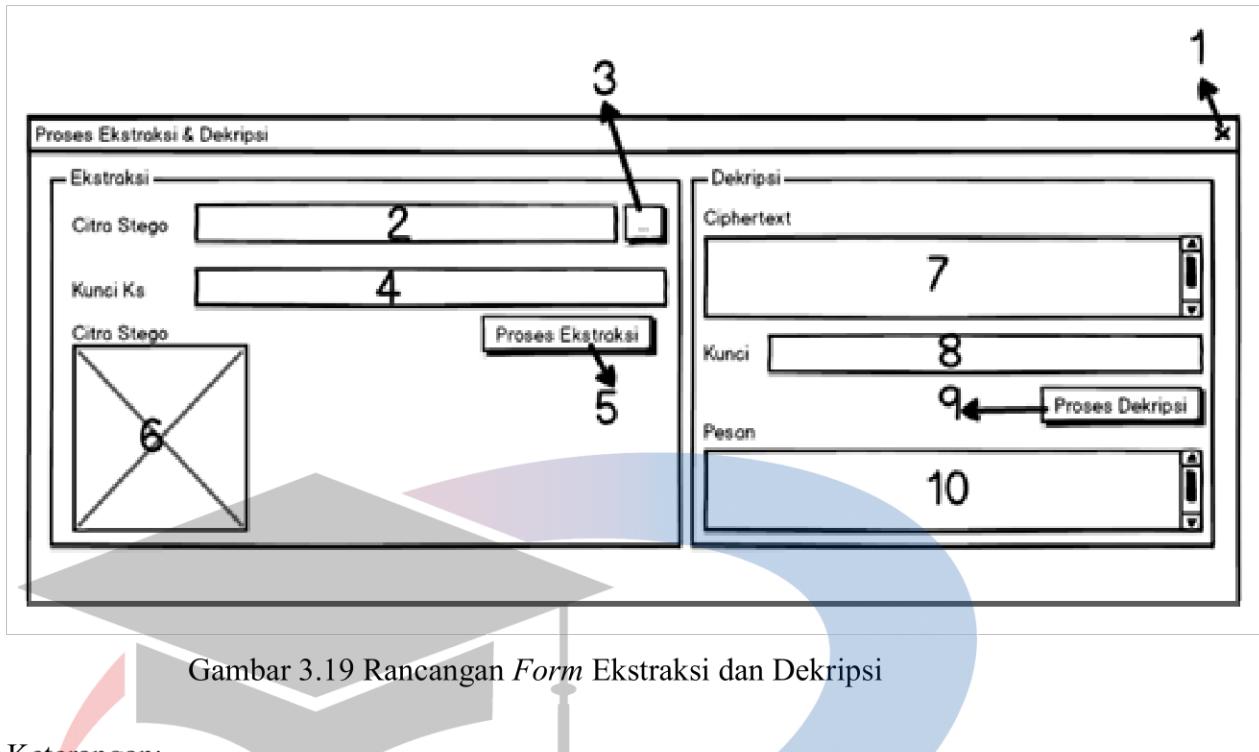
Gambar 3.18 Rancangan *Form* Enkripsi dan Penyisipan

Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.
2. *Textbox*, yang berfungsi sebagai tempat pengisian pesan.
3. Tombol, yang berfungsi untuk menampilkan kotak dialog *Open* untuk memilih *file* teks pesan.
4. *Textbox*, yang berfungsi sebagai tempat pengisian kunci.
5. Tombol, yang berfungsi untuk melakukan proses enkripsi.
6. *Textbox*, yang berfungsi untuk menampilkan *ciphertext* hasil enkripsi.
7. *Textbox*, yang berfungsi untuk menampilkan lokasi citra sampul.
8. Tombol, yang berfungsi untuk menampilkan kotak dialog *Open* untuk memilih *file* citra sampul.
9. *Textbox*, yang berfungsi sebagai tempat pengisian kunci Ks.
10. Tombol, yang berfungsi untuk melakukan proses penyisipan.
11. *Picturebox*, yang berfungsi untuk menampilkan citra sampul.
12. *Picturebox*, yang berfungsi untuk menampilkan citra stego.
13. Tombol, yang berfungsi untuk menampilkan kotak dialog *Save* untuk memilih lokasi penyimpanan citra stego.

### 3.2.3 *Form* Ekstraksi dan Dekripsi

*Form* Ekstraksi dan Dekripsi berfungsi untuk melakukan proses ekstraksi *ciphertext* dari citra stego dan mendekripsi *ciphertext* yang terekstrak. Rancangan *form* Ekstraksi dan Dekripsi dapat dilihat pada gambar berikut:



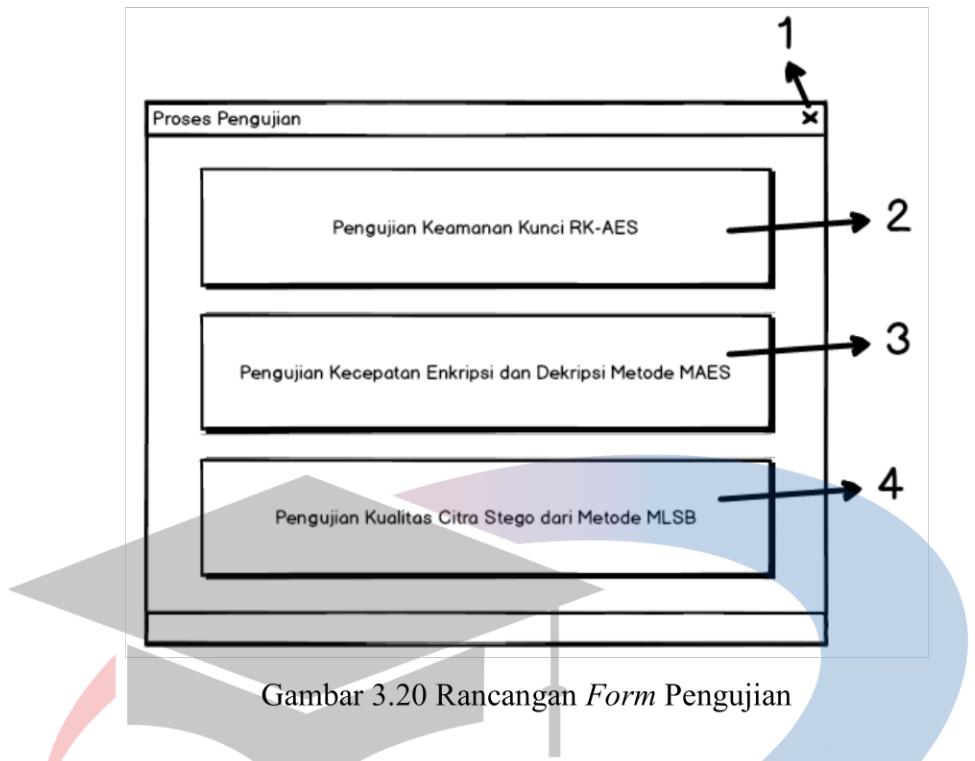
Gambar 3.19 Rancangan Form Ekstraksi dan Dekripsi

Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.
2. *Textbox*, yang berfungsi untuk menampilkan lokasi citra stego.
3. Tombol, yang berfungsi untuk menampilkan kotak dialog *Open* untuk memilih *file* citra stego.
4. *Textbox*, yang berfungsi sebagai tempat pengisian kunci Ks.
5. Tombol, yang berfungsi untuk melakukan proses ekstraksi.
6. *Picturebox*, yang berfungsi untuk menampilkan citra stego.
7. *Textbox*, yang berfungsi untuk menampilkan *ciphertext* yang terekstrak.
8. *Textbox*, yang berfungsi sebagai tempat pengisian kunci.
9. Tombol, yang berfungsi untuk melakukan proses dekripsi.
10. *Textbox*, yang berfungsi untuk menampilkan pesan hasil dekripsi.

### 3.2.4 Form Pengujian

*Form* Pengujian merupakan tampilan yang dapat digunakan untuk melakukan pengujian terhadap metode yang digunakan dalam sistem pengamanan data. Pada *form* Pengujian terdapat beberapa tombol yang dapat digunakan untuk mengakses *form* Pengujian lainnya yang terdapat pada sistem. Rancangan *form* Pengujian dapat dilihat pada gambar berikut:

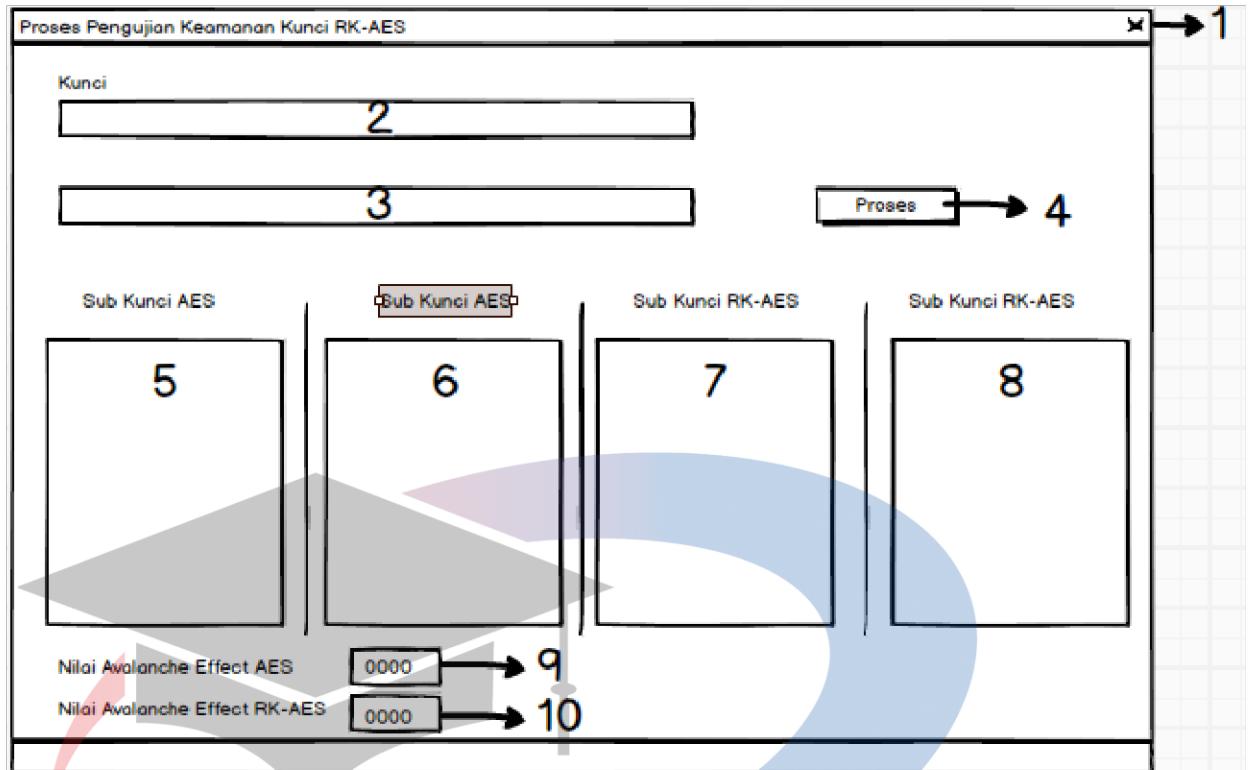


Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.
2. Tombol, yang berfungsi untuk menampilkan *form* Pengujian Keamanan Kunci.
3. Tombol, yang berfungsi untuk menampilkan *form* Pengujian Kecepatan Enkripsi dan Dekripsi.
4. Tombol, yang berfungsi untuk menampilkan *form* Pengujian Kualitas Citra Stego.

### 3.2.5 *Form* Pengujian Keamanan Kunci

*Form* Pengujian Keamanan Kunci berfungsi untuk melakukan pengujian keamanan kunci terhadap metode RK-AES. Rancangan *form* Pengujian Keamanan Kunci dapat dilihat pada gambar berikut:



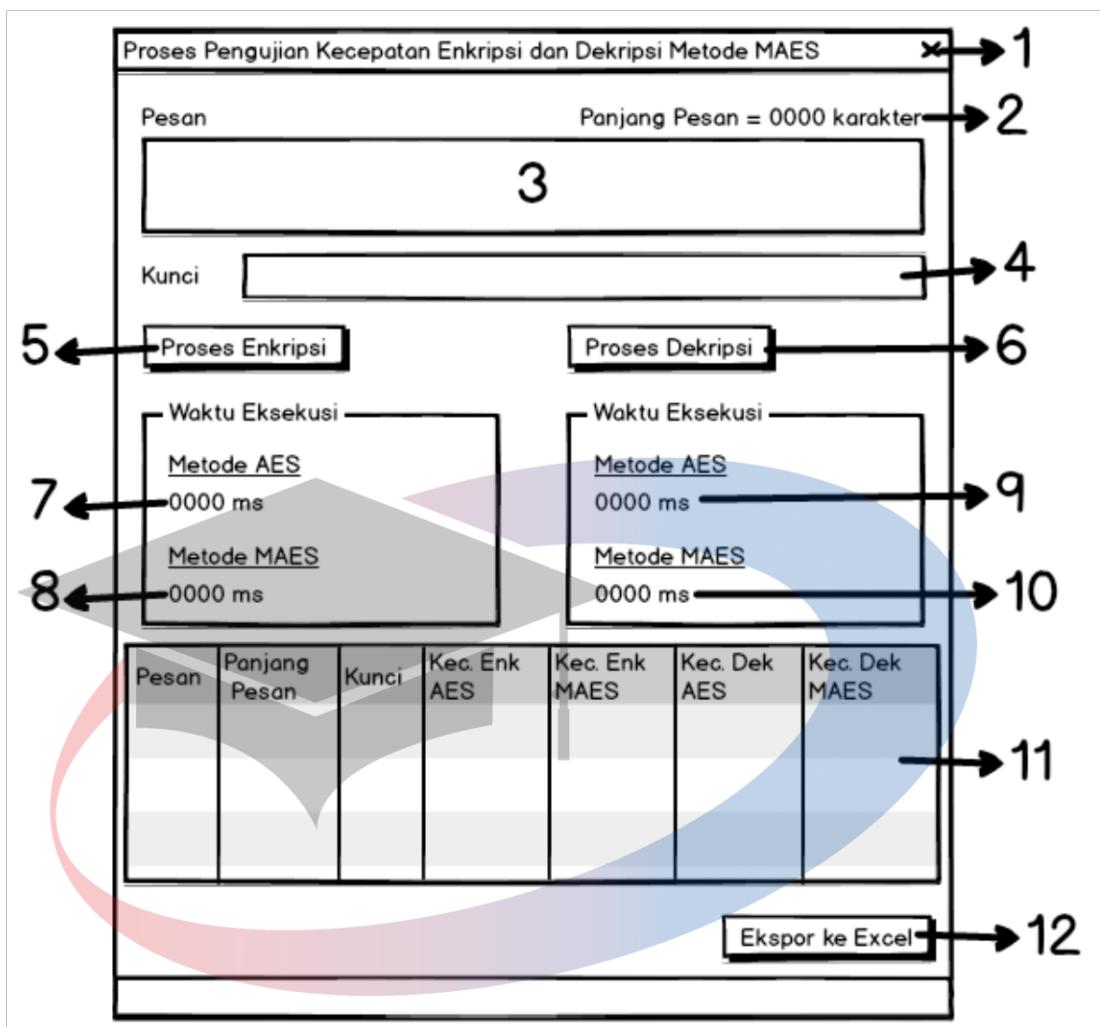
Gambar 3.21 Rancangan Form Pengujian Keamanan Kunci

Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.
2. *Textbox*, yang berfungsi sebagai tempat pengisian kunci pertama.
3. *Textbox*, yang berfungsi sebagai tempat pengisian kunci kedua.
4. Tombol, yang berfungsi untuk memulai proses pembentukan kunci dengan metode AES dan RK-AES.
5. *Textbox*, yang berfungsi untuk menampilkan sub kunci dari metode AES.
6. *Textbox*, yang berfungsi untuk menampilkan sub kunci dari metode AES.
7. *Textbox*, yang berfungsi untuk menampilkan sub kunci dari metode RK-AES.
8. *Textbox*, yang berfungsi untuk menampilkan sub kunci dari metode RK-AES.
9. *Textbox*, yang berfungsi untuk menampilkan hasil *avalanche effect* AES.
10. *Textbox*, yang berfungsi untuk menampilkan hasil *avalanche effect* RK-AES.

### 3.2.6 Form Pengujian Kecepatan Enkripsi dan Dekripsi

*Form Pengujian Kecepatan Enkripsi dan Dekripsi* berfungsi untuk melakukan pengujian kecepatan enkripsi dan dekripsi terhadap metode AES dan MAES. Rancangan *form* Pengujian Kecepatan Enkripsi dan Dekripsi dapat dilihat pada gambar berikut:



Gambar 3.22 Rancangan *Form* Pengujian Kecepatan Enkripsi dan Dekripsi

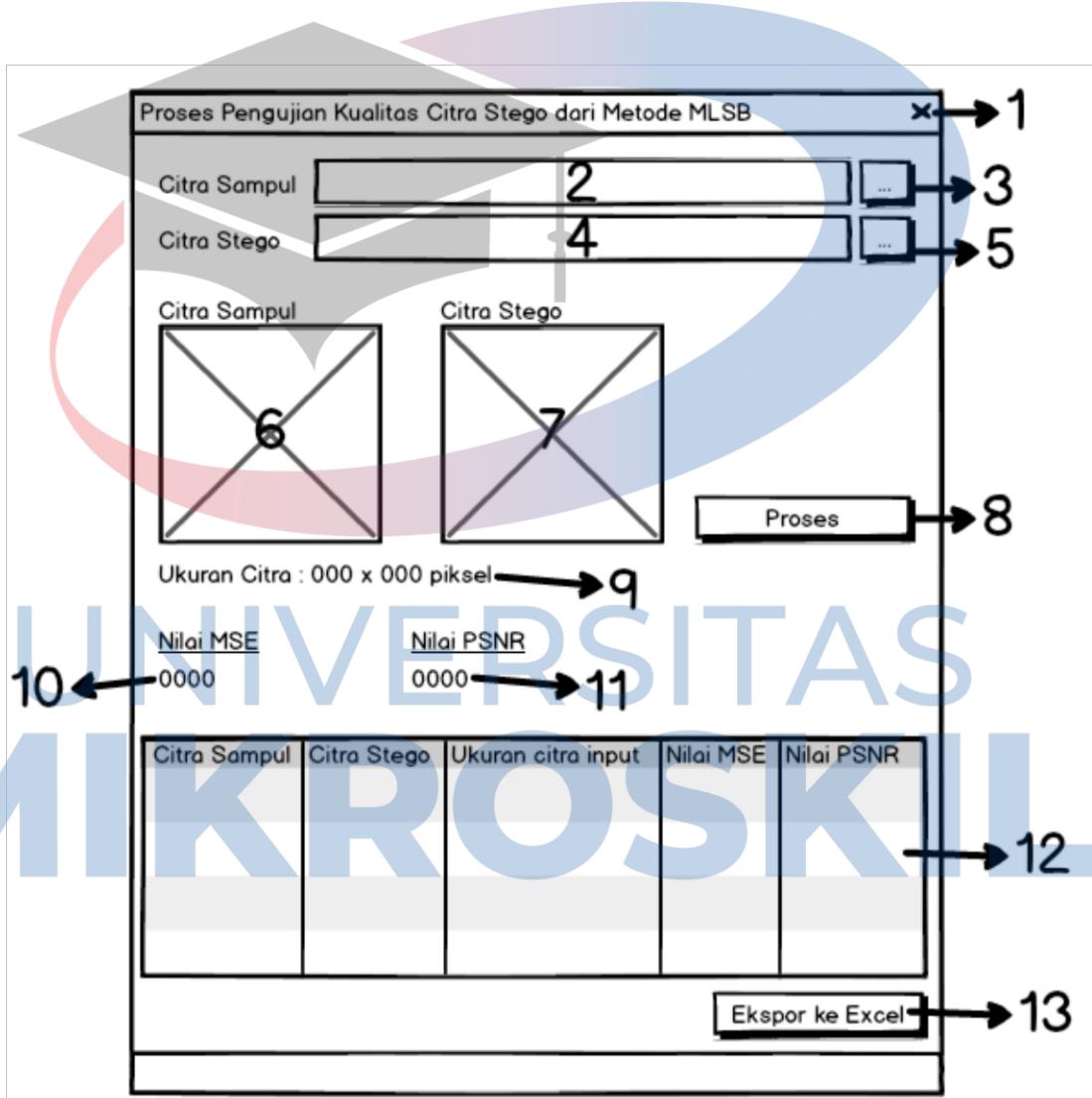
Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.
2. Label, yang berfungsi untuk menampilkan panjang karakter dari pesan yang dimasukkan.
3. Textbox, yang berfungsi sebagai tempat pengisian pesan.
4. Textbox, yang berfungsi sebagai tempat pengisian kunci.
5. Tombol, yang berfungsi untuk memulai proses enkripsi.
6. Tombol, yang berfungsi untuk memulai proses dekripsi.
7. Label, yang berfungsi untuk menampilkan lama waktu eksekusi dari proses enkripsi dengan metode AES.
8. Label, yang berfungsi untuk menampilkan lama waktu eksekusi dari proses enkripsi dengan metode MAES.
9. Label, yang berfungsi untuk menampilkan lama waktu eksekusi dari proses dekripsi dengan metode AES.

10. *Label*, yang berfungsi untuk menampilkan lama waktu eksekusi dari proses dekripsi dengan metode MAES.
11. Tabel hasil pengujian kecepatan waktu eksekusi.
12. Tombol, yang berfungsi untuk mengekspor hasil waktu eksekusi ke *file Microsoft Excel*.

### 3.2.7 Form Pengujian Kualitas Citra Stego

*Form Pengujian Kualitas Citra Stego* berfungsi untuk melakukan pengujian kualitas citra stego terhadap metode MLSB. Rancangan *form* Pengujian Kualitas Citra Stego dapat dilihat pada gambar berikut:



Gambar 3.23 Rancangan *Form Pengujian Kualitas Citra Stego*

Keterangan:

1. Tombol, yang berfungsi untuk menutup *form*.

2. *Textbox*, yang berfungsi untuk menampilkan ITabekokasi citra sampul.
3. Tombol, yang berfungsi untuk menampilkan kotak dialog *Open* untuk memilih *file* citra sampul.
4. *Textbox*, yang berfungsi untuk menampilkan lokasi citra stego.
5. Tombol, yang berfungsi untuk menampilkan kotak dialog *Open* untuk memilih *file* citra stego.
6. *Picturebox*, yang berfungsi untuk menampilkan citra sampul.
7. *Picturebox*, yang berfungsi untuk menampilkan citra stego.
8. Tombol, yang berfungsi untuk memulai proses pengujian.
9. *Label*, yang berfungsi untuk menampilkan ukuran citra yang dimasukkan.
10. *Label*, yang berfungsi untuk menampilkan nilai MSE yang diperoleh.
11. *Label*, yang berfungsi untuk menampilkan nilai PSNR yang diperoleh.
12. Tabel hasil pengujian kualitas citra stego.
13. Tombol, yang berfungsi untuk mengekspor hasil pengujian kualitas citra stego ke *file* Microsoft Excel.

