TUGAS TAMBAHAN

DES CIPHER

Disusun sebagai salah satu tugas

mata kuliah Kriptografi



Patricia Joanne

140810160065

Dikumpulkan tanggal

22 Desember 2018

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2018

**PENGERTIAN**

DES merupakan salah satu algoritma kriptografi cipher block dengan ukuran blok 64 bit dan ukuran kuncinya 56 bit. Algoritma DES dibuat di IBM dan merupakan modifikasi dari algoritma terdahulu yang bernama Lucifer. Lucifer merupakan algoritma cipher block yang beroperasi pada blok masukan 64 bit dan kuncinya berukuran 28 bit. Pengurangan jumlah bit kunci pada DES dilakukan dengan alasan agar mekanisme algoritma ini bisa diimplementasikan dalam satu chip.

**SEJARAH**

DES bermula dari hasil riset Tuchman Meyer yang diajukan sebagai kandidat sandi standard Nasional yang diusulkan oleh NBS (National Bureau Standard). Konon katanya, algoritma yang dikembangkan oleh Tuchman Meyer ini merupakan algoritma terbaik dari semua kandidat Sandi Standard Nasional.

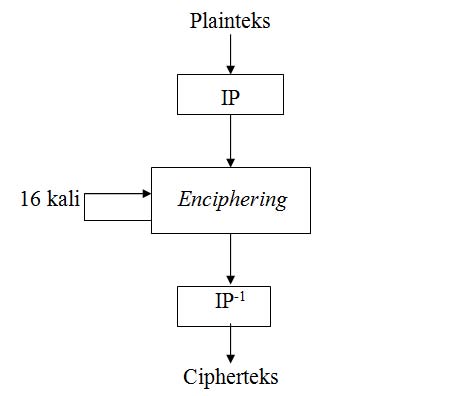
Pada mulanya, DES memiliki panjang kunci sandi 128 bit. Namun selama proses pengadopsian, NBS melibatkan NSA (National Security Agency) dan algoritma sandi ini mengalami pengurangan ukuran kunci sandi dari 128 bit menjadi 56 bit saja. Sebagian orang mungkin mengira bahwa pengurangan panjang kunci sandi ini merupakan usulan NSA untuk melemahkan algoritma Tuchman Meyer karena motif politik tertentu entah itu untuk mempermudah penyadapan atau untuk melemahkan pengamanan informasi lawan politik. Mungkin NSA menginginkan algoritma Tuchman Meyer ini “cukup aman” untuk digunakan warga sipil, tetapi mudah dipecahkan oleh organisasi besar semisal NSA dengan peralatan canggihnya. Bila dibandingkan dengan performa komputer personal pada saat itu, algoritma sandi dengan panjang kunci 56 bit dapat dikatakan cukup aman bila digunakan oleh orang-orang “biasa”, tapi dapat dengan mudah dipecahkan dengan peralatan canggih dan tentunya kepemilikan alat canggih ini hanya dapat dijangkau oleh organisasi elit seperti NSA. Dengan dukungan dana yang melimpah, pembuatan alat brute‐force DES bukanlah hal yang mustahil pada saat itu.

Kini algoritma DES sudah usang dan keamanannya pun sudah tidak dapat dipertanggungjawabkan lagi. DES telah secara resmi digantikan fungsinya oleh AES (Advanced Encryption Standard) dengan panjang kunci sandi 128, 192 dan 256 bit. Meskipun begitu, tidak ada salahnya jika kita mempelajari algoritma ini untuk tujuan hobi atau pendidikan.

**IMPLEMENTASI**

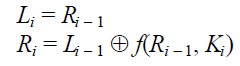
1. DES sudah diimplementasikan dalam bentuk chip. Setiap detik chip ini dapat mengenkripsikan 16,8 juta blok atau 1 gigabit per detik.
2. DES dapat melakukan enkripsi 32.000 blok per detik pada komputer mainframe IBM 3090.
3. DES digunakan pada enkripsi PIN (Personal Identification Numbers) mesin ATM (Automatic Teller Machine) dan transaksi perbankan lewat internet.
4. Organisasi pemerintahan di Amerika seperti Department of Energy, Justice Department, dan Federal Reserve System menggunakan DES untuk melindungi penyebaran data mereka.

**SKEMA GLOBAL DES**

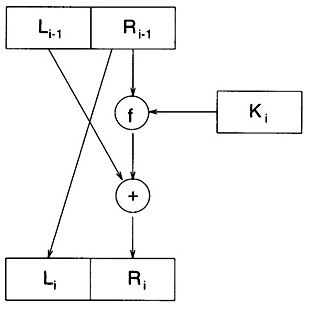


1. Blok plain text dipermutasi dengan matriks permutasi awal yang disebut juga dengan initial permutation atau IP.
2. Hasil permutasi awal kemudian di-enciphering sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.

* Di dalam proses enciphering, blok plainteks terbagi menjadi dua bagian, kiri (L) dan kanan (R), yang masing-masing panjangnya 32 bit.
* Pada setiap putaran i, blok R merupakan masukan untuk fungsi transformasi yang disebut f. Pada fungsi f, blok R dikombinasikan dengan kunci internal Ki. Keluaran dari fungsi f di-XOR-kan dengan blok L untuk mendapatkan blok R yang baru sedangkan blok L yang baru langsung diambil dari blok R sebelumnya. Ini adalah satu putaran DES.
* Secara matematis, satu putaran DES dinyatakan sebagai:



Atau dapat juga ditunjukkan dalam diagram sebagai berikut:

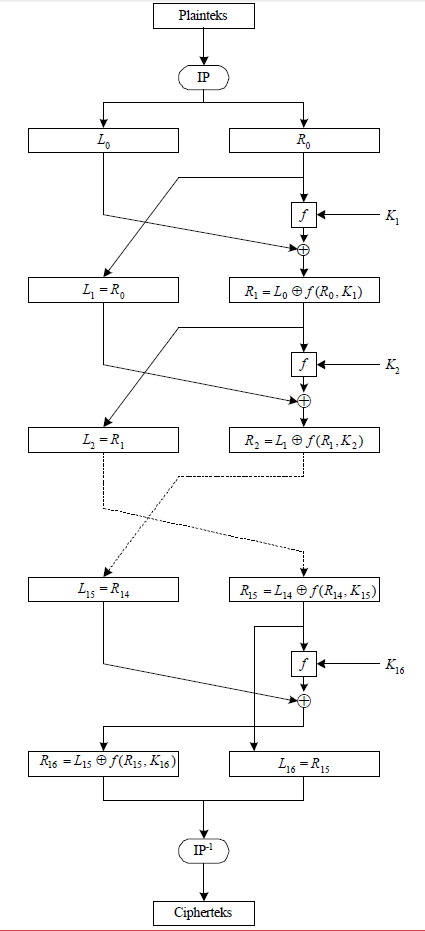


1. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP-1) menjadi blok cipher text.

**LANGKAH MENGHITUNG**

DES merupakan algoritma enkripsi blok simetris karena pemrosesan data baik enkripsi maupun dekripsi diimplementasikan per blok (dalam hal ini 8 byte). Algoritma yang digunakan untuk enkripsi relatif atau bahkan sama persis dengan algoritma yang digunakan dalam proses dekripsi.

1. Enkripsi
   1. Ubah plain text dan key ke dalam bentuk biner.
   2. Lakukan Initial Permutation pada plain text sehingga menghasilkan IP(x).
   3. Pecah dua bagian bit kiri dan kanan IP(x) menjadi L0 dan R0.
   4. Generate key dengan permutasi kompresi PC-1 (buang 1 bit blok kunci dari 64 bit menjadi 56 bit) sehingga menjadi CD(k).
   5. Pecah dua bagian bit kiri dan kanan CD(k) menjadi C0 dan D0.
   6. Lakukan left shift C0 dan D0 sebanyak 1 atau 2 kali berdasarkan putaran ke-inya.
   7. Setiap hasil putaran digabungkan kembali menjadi CiDi dan diinput ke dalam tabel permutasi kompresi 2 (PC-2) sehingga CiDi 56 bit menjadi CiDi 48 bit.
   8. Ekspansi data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1 ≤ i ≤ 16 menggunakan Tabel Ekspansi (E).
   9. Hasil E(Ri-1) kemudian di XOR dengan Ki dan menghasilkan Vektor Matriks Ai.
   10. Substitusikan Vektor Ai ke delapan buah S-Box (Substitution Box) dan menghasilkan output vektor Bi 32 bit menjadi nilai vektor Bi.
   11. Mutasikan bit vektor Bi menggunakan tabel P-Box kemudian dikelompokkan menjadi 4 blok dimana tiap-tiap blok memiliki 32 bit data.
   12. Hasil P(Bi) kemudian di XOR dengan Li-1 untuk mendapatkan nilai Ri.  
       Sedangkan nilai Li sendiri diperoleh dari Nilai Ri-1 untuk nilai 1≤ i ≤16.
   13. Gabungkan R16 dengan L16 kemudian dipermutasikan untuk terakhir kali dengan tabel Invers Initial Permutasi (IP-1) menghasilkan output cipher text.



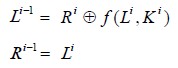
1. Dekripsi

Proses dekripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K1, K2, …, K16, maka pada proses dekripsi urutan kunci yang digunakanadalah K16, K15, …, K1.

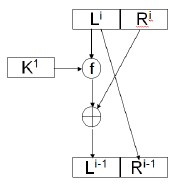
Cara untuk mendapatkan plain text kembali yaitu:

x = IP-1 (RD0 LD0)

Secara matematis, satu putarannya dinyatakan sebagai:



Atau dapat juga ditunjukkan dalam diagram sebagai berikut:



**CONTOH SOAL**

Diberikan:

Plain text : 0123456789ABCDEF (Hexadesimal)

Key : 133457799BBCDFF1 (Hexadesimal)

Ditanya: Cipher text?

Jawab:

* Ubah plain text ke dalam bentuk biner:

0000 0001 0010 0011

0100 0101 0110 0111

1000 1001 1010 1011

1100 1101 1110 1111‬

* Lakukan Initial Permutation pada plain text sehingga menghasilkan IP(x):

1100 1100

0000 0010

1100 1100

1111 1111

1111 0000

1010 1010

1111 0000

1010 1010

* Didapatkan L0 dan R0 dimana:

L0 = 1100 1100 0000 0010 1100 1100 1111 1111

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

* Ubah key ke dalam bentuk biner:

0001 0011 0011 0100

0101 0111 0111 1001

1001 1011 1011 1100

1101 1111 1111 0001‬

* Generate key ke dalam PC-1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |

* Didapatkan C0 dan D0 dimana C0 adalah 28 bit pertama dan D0 adalah 28 bit terakhir.
* Iterasi C0 dan D0 sebanyak 16 kali:

Ci = LSi(Ci-1)

Di = LSi(Di-1)

Misal iterasi pertama:

C1 = 1110000 1100110 0101010 1011111

D1 = 1010101 0110011 0011110 0011110

* Ki = PC-2(CiDi)

Lakukan iterasi hingga didapatkan K1-K16

* Mencari E(R0)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |

* Mencari E(R0) XOR K1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |

* S-Box

B1 = 011000

Row: 1100

Column: 00

C1 = 0101

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S1 | | | | | | | | | | | | | | | |
| 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

Lakukan sampai C8.

C = C1C2C3C4C5C6C7C8

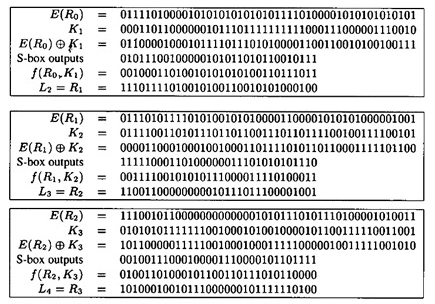
* Permutasi tetap C

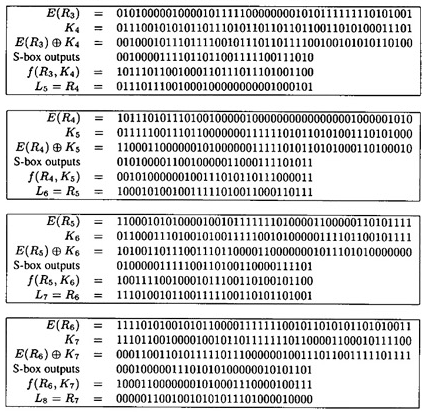
|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 |

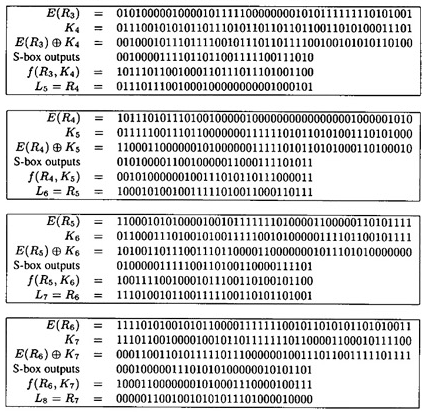
* F(R0, K1) atau R0 XOR K1

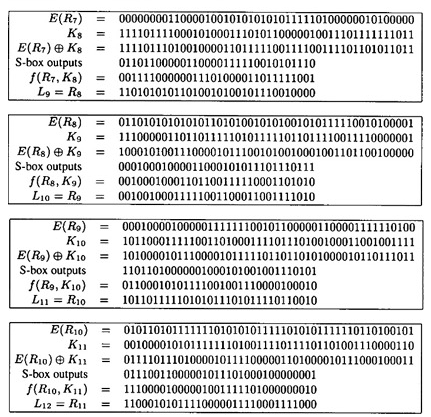
L2 = R1 = 1110 1111 0100 1010 0100 0101 0100 0100

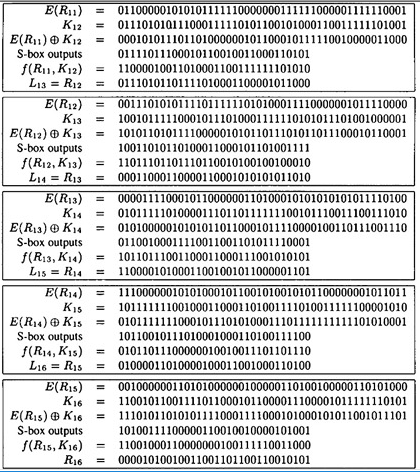
* Lakukan iterasi sebanyak 16 kali sehingga didapat L16 dan R16

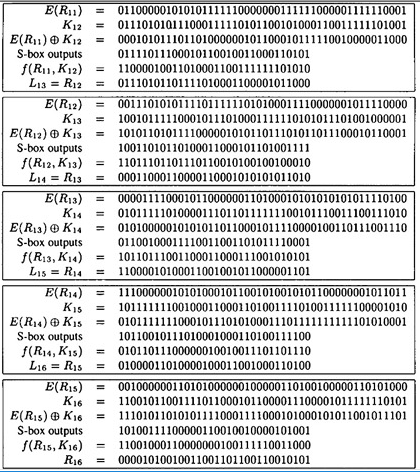












* Lakukan Inverse inisial permutasi terbalik

Cipher text = IP-1 (R16 L16)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

* Sehingga cipher text: 85E813540F0AB405 (Hexadesimal)

**PROGRAM**

Program DES Cipher ini dibuat menggunakan Visual Basic.

Module Module1

Sub Main()

Console.WriteLine("Algoritma DES (Data Encryption Standard)")

'1. Tentukan kalimat yang akan dienkrip

Console.WriteLine("Masukkan kalimat yang akan dienkrip: ")

Dim input As String = Console.ReadLine

Console.WriteLine("")

'2. Tentukan kata kunci enkripsi yang digunakan

Console.WriteLine("Masukkan kata kunci enkripsi: ")

Dim kataKunci As String = Console.ReadLine

Console.WriteLine("")

'3. Lakukan inisialisasi variabel yang digunakan oleh metode ini

Dim xc As New CryptCore()

xc.InitCore()

xc.Key = kataKunci

'4. Lakukan enkripsi kalimat awal menggunakan algoritma ini

Dim hasilEnkripsi As String = xc.Encrypt(input)

Console.WriteLine("Hasil enkripsi kalimat input adalah: " & vbCrLf & hasilEnkripsi.ToString & vbCrLf)

'5. Lakukan dekripsi dari kalimat yang telah terenkripsi

Dim hasilDekripsi As String = xc.Decrypt(hasilEnkripsi)

Console.WriteLine("Hasil dekripsi dari kalimat terenkripsi adalah: " & vbCrLf & hasilDekripsi & vbCrLf)

Console.ReadLine()

End Sub

End Module

'Class CryptCore adalah inti class untuk pemanggilan fungsi enkripsi dan dekripsi pada algoritma DES (Data Encryption Standard)

Public Class CryptCore

Private \_key As String = Nothing

Public Property Key() As String

Get

Return \_key

End Get

Set(value As String)

\_key = Me.formatKey(value)

End Set

End Property

Private Function formatKey(key As String) As String

If key Is Nothing OrElse key.Length = 0 Then

Return Nothing

End If

Return key.Trim()

End Function

Private DefaultKey As String = ""

Public Sub New()

DefaultKey = "enkripsi"

End Sub

Private \_coreSymmetric As CoreAlgoritmaSymmetric

Public Function InitCore() As Boolean

\_coreSymmetric = New CoreAlgoritmaSymmetric()

Return True

End Function

Public Function Decrypt(src As String) As String

Dim hasil As String = ""

If \_key Is Nothing Then

hasil = \_coreSymmetric.ProsesDecrypt(src, DefaultKey)

Else

hasil = \_coreSymmetric.ProsesDecrypt(src, \_key)

End If

Return hasil

End Function

Public Function Decrypt(src As String, key As String) As String

Dim hasil As String = ""

hasil = \_coreSymmetric.ProsesDecrypt(src, key)

Return hasil

End Function

Public Function Encrypt(src As String) As String

Dim hasil As String = ""

If \_key Is Nothing Then

hasil = \_coreSymmetric.ProsesEncrypt(src, DefaultKey)

Else

hasil = \_coreSymmetric.ProsesEncrypt(src, \_key)

End If

Return hasil

End Function

Public Function Encrypt(src As String, key As String) As String

Dim hasil As String = ""

hasil = \_coreSymmetric.ProsesEncrypt(src, key)

Return hasil

End Function

‘Pada saat melakukan proses InitCore, maka proses tersebut akan melakukan inisialisasi pada Class CoreAlgoritmaSymmetric

Public Class CoreAlgoritmaSymmetric

Private metodeEncode As System.Security.Cryptography. SymmetricAlgorithm

Public Sub New()

metodeEncode = New System.Security.Cryptography. DESCryptoServiceProvider()

End Sub

Private Function GetValidKey(Key As String) As Byte()

Dim sTemp As String

If metodeEncode.LegalKeySizes.Length > 0 Then

Dim lessSize As Integer = 0, moreSize As Integer = metodeEncode.LegalKeySizes(0).MinSize

While Key.Length \* 8 > moreSize AndAlso metodeEncode. LegalKeySizes(0).SkipSize > 0 AndAlso moreSize < metodeEncode. LegalKeySizes(0).MaxSize

lessSize = moreSize

moreSize += metodeEncode.LegalKeySizes(0).SkipSize

End While

If Key.Length \* 8 > moreSize Then

sTemp = Key.Substring(0, (moreSize / 8))

Else

sTemp = Key.PadRight(moreSize / 8, " "c)

End If

Else

sTemp = Key

End If

'Konversi kata kunci menjadi byte array

Return System.Text.ASCIIEncoding.ASCII.GetBytes(sTemp)

End Function

Private Function GetValidIV(InitVector As [String], panjangValid As Integer) As Byte()

If InitVector.Length > panjangValid Then

Return System.Text.ASCIIEncoding.ASCII.GetBytes (InitVector.Substring(0, panjangValid))

Else

Return System.Text.ASCIIEncoding.ASCII.GetBytes (InitVector.PadRight(panjangValid, " "c))

End If

End Function

‘Skrip ini akan menjalankan proses enkripsi yang terdapat dalam Class CoreAlgoritmaSymmetric

Public Function ProsesEncrypt(Source As String, Key As String) As String

If Source Is Nothing OrElse Key Is Nothing OrElse Source.Length = 0 OrElse Key.Length = 0 Then

Return Nothing

End If

If metodeEncode Is Nothing Then

Return Nothing

End If

Dim lPanjangStream As Long

Dim jumlahBufferTerbaca As Integer

Dim byteBuffer As Byte() = New Byte(2) {}

Dim srcData As Byte() = System.Text.ASCIIEncoding.ASCII. GetBytes(Source)

Dim encData As Byte()

Dim streamInput As New System.IO.MemoryStream()

streamInput.Write(srcData, 0, srcData.Length)

streamInput.Position = 0

Dim streamOutput As New System.IO.MemoryStream()

Dim streamEncrypt As System.Security.Cryptography. CryptoStream

metodeEncode.Key = GetValidKey(Key)

metodeEncode.IV = GetValidIV(Key, metodeEncode.IV.Length)

streamEncrypt = New System.Security.Cryptography. CryptoStream(streamOutput, metodeEncode.CreateEncryptor(), System. Security.Cryptography.CryptoStreamMode.Write)

lPanjangStream = streamInput.Length

Dim totalBufferTerbaca As Integer = 0

While totalBufferTerbaca < lPanjangStream

jumlahBufferTerbaca = streamInput.Read(byteBuffer, 0, byteBuffer.Length)

streamEncrypt.Write(byteBuffer, 0, jumlahBufferTerbaca)

totalBufferTerbaca += jumlahBufferTerbaca

End While

streamEncrypt.Close()

encData = streamOutput.ToArray()

'Konversi menjadi base64 agar dapat digunakan dalam xml

Return Convert.ToBase64String(encData)

End Function

‘Skrip ini akan menjalankan proses dekripsi yang terdapat dalam Class CoreAlgoritmaSymmetric

Public Function ProsesDecrypt(Source As String, Key As String) As String

If Source Is Nothing OrElse Key Is Nothing OrElse Source.Length = 0 OrElse Key.Length = 0 Then

Return Nothing

End If

If metodeEncode Is Nothing Then

Return Nothing

End If

Dim lPanjangStream As Long

Dim jumlahBufferTerbaca As Integer

Dim byteBuffer As Byte() = New Byte(2) {}

Dim encData As Byte() = Convert.FromBase64String(Source)

Dim decData As Byte()

Dim streamInput As New System.IO.MemoryStream(encData)

Dim streamOutput As New System.IO.MemoryStream()

Dim streamDecrypt As System.Security.Cryptography. CryptoStream

metodeEncode.Key = GetValidKey(Key)

metodeEncode.IV = GetValidIV(Key, metodeEncode.IV.Length)

streamDecrypt = New System.Security.Cryptography. CryptoStream(streamInput, metodeEncode.CreateDecryptor(), System. Security.Cryptography.CryptoStreamMode.Read)

lPanjangStream = streamInput.Length

Dim totalBufferTerbaca As Integer = 0

While totalBufferTerbaca < lPanjangStream

jumlahBufferTerbaca = streamDecrypt.Read(byteBuffer, 0, byteBuffer.Length)

If 0 = jumlahBufferTerbaca Then

Exit While

End If

streamOutput.Write(byteBuffer, 0, jumlahBufferTerbaca)

totalBufferTerbaca += jumlahBufferTerbaca

End While

streamDecrypt.Close()

decData = streamOutput.ToArray()

For i As Integer = 0 To decData.Length - 1

If decData(i) < 8 Then decData(i) = 0

Next

Dim encodeASCII As New System.Text.ASCIIEncoding()

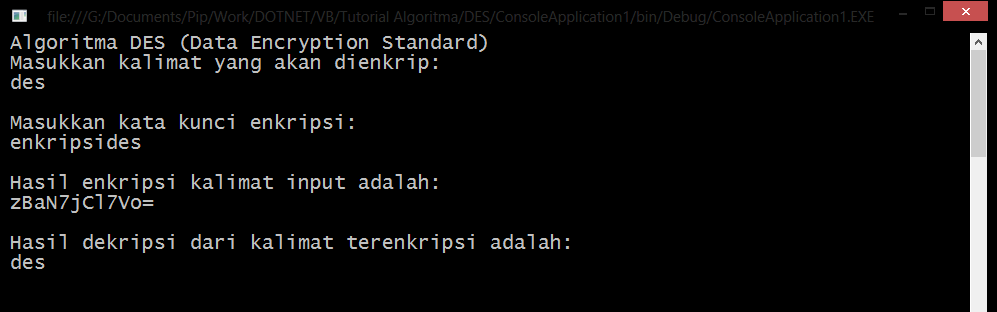
Return encodeASCII.GetString(decData)

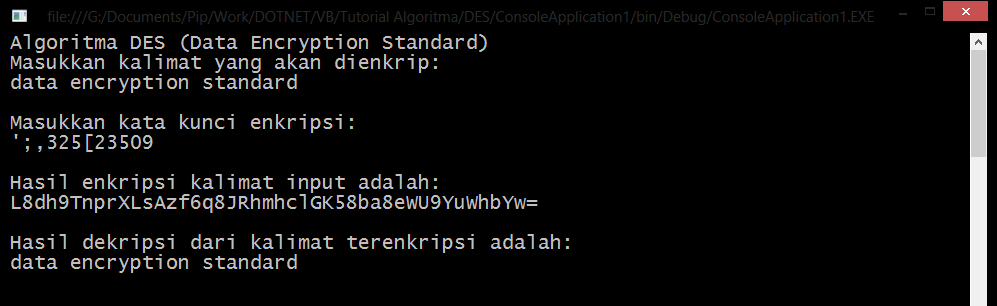
End Function

End Class

End Class

Hasil screenshot program yang dijalankan:





**DAFTAR PUSTAKA**

*Cryptography: Theory and Practice* by Douglas Stinson

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Data%20Encryption%20Standard%20(DES).pdf>

<http://itrezasaputra.blogspot.com/2016/03/contoh-soal-data-encryption-standard.html>

<http://octarapribadi.blogspot.com/2012/10/contoh-enkripsi-dengan-algoritma-des.html>

<http://studyinformatics.blogspot.com/2012/07/des-data-encryption-standard.html>

<https://makalah-update.blogspot.com/2012/11/makalah-pengertian-dan-sejarah-des-data.html>

<https://piptools.net/algoritma-des-data-encryption-standard/>

<https://www.academia.edu/12741838/Makalah-algoritma-kriptografi-des>