

# **KRIPTOGRAFI FUNGSI XOR**

## **KEAMANAN INFORMASI DAN JARINGAN**



**Disusun oleh:**

**Kahil Akbar Bayu Adityo 1203210038 (IF-01-01)**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS**  
**INSTITUT TEKNOLOGI TELKOM SURABAYA**

**2023**

## 1. Link Github:

<https://github.com/KahilAkbr/XOR-Encrypt-Decrypt>

## 2. Deskripsi Program

- Program ini menerapkan fungsi XOR untuk melakukan enkripsi dan dekripsi. Pada proses enkripsi melakukan XOR pada plainteks dan kunci sedangkan pada dekripsi melakukan XOR pada cipherteks dan kunci. Karena proses yang sama, maka baik pada proses enkripsi maupun dekripsi akan menjalankan fungsi yang sama.
- Terdapat 2 pilihan menu untuk melakukan proses enkripsi atau dekripsi yaitu input pesan berupa karakter ataupun input pesan berupa bit. Format bit yang digunakan adalah ASCII 8 bit yang dapat diakses pada link github yang sudah tertera.
- Output hasil enkripsi/dekripsi diletakkan pada file bernama result.txt. Hal ini dikarenakan 32 karakter pertama dari ASCII 8 bit tidak dapat dicetak pada terminal.
- Jika memilih menu dengan input pesan berupa karakter, maka input pesan harus berupa karakter dari ASCII 8 bit (misal: \$, a, #, dll) dan key harus berupa karakter dari ASCII 8 bit juga. Sedangkan jika memilih menu dengan input pesan berupa bit, maka input pesan harus berupa bit (8 bit) yang mana harus berkelipatan 8, tapi input key tetap berupa karakter. Serta input karakter minimal adalah 6 karakter dan untuk input bit minimal adalah 6 byte (48 bit).

## 3. Penjelasan Algoritma

- Melakukan pemilihan menu untuk menentukan tipe input apa yang ingin digunakan pada pesan.
- **Input berupa karakter**
  - Input pesan dan kunci berupa karakter.
  - Pengecekan panjang pesan dan kunci, apabila tidak lebih dari 5 karakter maka akan melakukan input ulang. Proses ini dilakukan dengan fungsi `checkInputLength()`.
  - Menyamakan panjang pesan dan key. Apabila pesan lebih panjang maka key akan berulang hingga panjangnya sama. Apabila pesan lebih pendek maka key akan dipotong sampai panjangnya sama dengan pesan. Proses ini dilakukan dengan fungsi `equalizeLength()`.

```
Penyamaan Panjang Pesan dan Kunci
Pesan: Kahil Akbar
Panjang Karakter Pesan: 11
Key: 12345612345
Panjang Karakter Kunci: 11
```

- Mengubah karakter menjadi bentuk 8 bit dengan fungsi `equalizeBitDigit()` yang juga memisahkan tiap 8 bit. Serta menampilkannya dengan fungsi `printMsgAndKey()`.

```
Bit Pesan:
01001011 01100001 01101000 01101001 01101100 00100000 01000001 01101011 01100010 01100001 01110010
Bit Key:
00110001 00110010 00110011 00110100 00110101 00110110 00110001 00110010 00110011 00110100 00110101
```

- Setelah dijadikan terpisah setiap 8 bit akan digabungkan kembali (menghapus spasi). Lalu, dilakukan fungsi XOR dengan cara, jika bit pesan = bit key maka akan mengembalikan 0 sedangkan, jika bit pesan != bit key maka akan mengembalikan 1. Proses ini dilakukan dengan fungsi `xorBits()` yang menerima parameter bit pesan dan bit key.
- Agar hasil XOR lebih jelas maka dipisahkan kembali setiap 8 bit dengan menggunakan fungsi `printXORResult()`.

```
Hasil XOR dari Bit Pesan dan Bit Key:
01111010 01010011 01011011 01011101 01011001 00010110 01110000 01011001 01010001 01010101 01000111
```

- Hasil XOR sebelum dipisah tiap 8 bit akan ditulis pada result.txt. Hasil XOR bentuk karakter akan mengubah bit hasil XOR dengan fungsi bitsToChar() dan akan ditulis juga pada result.txt.

```
Hasil Enkripsi Bentuk Bit:
0111101001010011011011101010110010001011001110000010110010101000101010101000111

Hasil Enkripsi Bentuk Karakter:
zS[]YsynpYQUG
```

Karakter yang berlatar merah merupakan karakter yang tidak dapat tampil dalam terminal.

- **Input berupa bit**

- Input pesan berupa bit dan kunci berupa karakter.
- Pengecekan panjang pesan, apabila pesan kurang dari 6 byte (48 bit) maka akan melakukan input ulang. Serta apabila pesan tidak berkelipatan 8 maka akan melakukan input ulang.
- Pengecekan panjang kunci, apabila tidak lebih dari 5 karakter maka akan melakukan input ulang.
- Menyamakan panjang pesan dan key. Apabila pesan lebih panjang maka key akan berulang hingga panjangnya sama. Apabila pesan lebih pendek maka key akan dipotong sampai panjangnya sama dengan pesan. Panjang pesan akan dibagi 8 karena inputnya berbentuk bit. Proses ini dilakukan dengan fungsi equalizeLength().

```
Penyamaan Panjang Pesan dan Kunci:
Pesan: 0111101001010011011011101010110010001011001110000010110010101000101010101000111
Panjang Karakter Pesan: 11
Key: 12345612345
Panjang Karakter Kunci: 11
```

- Memisahkan pesan tiap 8 bit dengan fungsi printXORResult() dan mengubah kunci menjadi bit lalu memisahkannya tiap 8 bit dengan fungsi equalizeBitDigit(). Serta menampilkannya dengan fungsi printMsgAndKey().

```
Bit Pesan:
01111010 01010011 01011011 01011101 01011001 00010110 01110000 01011001 01010001 01010101 01000111
Bit Key:
00110001 00110010 00110011 00110100 00110101 00110110 00110001 00110010 00110011 00110100 00110101
```

- Setelah dijadikan terpisah setiap 8 bit akan digabungkan kembali (menghapus spasi). Lalu, dilakukan fungsi XOR dengan cara, jika bit pesan = bit key maka akan mengembalikan 0 sedangkan, jika bit pesan != bit key maka akan mengembalikan 1. Proses ini dilakukan dengan fungsi xorBits() yang menerima parameter bit pesan dan bit key.
- Agar hasil XOR lebih jelas maka dipisahkan kembali setiap 8 bit dengan menggunakan fungsi printXORResult().

```
Hasil XOR dari Bit Pesan dan Bit Key:
01001011 01100001 01101000 01101001 01101100 00100000 01000001 01101011 01100010 01100001 01110010
```

- Hasil XOR sebelum dipisah tiap 8 bit akan ditulis pada result.txt. Hasil XOR bentuk karakter akan mengubah bit hasil XOR dengan fungsi bitsToChar() dan akan ditulis juga pada result.txt.

```
Hasil Enkripsi/Dekripsi Bentuk Bit:
0100101101100001011010000110100101101100001000000100000101101011011000100110000101110010

Hasil Enkripsi/Dekripsi Bentuk Karakter:
Kahil Akbar
```

#### 4. Hasil Demo

- **Input Karakter**  
Enc (Terminal):



```

PS C:\DATA\SEM5\KIJ\vigener-cipher_bitwise_XOR> & "C:/Program Files/Python311/python.exe" c:/DATA/SEM5/KIJ/vigener-cipher_bitwise_XOR/XOR-Encrypt-Decrypt.py
Enkripsi dan Dekripsi Berbasis ASCII 8 BIT

Tipe Input Pesan:
1. Karakter
2. Bit
Pilih tipe input (1/2): 1
Tipe Input: Karakter
Masukkan Pesan: Kahil Akbar
Masukkan Key: 123ABC

Penyamaan Panjang Pesan dan Kunci:
Pesan: Kahil Akbar
Panjang Karakter Pesan: 11
Key: 123ABC123AB
Panjang Karakter Kunci: 11

Bit Pesan:
01001011 01100001 01101000 01101001 01101100 00100000 01000001 01101011 01100010 01100001 01110010
Bit Key:
00110001 00110010 00110011 01000001 01000010 01000011 00110001 00110010 00110011 01000001 01000010

Hasil XOR dari Bit Pesan dan Bit Key:
01111010 01010011 01011011 00101000 00101110 01100011 01110000 01011001 01010001 00100000 00110000

=====

Hasil enkripsi berhasil disimpan di file result.txt
=====

```

Enc (result.txt):

```

Hasil Enkripsi Bentuk Bit:
0111101001010011010110110010100000101110011000110111000001011001010100010010000000110000

Hasil Enkripsi Bentuk Karakter:
zS[(.cpYQ 0

```

Dec (Terminal):

```

PS C:\DATA\SEM5\KIJ\vigener-cipher_bitwise_XOR> & "C:/Program Files/Python311/python.exe" c:/DATA/SEM5/KIJ/vigener-cipher_bitwise_XOR/XOR-Encrypt-Decrypt.py
Enkripsi dan Dekripsi Berbasis ASCII 8 BIT

Tipe Input Pesan:
1. Karakter
2. Bit
Pilih tipe input (1/2): 1
Tipe Input: Karakter
Masukkan Pesan: zS[(.cpYQ 0
Masukkan Key: 123ABC

Penyamaan Panjang Pesan dan Kunci:
Pesan: zS[(.cpYQ 0
Panjang Karakter Pesan: 11
Key: 123ABC123AB
Panjang Karakter Kunci: 11

Bit Pesan:
01111010 01010011 01011011 00101000 00101110 01100011 01110000 01011001 01010001 00100000 00110000
Bit Key:
00110001 00110010 00110011 01000001 01000010 01000011 00110001 00110010 00110011 01000001 01000010

Hasil XOR dari Bit Pesan dan Bit Key:
01001011 01100001 01101000 01101001 01101100 00100000 01000001 01101011 01100010 01100001 01110010

=====

Hasil enkripsi berhasil disimpan di file result.txt
=====

```

Dec (result.txt):

```

Hasil Enkripsi Bentuk Bit:
0100101101100001011010000110100101101100001000000100000101101011011000100110000101110010

Hasil Enkripsi Bentuk Karakter:
Kahil Akbar

```

- **Input Bit**

Misal pesannya adalah Kahil Akbar Bayu Adityo yang mana memiliki bit seperti berikut.

```

01001011 01100001 01101000 01101001 01101100 00100000 01000001 01101011 01100010
01100001 01110010 00100000 01000010 01100001 01111001 01110101 00100000 01000001
01100100 01101001 01110100 01111001 01101111

```

Enc (Terminal):



IT Telkom  
Surabaya



Yayasan  
Pendidikan  
Telkom



KIJ IF-01-01

```
PS C:\DATA\SEM5\KIJ>vigenere-cipher_bitwise_XOR & "C:/Program Files/Python311/python.exe" c:/DATA/SEM5/KIJ/vigenere-cipher_bitwise_XOR/XOR-Encrypt-Decrypt.py
Enkripsi dan Dekripsi Berbasis ASCII 8 BIT

Tipe Input Pesan:
1. Karakter
2. Bit
Pilih tipe input (1/2): 2
Tipe Input: Bit
Pesan setidaknya harus terdiri dari 6 byte yang masing-masing merupakan kode 8 bit
Masukkan Pesan: 010010110110000101101000110101011010000100000010000010011000010111001011010010000001000001011001000110101000111001010111
Masukkan Key: 123ABC

Penyamaan Panjang Pesan dan Kunci:
Pesan: 01001011011000010110100011010101101100001000000101101011000100110000101110010001000000100001001100001011100101101000111001010111
Panjang Karakter Pesan: 23
Key: 123ABC123ABC123ABC123AB
Panjang Karakter Kunci: 23

Bit Pesan:
01001011 01100001 01101000 01101100 00100000 01000001 01101011 01100010 01100001 01110010 00100000 01000010 01100001 01111001 01110101 00100000 01000001 01100100 01101001 01110100 01111001 01101111
Bit Key:
00110001 00110010 00110011 01000001 01000010 01000011 00110010 00110010 00110011 01000001 01000010 01000011 00110001 00110010 00110011 01000001 01000010 00110001 00110010 00110011 01000001 01000010

Hasil XOR dari Bit Pesan dan Bit Key:
01111010 01010011 01011011 00101000 00101110 01100011 01110000 01011001 01010001 00100000 00110000 01100011 01110011 01010011 01001010 00110100 01100010 00000010 01010101 01011011 01000111 00111000 00101101

=====

Hasil enkripsi/dekripsi berhasil disimpan di file result.txt

=====
```

## Enc (result.txt):

```
Hasil Enkripsi/Dekripsi Bentuk Bit:
011110100100110101101100101000001011100110001101110000010110010101000100100000011000011000110111001101001010010100011010001100010000000100101010110110100011100111000
00101101

Hasil Enkripsi/Dekripsi Bentuk Karakter:
z5[(-cpYQ 0cs5J4bmU/G8.
```

## Dec (Terminal):

```
PS C:\DATA\SEM5\KIJ>vigenere-cipher_bitwise_XOR & "C:/Program Files/Python311/python.exe" c:/DATA/SEM5/KIJ/vigenere-cipher_bitwise_XOR/XOR-Encrypt-Decrypt.py
Enkripsi dan Dekripsi Berbasis ASCII 8 BIT

Tipe Input Pesan:
1. Karakter
2. Bit
Pilih tipe input (1/2): 2
Tipe Input: Bit
Pesan setidaknya harus terdiri dari 6 byte yang masing-masing merupakan kode 8 bit
Masukkan Pesan: 01111010010100110101101100101000001011100110011011000001011001010100010010000001100001100010111001101010011010011010001100110001100000010010101011011010001110011100000101101
Masukkan Key: 123ABC

Penyamaan Panjang Pesan dan Kunci:
Pesan: 0111101001010011010110110010000010111001100110110000010110010101000100100000011000011000101110011010100110100110001100011000110001100011001100000010010101011011010001110011100000101101
Panjang Karakter Pesan: 23
Key: 123ABC123ABC123ABC123AB
Panjang Karakter Kunci: 23

Bit Pesan:
01111010 01010011 01011011 00101000 00101110 01100011 01110000 01011001 01010001 00100000 00110000 01100011 01110011 01010011 01001010 00110100 01100010 00000010 01010101 01011011 01000111 00111000 00101101
Bit Key:
00110001 00110010 00110011 01000001 01000010 01000011 00110001 00110010 00110011 01000001 01000010 01000011 00110001 00110010 00110011 01000001 01000010 00000010 00110001 00110010 01000001 01000010

Hasil XOR dari Bit Pesan dan Bit Key:
01001011 01100001 01010000 01010101 01011000 00100000 01000001 01101011 01100010 01100001 01110010 00100000 01000010 01110001 01110101 00100000 01000001 01100100 01101001 01110100 01111001 01101111

=====

Hasil enkripsi/dekripsi berhasil disimpan di file result.txt

=====
```

## Dec (result.txt):

```
Hasil Enkripsi/Dekripsi Bentuk Bit:
0100101101100001011010000110100101101100001000000100000101101011011000100110000101110010111010100100000010000010110010001110010001111001
01101111

Hasil Enkripsi/Dekripsi Bentuk Karakter:
Kahil Akbar Bayu Adityo
```