

**TUGAS 4**  
**KEAMANAN INFORMASI DAN JARINGAN**



**NAMA : ARDINA FIRDATANA**  
**NIM : 1203210018**  
**KELAS : IF-01-03**

**PRODI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS**  
**INSTITUT TEKNOLOGI TELKOM SURABAYA**  
**SEMESTER GANJIL 2023/2024**

# PROGRAM SEDERHANA VIGENERE CIPHER BERBASIS BIT

## MENGGUNAKAN FUNGSI XOR

1. Buatlah program sederhana vigenere chiper berbasiskan bit menggunakan fungsi XOR
2. Plaintext dan kunci harus menggunakan lebih dari 5 karakter
3. Terdapat fungsi enkripsi dan dekripsi

### Source Code

```
while True:
    print("\n=== VIGENERE CIPHER ===")
    print("Menu")
    print("1. Enkripsi")
    print("2. Dekripsi")
    print("3. Keluar")

    action = input("Pilih menu: ")

    if action == '3':
        print("Keluar dari program.")
        break

    message = input("Masukkan pesan: ")
    key = input("Masukkan key: ")

    # Menyamakan panjang pesan dan kunci
    key = key * (len(message) // len(key)) + key[:len(message) % len(key)]

    # Konversi karakter ke bit
    message_bits = ''.join([format(ord(char), '08b') for char in message])
    key_bits = ''.join([format(ord(char), '08b') for char in key])

    print("\nBit pesan:")
    for i, char in enumerate(message):
        print(f"Karakter {char}: {message_bits[i*8:(i+1)*8]}")

    print("\nHasil XOR dari bit pesan dan bit key:")
    for i, char in enumerate(message):
        xor_result = '1' if message_bits[i*8:(i+1)*8] != key_bits[i*8:(i+1)*8]
    else '0'
        print(f"{message_bits[i*8:(i+1)*8]} XOR {key_bits[i*8:(i+1)*8]} = {xor_result}")

    # Proses enkripsi atau dekripsi
```

```

    result_bits = ''.join('1' if message_bits[i] != key_bits[i] else '0' for i
in range(len(message_bits))) if action == '1' else ''.join('1' if
message_bits[i] != key_bits[i] else '0' for i in range(len(message_bits)))

    print(f"\nHasil {'enkripsi' if action == '1' else 'dekripsi'} bentuk
bit:")
    print(result_bits)

    # Output hanya karakter terenkripsi atau terdekripsi
    result_message = ''.join(chr(int(result_bits[i*8:(i+1)*8], 2)) for i in
range(len(result_bits)//8))

    print(f"\nHasil {'enkripsi' if action == '1' else 'dekripsi'} bentuk
karakter:")
    print(result_message)

```

## Penjelasan

Kode program diatas merupakan implementasi penggunaan *Vigenere Cipher*. Terdapat 3 opsi pada menu utamanya, yakni **Enkripsi**, **Dekripsi**, dan **Keluar**. Program ini akan terus berjalan dalam *loop* tanpa henti hingga *user* memilih opsi 3 (**Keluar**) yang akan mencetak pesan “Keluar dari program” dan akan keluar dari *loop* utama. Saat program pertama kali dijalankan, *user* akan diminta untuk memilih dari ketiga opsi pada menu, kemudian memasukkan pesan (*plaintext*) dan kunci (*key*). Panjang pesan akan diukur menggunakan **len(message)** dan panjang kunci diukur menggunakan **len(key)**, kemudian keduanya akan dibagi dan baru bisa diputuskan berapa kali kuncinya bisa dimuat/diulang pada pesan. Apabila panjang kuncinya lebih pendek dari panjang pesan, maka kunci akan diulang hingga panjangnya sesuai dengan pesan. Akan tetapi, jika kunci lebih panjang daripada pesan, maka kunci yang akan digunakan hanya sampai panjangnya sama dengan pesan.

## Output

### Proses Enkripsi

- Memilih opsi **Enkripsi** dan memasukkan pesan (*plaintext*) serta *key*.

```

=== VIGENERE CIPHER ===
Menu
1. Enkripsi
2. Dekripsi
3. Keluar
Pilih menu: 1
Masukkan pesan: Ketintang Surabaya
Masukkan key: 1234E

```

- Konversi setiap karakter pesan menjadi representasi biner 8-bit.

```

Bit pesan:
Karakter K: 01001011
Karakter e: 01100101
Karakter t: 01110100
Karakter i: 01101001
Karakter n: 01101110
Karakter t: 01110100
Karakter a: 01100001
Karakter n: 01101110
Karakter g: 01100111
Karakter : 00100000
Karakter S: 01010011
Karakter u: 01110101
Karakter r: 01110010
Karakter a: 01100001
Karakter b: 01100010
Karakter a: 01100001
Karakter y: 01111001
Karakter a: 01100001

```

- Kemudian, dilakukan penghitungan XOR antara bit pesan dan *key* dari setiap karakter.

```

Hasil XOR dari bit pesan dan bit key:
01001011 XOR 00110001 = 1
01100101 XOR 00110010 = 1
01110100 XOR 00110011 = 1
01101001 XOR 00110100 = 1
01101110 XOR 01000101 = 1
01110100 XOR 00110001 = 1
01100001 XOR 00110010 = 1
01101110 XOR 00110011 = 1
01100111 XOR 00110100 = 1
00100000 XOR 01000101 = 1
01010011 XOR 00110001 = 1
01110101 XOR 00110010 = 1
01110010 XOR 00110011 = 1
01100001 XOR 00110100 = 1
01100010 XOR 01000101 = 1
01100001 XOR 00110001 = 1
01111001 XOR 00110010 = 1
01100001 XOR 00110011 = 1

```

- Menampilkan hasil berupa bit

```

Hasil enkripsi bentuk bit:
011110100101011101000111010111001010110000101010011010111010101001101100100100011101000001010101001001110100000100101101010010

```

- Konversi bit hasil enkripsi menjadi karakter dan hasilnya akan ditampilkan.

```

Hasil enkripsi bentuk karakter:
zWG]+ES]SebGAU'PKR

```

## Proses Dekripsi

- Program akan berulang dan kembali menampilkan menu dan memilih opsi 2 (**Dekripsi**). Kemudian, memasukkan pesan dari hasil enkripsi sebelumnya serta harus menggunakan *key* yang sama ketika melakukan proses enkripsi.

```
=== VIGENERE CIPHER ===
Menu
1. Enkripsi
2. Dekripsi
3. Keluar
Pilih menu: 2
Masukkan pesan: zWG]+ES]SebGAU'PKR
Masukkan key: 1234E
```

- Konversi setiap karakter pesan menjadi representasi biner 8-bit.

```
Bit pesan:
Karakter z: 01111010
Karakter W: 01010111
Karakter G: 01000111
Karakter ]: 01011101
Karakter +: 00101011
Karakter E: 01000101
Karakter S: 01010011
Karakter ]: 01011101
Karakter S: 01010011
Karakter e: 01100101
Karakter b: 01100010
Karakter G: 01000111
Karakter A: 01000001
Karakter U: 01010101
Karakter ': 00100111
Karakter P: 01010000
Karakter K: 01001011
Karakter R: 01010010
```

- Kemudian, dilakukan penghitungan XOR antara bit pesan dan *key* dari setiap karakter yang sudah dienkripsi.

```
Hasil XOR dari bit pesan dan bit key:
01111010 XOR 00110001 = 1
01010111 XOR 00110010 = 1
01000111 XOR 00110011 = 1
01011101 XOR 00110100 = 1
00101011 XOR 01000101 = 1
01000101 XOR 00110001 = 1
01010011 XOR 00110010 = 1
01011101 XOR 00110011 = 1
01010011 XOR 00110100 = 1
01100101 XOR 01000101 = 1
01100010 XOR 00110001 = 1
01000111 XOR 00110010 = 1
01000001 XOR 00110011 = 1
01010101 XOR 00110100 = 1
00100111 XOR 01000101 = 1
01010000 XOR 00110001 = 1
01001011 XOR 00110010 = 1
01010010 XOR 00110011 = 1
```

- Menampilkan hasil berupa bit

```
Hasil dekripsi bentuk bit:
01001011011001010111000110100101101110011000010110111001100110010000001010011011010101110011000010110001001100001011100101100001
```

- Konversi bit hasil dekripsi menjadi karakter dan hasilnya akan ditampilkan berupa pesan awal sebelum dilakukan enkripsi.

```
Hasil dekripsi bentuk karakter:  
Ketintang Surabaya
```

### Proses Keluar dari Program

- Program akan melakukan *looping* menampilkan menu dan *user* memilih opsi 3 (**Keluar**). Kemudian, *user* akan dikeluarkan dari *loop* menu utama dengan menampilkan pesan “Keluar dari program”.

```
=== VIGENERE CIPHER ===  
Menu  
1. Enkripsi  
2. Dekripsi  
3. Keluar  
Pilih menu: 3  
Keluar dari program.  
PS C:\Users\ASUS\OneDrive\Documents\kulyah\SMT 5\kij\tugas\vigenereCipher> █
```

### LINK GITHUB

<https://github.com/ardinana/Vigenere-Cipher-Berbasis-Bit.git>