**BAB III**

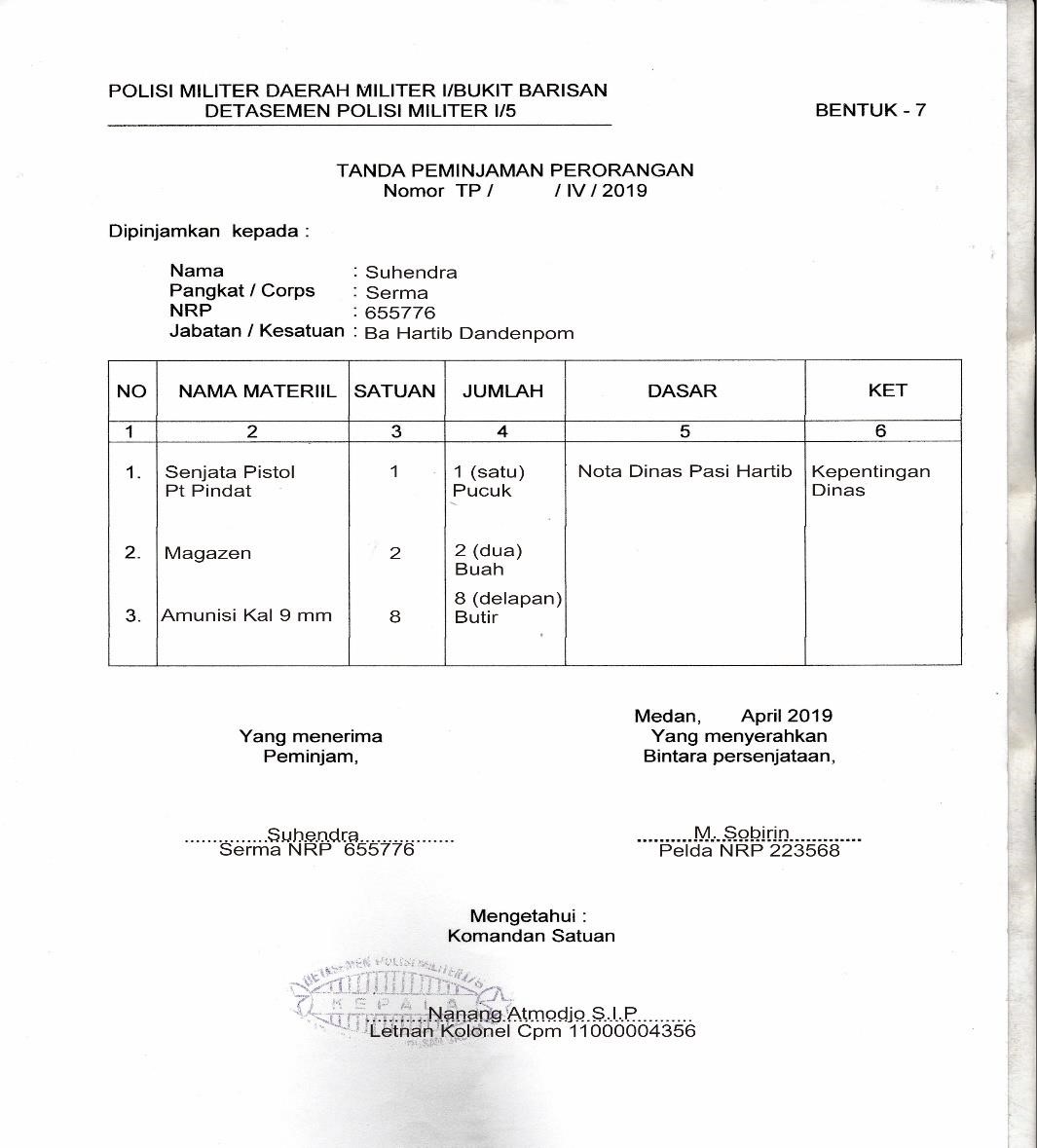
**METODOLOGI PENELITIAN**

## 3.1 Metodologi Penelitian

Penelitian yang baik harus berdasarkan dengan metodologi yang baik pula. Berikut ini adalah metodologi dalam penelitian ini, yaitu:

1. *Data Collecting* (Pengumpulan Data)

Ada 2 teknik yang digunakan dalam proses pengumpulan data yaitu: observasi dan wawancara. Dalam penelitian ini dilakukan observasi pra-riset telebih dahulu untuk mencari masalah yang terjadi di Detasemen Polisi Militer I/5 Medan terkhusus dalam pengamanan data peminjaman senjata api, dari masalah tersebut masalah akan dirumusakan dalam penelitian ini sehingga dapat menemukan rumusan apa saja yang perlu dipersiapkan untuk bagaimana cara menyelesaikan masalah tersebut. Sedangkan dalam mendapatkan data yang baik, dalam penelitian ini dilakukan wawancara kepada Pasi TUUD (perwira seksi tata usaha urusan dalam), atau pihak-pihak yang terlibat dalam mendukung penelitian ini. Selain itu juga, penelitian mencoba mencari data sekunder dengan melakukan surfing di mesin pencarian terkait hal-hal penting dalam pengamanan data peminjaman senjata api.



Gambar 3.1 Contoh Surat Peminjaman Senjata Api Perorangan

1. *Study of Literature* (Studi Kepustakaan)

Studi *literature* merupakan hal terpenting dalam penelitian dikarenakan merupakan sebuah rujukan terkait bagaimana masalah ini dibahas oleh orangorang kebanyakan. Dalam penelitian ini studi kepustakaannya berasal dari bukubuku dan jurnal serta sumber lainnya yang mendukung untuk menjelaskan terkait masalah yang dibahas dan referensi dari metode yang digunakan.

## 3.2 Metode Perancangan Sistem

Dalam konsep penulisan metode perancangan sistem merupakan salah satu unsur penting dalam penelitian. Dalam metode perancangan sistem khususnya software atau perangkat lunak kita dapat mengadopsi beberapa metode di antaranya algoritma *waterfall* atau algoritma air terjun. Berikut ini adalah contoh penulisan Metode Perancangan Sistem.

Di dalam penelitian ini, di adopsi sebuah metode perancangan sistem yaitu *waterfall algorithm*. Berikut ini adalah fase yang dilakukan dalam penelitian ini yaitu:

1. Analisis Masalah Dan Kebutuhan

Dalam mengetahui masalah yang terjadi dilapangan, perlu di lakukan analisa masalah yang terjadi dalam keamanan data peminjaman senjata, serta menganalisa apaapa saja yang kebutuhan terhadap sistem yang akan di rancang nantinya.

1. Desain Sistem

Dalam fase ini dibagi beberapa indikator atau elemen yaitu: (1) pemodelan sistem dengan menggunakan UML (*Unified Modelling Language)*, (2) pemodelan dengan menggunakan *flowchart system*, (3) desain *input*, dan (4) desain *output* dari sistem keamanan yang akan dirancang dalam pemecahan masalah pengamanan data peminjaman senjata api di Detasemen Polisi Militer I/5 Medan.

1. Pembangun Sistem

Fase ini menjelaskan tentang bagaimana melakukan pengkodingan terhadap desain sistem yang dirancang baik dari sistem *input*, proses dan *output* menggunakan pemrograman *Microsoft Visual Studio* 2019 dalam kodingnya.

1. Uji Coba Sistem

Fase ini merupakan fase terpenting untuk pembangunan sistem keamanan. Hal ini dikarenakan pada fase ini akan dilakukan untuk mencari kendala atau masalah-masalah yang terjadi saat sistem sedang berjalan.

## 3.3 Algoritma Sistem

*Advance Encryption Standard* merupakan satu dari banyak algoritma kriptografi. Advance Encryption Standard menggunakan 4 tahapan dalam proses penyandian data, antara lain adalah *SubBytes, ShiftRows, MixColumns* dan *AddRoundKey.* Proses penyandian dilakukan berulang atau bisa disebut dengan ronde. Banyaknya ronde pada Advance Encryption Standard dengan panjang kunci 128 bit adalah 10 ronde. Plaintext yang akan disandikan akan diurutkan dan dimasukkan ke dalam state 4 x 4. Plaintext yang telah dimasukkan ke dalam state akan diproses 4 kali transformasi dan akan di XOR-kan dengan masing-masing kunci yang berbeda setiap rondenya (*RoundKey*).

**3.3.1** ***Flowchart* dari Metode Penyelesaian**

Berikut ini adalah *flowchart* dari proses enkripsi dan dekripsi dari algoritma AES yaitu sebagai berikut:



Gambar 3.2 *Flowchart* Proses Enkripsi AES



Gambar 3.3 *Flowchart* Proses Deripsi AES

### 3.3.2 Deskripsi Data Dari Penelitian

Berikut ini adalah data peminjaman senjata api di Detasemen Polisi Militer I/5 Medan, yang akan diamankan. Dalam pengujiannya, sebagai contoh data yang digunakan sebagai sampel dalam penelitian ini yaitu sebagai berikut:

#### 3.3.2.1 Proses Enkripsi Algoritma AES

##### 1. Proses Ekspansi Kunci

Kunci ronde (*round key*) dibutuhkan untuk enkripsi dan dekripsi pada algoritma *Advanced Encryption Standard.* Maksimal panjang kunci adalah sebanyak 16 digit dan jumlah kunci ronde yang diperlukan adalah 10 kunci yang akan diperoleh dari proses *ekspansi* kunci. Pada kasus ini, kunci yang akan digunakan yaitu “yoga arif wibowo”

1. Urutkan kunci kedalam blok berukuran 128 bit (16 kode ASCII). Lalu ubah kunci kedalam bentuk heksadecimal.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| y | o | g | a |  | a | r | i | f |  | w | i | b | o | w | o |
| 79 | 6F | 67 | 61 | 20 | 61 | 72 | 69 | 66 | 20 | 77 | 69 | 62 | 6F | 77 | 6F |

1. Langkah selanjutnya yaitu susun kunci ke dalam state berukuran 4 x 4 seperti dibawah ini :

|  |  |  |  |
| --- | --- | --- | --- |
| *RoundKey* Ke-1 | | | |
| D0 | F0 | 96 | F4 |
| 9A | FB | DB | B4 |
| CF | BD | CA | BD |
| CB | A2 | CB | A4 |

1. Kemudian, lakukan fungsi *RotWord,* yaitu dengan menggeser setiap bit pada kolom ke-4 ke atas dari kunci ke-0.

|  |
| --- |
| 6F |
| 77 |
| 6F |
| 62 |

|  |
| --- |
| 62 |
| 6F |
| 77 |
| 6F |

*RotWord*

1. Hasil dari *RotWord*  disubstitusikan dengan nilai pada table S-*Box* (*SubBytes*).

|  |
| --- |
| A8 |
| F5 |
| A8 |
| AA |

|  |
| --- |
| 6F |
| 77 |
| 6F |
| 62 |

1. Tahap akhir yaitu dengan melakukan proses XOR antar kolom pertama dari kunci ronde ke-0 dan hasil dari *SubBytes* lalu di-XOR-kan lagi dengen *RCon* Kolom pertama (wi) pada kunci ronde selanjutnya (ronde ke-1) = K1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8F |  | 54 |  | 01 | = | 2A |
| F9 | 72 | 00 | 8B |
| EF | 69 | 00 | 86 |
| 4D | 67 | 00 | 2A |

1. Untuk mendapatkan kolom kedua. Diperoleh dengan proses XOR antara *Wi* dengan kolom kedua dari kunci ronde ke-0. Sedangkan untuk mendapatkan kolom ketiga dan keempat kunci ronde ke-1, dilakukan proses seperti memperoleh kolom kedua.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D0 |  | 20 | = | F0 |  | F0 |  | 66 | = | 96 |  | 96 |  | 62 | = | F4 |
| 9A | 61 | FB | FB | 20 | DB | DB | 6F | B4 |
| CF | 72 | BD | BD | 77 | CA | CA | 77 | BD |
| CB | 69 | A2 | A2 | 69 | CB | CB | 6F | A4 |

1. Dari seluruh proses diatas, maka diperoleh kunci untuk ronde ke-1 yaitu:

|  |  |  |  |
| --- | --- | --- | --- |
| D0 | F0 | 96 | F4 |
| 9A | FB | DB | B4 |
| CF | BD | CA | BD |
| CB | A2 | CB | A4 |

Untuk mendapatkan kunci ronde ke-2 sampai ke-10, proses diatas diulang smpai 10 kali. Kunci masing-masing ronde akan digunakan saat proses enkripsi dan dekripsi. Dibawah ini adalah hasil *ekspansi*  kunci hingga ronde ke-10:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-1 | | | |  | *RoundKey* Ke-2 | | | |  | *RoundKey* Ke-3 | | | |
| D0 | F0 | 96 | F4 | 5F | AF | 39 | CD | C9 | 66 | 5F | 92 |
| 9A | FB | DB | B4 | E0 | 1B | C0 | 74 | C9 | D2 | 12 | 66 |
| CF | BD | CA | BD | 86 | 3B | F1 | 4C | D0 | EB | 1A | 56 |
| CB | A2 | CB | A4 | 74 | D6 | 1D | B9 | C9 | 1F | 2 | BB |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-4 | | | | | | |  | | *RoundKey* Ke-5 | | | | | | | |  | | *RoundKey* Ke-6 | | | | | |
| F2 | 94 | | CB | | 59 | | FF | | 6B | | A0 | | F9 | | 58 | | 33 | | 93 | 6A |
| 78 | AA | | B8 | | DE | | 26 | | 8C | | 34 | | EA | | 41 | | CD | | F9 | 13 |
| 3A | D1 | | CB | | 9D | | 8D | | 5C | | 97 | | A | | 25 | | 79 | | EE | E4 |
| 86 | 99 | | 9B | | 20 | | 4D | | D4 | | 4F | | 6F | | D4 | | 0 | | 4F | 20 |
| *RoundKey* Ke-7 | | | | | |  | | *RoundKey* Ke-8 | | | | | | | |  | | *RoundKey* Ke-9 | | | | | | |
| 65 | 56 | C5 | | AF | | 93 | | C5 | | 0 | | AF | | 7F | | BA | | BA | | 15 |
| 28 | E5 | 1C | | F | | D0 | | 35 | | 29 | | 26 | | CF | | FA | | D3 | | F5 |
| 92 | EB | 5 | | E1 | | C4 | | 2F | | 2A | | CB | | F | | 20 | | A | | C1 |
| D6 | D6 | 99 | | B9 | | AF | | 79 | | E0 | | 59 | | D6 | | AF | | 4F | | 16 |

|  |  |  |  |
| --- | --- | --- | --- |
| *RoundKey* Ke-10 | | | |
| AF | 15 | AF | BA |
| B7 | 4D | 9E | 6B |
| 48 | 68 | 62 | A3 |
| 8F | 20 | 6F |  |

##### 2. Proses Enkirpsi

*Plaintext* yang akan digunakan yaitu “Suhendra\_0655776”. Kemudian urutkan kedalam blok lalu ubah kedalam bilangan hexsadesimal.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | u | h | e | n | d | r | a | \_ | 0 | 6 | 5 | 5 | 7 | 7 | 6 |
| 53 | 75 | 68 | 65 | 6E | 64 | 72 | 61 | 5F | 30 | 36 | 35 | 35 | 37 | 37 | 36 |

Susun 16 *byte* pertama dari *plaintext* yang telah diubah kedalam *state* 4x4:

|  |  |  |  |
| --- | --- | --- | --- |
| 53 | 6E | 5F | 35 |
| 75 | 64 | 30 | 37 |
| 68 | 72 | 36 | 37 |
| 65 | 61 | 35 | 36 |

1. Lakukan XOR antara *plainteks* dengan *RoundKey* 0*.* Proses ini dinamakan *AddRoundKey*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 53 | 6E | 5F | 35 |  | 79 | 20 | 66 | 62 | = | 2A | 4E | 39 | 57 |
| 75 | 64 | 30 | 37 | 6F | 61 | 20 | 6F | 1A | 5 | 10 | 58 |
| 68 | 72 | 36 | 37 | 67 | 72 | 77 | 77 | F | 0 | 41 | 40 |
| 65 | 61 | 35 | 36 | 61 | 69 | 69 | 6F | 4 | 8 | 5C | 59 |

Proses *AddRoundKey* diatas masih sebagai *pra-round* dan akan menjadi masukan untuk ronde ke1 yang akan diproses dengan 4 tranformasi yaitu *SubByte, ShifRows, MixColumns* dan *AddRoundKey.*

1. Hasil di *pra-round* disubtitusikan dengan nilai pada table *S-Box* (*SubBytes*).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2A | 4E | 39 | 57 |  | E5 | 2F | 12 | 5B |
| 1A | 5 | 10 | 58 | A2 | 6B | CA | 6A |
| F | 0 | 41 | 40 | 76 | 63 | 83 | 9 |
| 4 | 8 | 5C | 59 | F2 | 30 | 4A | CB |

1. Lakukan *ShiftRows* pada hasil dari subtitusi *SubBytes* yang dieksekusi lewat pergeseran *siklik* secara memutar dengan geseran yang acak pada tiga baris terakhir *state* (baris pertama, r = 0, tidak digeser). Baris ke dua digeser secara *siklik* ke kiri sekali, baris ke tiga dua kali, dan baris ke empat tiga kali.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E5 | 2F | 12 | 5B |  | E5 | 2F | 12 | 5B |
| 2A | 6B | CA | 6A | 6B | CA | 6C | A2 |
| 76 | 63 | 83 | 9 | 83 | 9 | 76 | 63 |
| F2 | 30 | 4A | CB | CB | F2 | 30 | 4A |

3. Transformasi *MixColumns* dengan mengoprasikan *state* kolom demi kolom pada *state,* dengan mengkonversikan setiap kolom poliniminal.

*MixColumns*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *S0.1* |  | |  |  | *S01.1* |  | |
| *S0.0* | *S1.1* | *S0.2* | *S0.3* | *S01.0* | *S11.1* | *S02.2* | *S01.3* |
| *S1.0* | *S2.1* | *S1.2* | *S1.3* | *S11.0* | *S21.2* | *S11.2* | *S11.3* |
| *S2.0* | *S3.1* | *S2.2* | *S2.3* | *S21.0* | *S31.1* | *S21.2* | *S21.3* |
| *S3.0* |  | *S3.2* | *S3.3* | *S31.0* |  | *S31.2* | *S31.3* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |  | *S0.1* | = | *S01.1* |  |
| 01 | 02 | 03 | 01 | *S1.1* | *S11.1* |
| 01 | 01 | 02 | 03 | *S2.1* | *S21.2* |
| 03 | 01 | 01 | 02 | *S3.1* | *S31.1* |

𝑆01.1 = ([02]. 𝑆0.1) 𝑥𝑜𝑟 ([03]. 𝑆1.1) 𝑥𝑜𝑟 ([01]. 𝑆2.1) 𝑥𝑜𝑟 ([01]. 𝑆3.1)

= ([02]. 𝑆0.1) = (02). (5B)

= ([02]. 𝑆0.1) = (10). (10010010)

= ([02]. 𝑆0.1) = (𝑥). (𝑥7 + 𝑥4 + 𝑥)

= ([02]. 𝑆0.1) = (𝑥8 + 𝑥5 + 𝑥2 + 𝑥8 + 𝑥4 + 𝑥3 + 𝑥 + 1 )

= ([02]. 𝑆0.1) = (𝑥5 + 𝑥4 + 𝑥3 + 𝑥2 + 𝑥 + 1 )

= ([02]. 𝑆0.1) = 00111111

= ([03]. 𝑆1.1) = (03). (F9)

= ([03]. 𝑆1.1) = (11). (11111001)

= ([03]. 𝑆1.1) = (𝑥 + 1). (𝑥7 + 𝑥6 + 𝑥5 + 𝑥4 + 𝑥3 + 1)

= ([03]. 𝑆1.1) = 𝑥8 + 𝑥7 + 𝑥6 + 𝑥5 + 𝑥4 + 𝑥 + 𝑥7 + 𝑥6 + 𝑥5 +

𝑥4 + 𝑥3 + 1

= ([03]. 𝑆1.1) = 𝑥8 + 𝑥3 + 1 + 𝑥8 + 𝑥4 + 𝑥3 + 𝑥 + 1

= ([03]. 𝑆1.1) = 𝑥4 + 𝑥 + 1

= ([03]. 𝑆1.1) = 00010011

= ([01]. 𝑆2.1) = (01). (05) = 00000101

= ([01]. 𝑆3.1) = (01). (B7) = 10110111

𝑆01.0 = 00111111 𝑥𝑜𝑟 00010011 𝑥𝑜𝑟 00000101 𝑥𝑜𝑟 10110111

𝑆01.0 = 10011101 = 9D

𝑆11.0 = ([01]. 𝑆0.1) 𝑥𝑜𝑟 ([02]. 𝑆1.1) 𝑥𝑜𝑟 ([03]. 𝑆2.1) 𝑥𝑜𝑟 ([01]. 𝑆3.1)

𝑆11.0 = 11000011 = C3

𝑆21.0 = ([01]. 𝑆0.2) 𝑥𝑜𝑟 ([01]. 𝑆1.2) 𝑥𝑜𝑟 ([02]. 𝑆2.2) 𝑥𝑜𝑟 ([03]. 𝑆3.2)

𝑆21.0 = 10100011 = A3

𝑆31.0 = ([03]. 𝑆0.3) 𝑥𝑜𝑟 ([01]. 𝑆1.3) 𝑥𝑜𝑟 ([01]. 𝑆2.3) 𝑥𝑜𝑟 ([02]. 𝑆3.3) 𝑆31.0 = 00100100 = 24

Lakukan perulangan seperti yang diatas, hingga didapatkan hasil *MixColumns* seperti sebagai berikut.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E5 | 2F | 12 | 5B |  | 24 | E0 | DC | 62 |
| 6B | CA | 6A | A2 | 66 | 49 | 6C | EB |
| 83 | 9 | 76 | 63 | D5 | FA | C4 | E1 |
| CB | F2 | 30 | 4A | 51 | 4D | 4A | B8 |

4. Langkah terakhir untuk mendapat enkripsi putaran pertama, lakukan XOR antara hasil *MixColumns* dengan *RoundKey* Ke-1, proses ini disebut *AddRoundKey.*

*MixColumns RoundKey* Ke*-*1 *AddRoundKey*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24 | E0 | DC | 62 |  | D0 | F0 | 96 | F4 | **=** | F4 | 10 | 4A | 96 |
| 66 | 49 | 6C | EB | 9A | FB | DB | B4 | FC | B2 | B7 | 5F |
| D5 | FA | C4 | E1 | CF | BD | CA | BD | 1A | 47 | E | 5C |
| 51 | 4D | 4A | B8 | CB | A2 | CB | A4 | 9A | EF | 81 | 1C |

Lakukan proses diatas sampai 10 kali putaran (*round*). Berikut adalah hasil enkripsi hingga *round* ke 10:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-1 | | | |  | | *RoundKey* Ke-2 | | | | |  | | *RoundKey* Ke-3 | | | | |
| D0 | F0 | 96 | F4 | |  | | 5F | AF | 39 | CD | |  | | C9 | 66 | 5F | 92 | |
| 9A | FB | DB | B4 | | E0 | 1B | C0 | 74 | | C9 | D2 | 12 | 66 | |
| CF | BD | CA | BD | | 86 | 3B | F1 | 4C | | D0 | EB | 1A | 56 | |
| CB | A2 | CB | A4 | | 74 | D6 | 1D | B9 | | C9 | 1F | 2 | BB | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-4 | | | |  | *RoundKey* Ke-5 | | | |  | *RoundKey* Ke-6 | | | |
| F2 | 94 | CB | 59 | FF | 6B | A0 | F9 | 58 | 33 | 93 | 6A |
| 78 | AA | B8 | DE | 26 | 8C | 34 | EA | 41 | CD | F9 | 13 |
| 3A | D1 | CB | 9D | 8D | 5C | 97 | A | 25 | 79 | EE | E4 |
| 86 | 99 | 9B | 20 | 4D | D4 | 4F | 6F | D4 | 0 | 4F | 20 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-7 | | | |  | *RoundKey* Ke-8 | | | |  | *RoundKey* Ke-9 | | | |
| 65 | 56 | C5 | AF | 93 | C5 | 0 | AF | 7F | BA | BA | 15 |
| 28 | E5 | 1C | F | D0 | 35 | 29 | 26 | CF | FA | D3 | F5 |
| 92 | EB | 5 | E1 | C4 | 2F | 2A | CB | F | 20 | A | C1 |
| D6 | D6 | 99 | B9 | AF | 79 | E0 | 59 | D6 | AF | 4F | 16 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Round* Ke-10 | | | |
| AF | 15 | AF | BA |
| B7 | 4D | 9E | 6B |
| 48 | 68 | 62 | A3 |
| 8F | 20 | 6F | 79 |

Hasil dari proses *AddRounKey* atau *round* ke-10 diubah ke bentuk karakter dalam table ASCII

|  |  |  |
| --- | --- | --- |
| ***Round*** | **Kode ASCII** | **Karakter** |
| 41 | 65 | A |
| 97 | 151 | — |
| 68 | 104 | h |
| 3A | 58 | : |
| CE | 206 | Î |
| 11 | 17 | [] |
| 69 | 105 | i |
| D7 | 215 | × |
| 45 | 69 | E |
| 67 | 103 | g |
| D5 | 213 | Õ |
| 40 | 64 | @ |
| B | 11 | [] |
| 7 | 7 | [] |
| 5F | 95 | \_ |
| 89 | 137 | ‰ |

Dan hasil enkripsi AES menghasilkan cipherteks “A — h : Î [] i × E g Õ @   
 [] [] \_ ‰”.

**3.3.2.2 Proses Dekripsi Algoritma AES**

*Ciphertext* yang akan digunakan yaitu “A — h : Î [] i × E g Õ @ [] [] \_ ‰”. Urutkan kedalam sebuah blok dan ubah ke bilangan heksadesimal.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | — | h | : | Î | [] | i | × | E | g | Õ | @ | [] | [] | \_ | ‰ |
| 41 | 97 | 68 | 3A | CE | 11 | 69 | D7 | 45 | 67 | D5 | 40 | B | 7 | 5F | 89 |

Kemudian susun 16 *byte* pertama dari *ciphertext* yang telah diubah ke bentuk heksadesimal kedalam *state* 4x4:

|  |  |  |  |
| --- | --- | --- | --- |
| 41 | CE | 45 | B |
| 97 | 11 | 67 | 7 |
| 68 | 69 | D5 | 5F |
| 3A | D7 | 40 | 89 |

Lakukan XOR antara *ciphertext* dengan *RoundKey*  Ke-10. Proses ini dinamakan *AddRoundKey.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 41 | CE | 45 | B |  | AF | 15 | AF | BA | = | EE | DB | EA | B1 |
| 97 | 11 | 67 | 7 | B7 | 4D | 9E | 6B | 20 | 5C | F9 | 6C |
| 68 | 69 | D5 | 5F | 48 | 68 | 62 | A3 | 20 | 1 | B7 | FC |
| 3A | D7 | 40 | 89 | 8F | 20 | 6F | 79 | B5 | F7 | 2F | F0 |

Proses *AddRoundKey* diatas masih dalam *initial-round,* dan akan menjadi masukan untuk round ke-1 yang akan diperoses dengan 4 transformasi yaitu *InvShifRows, InvSubBytes, AddRoundKey,* dan *InvMixColumns.* Kunci yang digunakan untuk dekripsi sama dengan proses enkripsi. Berikut adalah proses dekripsi dari hasil *ciphertext* yang diperoleh dari proses enkripsi sebelumnya.

1. Lakukan *InvShifRows* pada hasil *initial-round* dari *AddRoundKey* yang dieksekusi lewat pergeseran *siklik* secara memutar dengan geseran yang acak pada tiga baris terakhir *state* (baris pertama, *r* = 0, tidak digeser). Beris ke dua digeser secara *sikilik*  ke kiri tiga kali, baris ke tiga dua kali, baris ke empat sekali.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EE | DB | EA | B1 |  | EE | DB | EA | B1 |
| 20 | 5C | F9 | 6C | 6C | 20 | 5C | F9 |
| 20 | 1 | B7 | FC | B7 | FC | 20 | A |
| B5 | F7 | 2F | F0 | F7 | 2F | F0 | B5 |

1. Hasil dari *InvShifRows* disubtitusikan dengan nilai pada table *S-B -1 (InvSubBytes).*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EE | DB | EA | B1 |  | 99 | 9F | BB | 56 |
| 6C | 20 | 5C | F9 | B8 | 54 | A7 | 69 |
| B7 | FC | 20 | 1 | 20 | 55 | 54 | 9 |
| F7 | 2F | F0 | B5 | 26 | 4E | 17 | D2 |

1. XOR kan hasil dari *InvSubBytes* dengan *RoundKey* Ke-9. Proses ini disebut *AddRoundKey.*

|  |  |  |  |
| --- | --- | --- | --- |
| E6 | 25 | 1 | 43 |
| 77 | AE | 74 | 9C |
| 2F | 75 | 5E | C8 |
| F0 | E1 | 58 | C4 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 99 | 9F | BB | 56 | ⊕ | 7F | BA | BA | 15 |
| B8 | 54 | A7 | 69 | CF | FA | D3 | F5 |
| 20 | 55 | 54 | 9 | F | 20 | A | C1 |
| 26 | 4E | 17 | D2 | D6 | AF | 4F | 16 |

=

1. Hasil dari *AddInvRoundKey* ditransformasikan oleh *InvMixColumns* dengan mengoprasikan *stare* kolom demi kolom. Oprasi ini dilakukan pada *state* kolom, dengan mengkonversikan setiap kolom sebagai polinominal.

*MixColumns*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *S0.1* |  | |  |  | *S01.1* |  | |
| *S0.0* | *S1.1* | *S0.2* | *S0.3* | *S01.0* | *S11.1* | *S02.2* | *S01.3* |
| *S1.0* | *S2.1* | *S1.2* | *S1.3* | *S11.0* | *S21.2* | *S11.2* | *S11.3* |
| *S2.0* | *S3.1* | *S2.2* | *S2.3* | *S21.0* | *S31.1* | *S21.2* | *S21.3* |
| *S3.0* |  | *S3.2* | *S3.3* | *S31.0* |  | *S31.2* | *S31.3* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |  | *S0.1* | = | *S01.1* |  |
| 01 | 02 | 03 | 01 | *S1.1* | *S11.1* |
| 01 | 01 | 02 | 03 | *S2.1* | *S21.2* |
| 03 | 01 | 01 | 02 | *S3.1* | *S31.1* |

𝑆01.1 = ([0E]. 𝑆0.1) 𝑥𝑜𝑟 ([0B]. 𝑆1.1) 𝑥𝑜𝑟 ([0D]. 𝑆2.1) 𝑥𝑜𝑟 ([09]. 𝑆3.1)

= ([0E]. 𝑆0.1) = (0D). (5B)

= ([0E]. 𝑆0.1) = (1101). (01011011)

= ([0E]. 𝑆0.1) = (𝑥3+ 𝑥2 + 1). (𝑥6+ 𝑥4+ 𝑥3 + 𝑥 + 1)

= ([0E]. 𝑆0.1) = 𝑥7 + 𝑥6 + 𝑥5 + 𝑥4 + 𝑥2 + 𝑥

= ([0E]. 𝑆0.1) = 11110110

= ([0B]. 𝑆1.1) = (0B). (74)

= ([0B]. 𝑆1.1) = (1011). (01110100)

= ([0B]. 𝑆1.1) = (𝑥3 + 𝑥 + 1). (𝑥6 + 𝑥5 + 𝑥4 + 𝑥2)

= ([0B]. 𝑆1.1) = 𝑥7+ 𝑥5 + 𝑥3 + 𝑥 + 1

= ([0B]. 𝑆1.1) = 10101110

= ([0D]. 𝑆2.1) = (0D). (D7)

= ([0D]. 𝑆2.1) = (1101). (11010111)

= ([0D]. 𝑆2.1) = (𝑥3 + 𝑥2 + 1). (𝑥7 + 𝑥6 + 𝑥4 + 𝑥2 + 𝑥 + 1)

= ([0D]. 𝑆2.1) = 𝑥7+ 𝑥5 + 𝑥3 + 𝑥 + 1

= ([0D]. 𝑆2.1) = 10101010

= ([09]. 𝑆3.1) = (09). (E5)

= ([09]. 𝑆3.1) = (1001). (11100101)

= ([09]. 𝑆3.1) = (𝑥3 + 1). (𝑥7 + 𝑥6 + 𝑥5 + 𝑥2 + 1)

= ([09]. 𝑆3.1) = 𝑥7+ 𝑥5 + 𝑥3 + 𝑥 + 1

= ([09]. 𝑆3.1) = 10101010

𝑆01.0 = 01111111 𝑥𝑜𝑟 10111110 𝑥𝑜𝑟 00111011 𝑥𝑜𝑟 10100110

𝑆01.0 = 11110110 = F6

𝑆11.0 = ([09]. 𝑆0.1) 𝑥𝑜𝑟 ([0E]. 𝑆1.1) 𝑥𝑜𝑟 ([0B]. 𝑆2.1) 𝑥𝑜𝑟 ([0D]. 𝑆3.1)

𝑆01.0 = 10101110 = AE

𝑆21.0 = ([0D]. 𝑆0.2) 𝑥𝑜𝑟 ([09]. 𝑆1.2) 𝑥𝑜𝑟 ([0E]. 𝑆2.2) 𝑥𝑜𝑟 ([0B]. 𝑆3.2)

𝑆01.0 = 01000000 = 40

𝑆31.0 = ([0B]. 𝑆0.3) 𝑥𝑜𝑟 ([0D]. 𝑆1.3) 𝑥𝑜𝑟 ([09]. 𝑆2.3) 𝑥𝑜𝑟 ([0E]. 𝑆3.3)

𝑆01.0 = 000000101 = 05

Lakukan perulangan seperti yang diatas, hingga didapatkan hasil *InvMixColumns* seperti sebagai berikut.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E6 | 25 | 1 | 43 |  | DE | 94 | 4A | ED |
| 77 | AE | 74 | 9C | E5 | C3 | 56 | 5E |
| 2F | 75 | 5E | C8 | 8B | FA | F3 | DB |
| F0 | E1 | 58 | C4 | FE | B2 | 9C | BB |

Proses diatas diulang sampai 10 kali putaran (*round*). Berikut adalah hasil dari dekripsi hingga *round* ke 10:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-1 | | | |  | | *RoundKey* Ke-2 | | | | |  | | *RoundKey* Ke-3 | | | | |
| 7F | BA | BA | 15 | |  | | 93 | C5 | 0 | AF | |  | | 65 | 56 | C5 | AF | |
| CF | FA | D3 | F5 | | D0 | 35 | 29 | 26 | | 28 | E5 | 1C | F | |
| F | 20 | A | C1 | | C4 | 2F | 2A | CB | | 92 | EB | 5 | E1 | |
| D6 | AF | 4F | 16 | | AF | 79 | E0 | 59 | | D6 | D6 | 99 | B9 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-4 | | | |  | | *RoundKey* Ke-5 | | | | |  | | *RoundKey* Ke-6 | | | | |
| 58 | 33 | 93 | 6A | |  | | FF | 6B | A0 | F9 | |  | | F2 | 94 | CB | 59 | |
| 41 | CD | F9 | 13 | | 26 | 8C | 34 | EA | | 78 | AA | B8 | DE | |
| 25 | 79 | EE | E4 | | 8D | 5C | 97 | A | | 3A | D1 | CB | 9D | |
| D4 | 0 | 4F | 20 | | 4D | D4 | 4F | 6F | | 86 | 99 | 9B | 20 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *RoundKey* Ke-7 | | | |  | | *RoundKey* Ke-8 | | | | |  | | *RoundKey* Ke-9 | | | | |
| C9 | 66 | 5F | 92 | |  | | 5F | AF | 39 | CD | |  | | D0 | F0 | 96 | F4 | |
| C9 | D2 | 12 | 66 | | E0 | 1B | C0 | 74 | | 9A | FB | DB | B4 | |
| D0 | EB | 1A | 56 | | 86 | 3B | F1 | 4C | | CF | BD | CA | BD | |
| C9 | 1F | 2 | BB | | 74 | D6 | 1D | B9 | | CB | A2 | CB | A4 | |

Khusus round ke-10 transformasi *InvMixColumns* tidak dilakukan, cukup hanya tranformasi *InvShifRow, InvSubBytes,* dan *AddInvRoundKey.*

1. *InvShifRows*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E5 | 2F | 12 | 5B |  | E5 | 2F | 12 | 5B |
| 6B | CA | 6A | A2 | A2 | 6B | CA | 6A |
| 83 | 9 | 76 | 63 | 76 | 63 | 83 | 9 |
| CB | F2 | 30 | 4A | F2 | 30 | 4A | CB |

1. *InvSubBytes*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E5 | 2F | 12 | 5B |  | 2A | 4E | 39 | 57 |
| A2 | 6B | CA | 6A | 1A | 5 | 10 | 58 |
| 76 | 63 | 83 | 9 | 0F | 0 | 41 | 40 |
| F2 | 30 | 4A | CB | 4 | 8 | 5C | 59 |

1. *AddInvRoundKey*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2A | 4E | 39 | 57 |  | 79 | 20 | 66 | 62 | = | 53 | 6E | 5F | 35 |
| 1A | 5 | 10 | 58 | 6F | 61 | 20 | 6F | 75 | 64 | 30 | 37 |
| 0F | 0 | 41 | 40 | 67 | 72 | 77 | 77 | 68 | 72 | 36 | 37 |
| 4 | 8 | 5C | 59 | 61 | 69 | 69 | 6F | 65 | 61 | 35 | 36 |

Hasil dari proses *AddRoundKey* atau *round* ke-10 ubah ke bentuk karakter dalam table ASCII.

|  |  |  |
| --- | --- | --- |
| ***Round*** | **Kode ASCII** | **Karakter** |
| 53 | 83 | S |
| 75 | 117 | u |
| 68 | 104 | h |
| 65 | 101 | e |
| 6E | 110 | n |
| 64 | 100 | d |
| 72 | 114 | r |
| 61 | 97 | a |
| 5F | 95 | \_ |
| 30 | 48 | 0 |
| 36 | 54 | 6 |
| 35 | 53 | 5 |
| 35 | 53 | 5 |
| 37 | 55 | 7 |
| 37 | 55 | 7 |
| 36 | 54 | 6 |

Dan hasil dekripsi dari AES menghasilkan *plainteks* “S u h e n d r a \_ 0 6 5 5 7 7 6”.