

TUGAS

VIGENERE CHIPER BERBASIS BIT



Disusun Oleh:

Emilia Ramona 1203210017

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
TAHUN AJARAN 2021/2022

Kode program

```
def char_to_bin(char, bit_length=8):
    # Mengonversi karakter ke representasi biner dengan panjang tetap
    return format(ord(char), f'0{bit_length}b')

def xor_bits(bit1, bit2):
    # Operasi XOR antara dua bit
    return '1' if bit1 != bit2 else '0'

def equalize_length(message, key):
    # Menyamakan panjang pesan dan kunci dengan mengulang kunci jika lebih pendek
    key = key * (len(message) // len(key)) + key[:len(message) % len(key)]
    return key

def encrypt_bitwise(plaintext, key):
    # Enkripsi berbasis bit menggunakan XOR
    ciphertext_bits = [xor_bits(p, k) for p, k in zip(plaintext, key)]
    return ''.join(ciphertext_bits)

def decrypt_bitwise(ciphertext, key):
    # Dekripsi berbasis bit menggunakan XOR
    decrypted_bits = [xor_bits(c, k) for c, k in zip(ciphertext, key)]
    return ''.join(decrypted_bits)

def main():
    print("Enkripsi dan Dekripsi berbasis ASCII 8 BIT")
    print("Pilih")
    print("1. Enkripsi")
    print("2. Dekripsi")

    action = input("Pilih aksi (1/2): ")

    message = input("Masukkan pesan: ")
    key = input("Masukkan kunci (integer dan string / integer saja / string saja): ")

    # Menyamakan panjang pesan dan kunci
    key = equalize_length(message, key)

    # Konversi karakter ke bit
    message_bits = ''.join([char_to_bin(char, bit_length=8) for char in message])
    key_bits = ''.join([char_to_bin(char, bit_length=8) 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 kunci:")
for i, char in enumerate(message):
    xor_result = xor_bits(message_bits[i*8:(i+1)*8],
key_bits[i*8:(i+1)*8])
    print(f"{message_bits[i*8:(i+1)*8]} XOR {key_bits[i*8:(i+1)*8]} =
{xor_result}")

if action == '1':
    # Enkripsi
    encrypted_bits = encrypt_bitwise(message_bits, key_bits)

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

    print("Hasil ada di hasil.txt")
    with open("hasil.txt", "w") as file:
        file.write("Encrypt Bit: " + encrypted_bits + "\n")
        file.write("Encrypt Message: " + encrypted_message)
elif action == '2':
    # Dekripsi
    decrypted_bits = decrypt_bitwise(message_bits, key_bits)

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

    print("Hasil ada di hasil.txt")
    with open("hasil.txt", "w") as file:
        file.write("Decrypt Bit: " + decrypted_bits + "\n")
        file.write("Decrypt Message: " + decrypted_message)
else:
    print("Pilihan aksi tidak valid.")

if __name__ == "__main__":
    main()

```

Penjelasan Algoritma

Program ini adalah implementasi enkripsi dan dekripsi menggunakan operasi XOR berbasis ASCII dengan panjang bit per karakter sebanyak 8 bit. Dimana nanti dapat memilih antara melakukan enkripsi atau dekripsi serta memasukkan pesan dan kunci, apabila pesan

Penjelasan hasil output Enkripsi

```
Enkripsi dan Dekripsi berbasis ASCII 8 BIT
Pilih
1. Enkripsi
2. Dekripsi
Pilih aksi (1/2): 1
```

Masukkan pesan: Kota Surabaya

Masukkan kunci (integer dan string / integer saja / string saja): 1234W

```

Bit pesan:
Karakter K: 01001011
Karakter o: 01101111
Karakter t: 01110100
Karakter a: 01100001
Karakter r: 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

```

Bit pesan dan kunci

```
Hasil XOR dari bit pesan dan bit kunci:
01001011 XOR 00110001 = 1
01101111 XOR 00110010 = 1
01110100 XOR 00110011 = 1
01100001 XOR 00110100 = 1
00100000 XOR 01010111 = 1
01010011 XOR 00110001 = 1
01110101 XOR 00110010 = 1
01110010 XOR 00110011 = 1
01100001 XOR 00110100 = 1
01100010 XOR 01010111 = 1
01100001 XOR 00110001 = 1
01111001 XOR 00110010 = 1
01100001 XOR 00110011 = 1
Hasil ada di hasil.txt
PS C:\Users\EMIL\Documents\SEMESTER 5\KEAMANAN INFOR
```

Proses enkripsi berbasis XOR dari program ini mengubah setiap karakter pesan dan kunci menjadi representasi biner dengan panjang tetap 8-bit. Panjang kunci kemudian sama dengan panjang pesan, dan operasi XOR bit per bit dilakukan secara berurutan antara representasi biner dari karakter pesan dan kunci. Hasil XOR dari setiap pasangan bit digunakan untuk membentuk pesan terenkripsi, dimana dengan cara jika 1 1 atau 0 0 hasilnya 0 dan jika 1 0 atau 0 1 hasilnya 1,

Hasil enkripsi disimpan di hasil.txt

```
hasil.txt
1 Encrypt Bit: 011110100101110101000111010101010111011101100010010001110100000101010100011010101010000010010110101
2 Encrypt Message: z]GUwbGAUSPKR
```

Penjelasan Hasil Output Dekripsi

Pilih dekripsi

```
python3/python.exe C:\Users\EMIL\Documents\SEMESTER 5\KEAMANAN INFOR
Enkripsi dan Dekripsi berbasis ASCII 8 BIT
Pilih
1. Enkripsi
2. Dekripsi
Pilih aksi (1/2): 2
```

Memasukkan pesan terenkripsi

```
Masukkan pesan: z]GUwbGAUSPKR
```

Memasukkan kunci

```
Masukkan kunci (integer dan string / integer saja / string saja): 1234W
```

Konversi ke biner dimana setiap karakter dari pesan dan kunci dikonversi ke representasi biner dengan panjang 8 bit yang menggunakan fungsi 'char_to_bin',

panjang kunci disamakan dengan panjang pesan yang menggunakan fungsi 'equalize_length'. Dilakukan operasi XOR bit per bit diantara representasi biner karakter pesan dan kunci secara berurutan yang menggunakan fungsi 'xor_bits'. Hasil XOR untuk setiap pasangan bit ditampilkan.

```
Bit pesan:
Karakter z: 01111010
Karakter j: 01011101
Karakter G: 01000111
Karakter U: 01010101
Karakter w: 01110111
Karakter b: 01100010
Karakter G: 01000111
Karakter A: 01000001
Karakter U: 01010101
Karakter S: 00110101
Karakter P: 01010000
Karakter K: 01001011
Karakter R: 01010010
```

```
Hasil XOR dari bit pesan dan bit kunci:
01111010 XOR 00110001 = 1
01011101 XOR 00110010 = 1
01000111 XOR 00110011 = 1
01010101 XOR 00110100 = 1
01110111 XOR 01010111 = 1
01100010 XOR 00110001 = 1
01000111 XOR 00110010 = 1
01000001 XOR 00110011 = 1
01010101 XOR 00110100 = 1
00110101 XOR 01010111 = 1
01010000 XOR 00110001 = 1
01001011 XOR 00110010 = 1
01010010 XOR 00110011 = 1
Hasil ada di hasil.txt
```

Menghasilkan pesan terdekripsi Kota Surabaya, yang disimpan di hasil.txt

```
hasil.txt
1 Decrypt Bit: 0100101101101111011101000110000100100000010100110111010111001001100001011000100110010110
2 Decrypt Message: Kota Surabaya
```

Kode ASCII -8bit

<https://repository.unikom.ac.id/54747/1/ASCII%208%20bit.pdf>

Link Github

<https://github.com/3milia123/XOR.git>