

LAPORAN IMPLEMENTASI
CYPER KLASIK MENGGUNAKAN PYTHON
MATA KULIAH KRIPTOGRAFI
Dosen Pengampu: Kodrat Mahatma



Disusun oleh: Kelompok 5

Siti Fatimah (20123062)

Jesi Rosyanti (20123053)


PROGRAM STUDI INFOMATIKA S1
UNIVERSITAS TEKNOLOGI DIGITAL

2025

Dasar Teori Singkat

1. Caesar Cipher

Caesar Cipher adalah pergeseran huruf asli menjadi huruf baru berdasarkan jarak tertentu dalam alfabet. Misalnya jika $n=3$, maka ABC menjadi DEF.

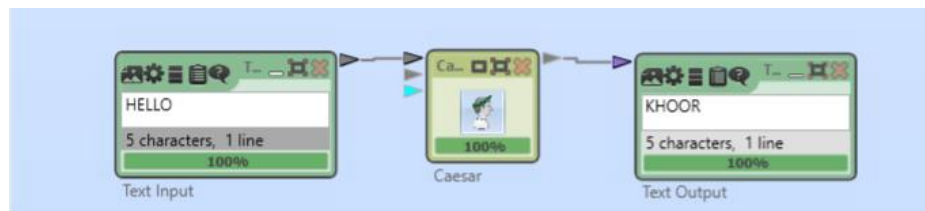


```
def caesar_encrypt(text, shift):
    result = ''
    for char in text:
        if char.isalpha():
            base = ord('A') if char.isupper() else ord('a')
            result += chr((ord(char) - base + shift) % 26 + base)
        else:
            result += char
    return result
print(caesar_encrypt('HELLO', 3))
```

KHOOR

Kode diatas merupakan pergeseran sebanyak tiga posisi ke sebelah kanan dalam alfabet. Dimana jika karakter bukan berupa huruf maka tidak diubah. Dari hasil kode tersebut menunjukkan bahwa plaintext HELLO berhasil di enkripsi menjadi Ciphertext KHOOR dengan melalui proses enkripsi sebagai berikut:

- Huruf H digeser 3 langkah menjadi K
- Huruf E digeser 3 langkah menjadi H
- Huruf L digeser 3 langkah menjadi O
- Huruf L digeser 3 langkah menjadi O
- Huruf O digeser 3 langkah menjadi R



Gambar diatas menunjukkan bahwa hasil enkripsi pada Cryptool 2 menghasilkan Ciphertext KHOOR, sama dengan hasil program python yang dikerjakan di google collab.

2. Vigenere Cipher

Vigenere Cipher adalah metode enkripsi yang menggunakan kata kunci (key) untuk menentukan jumlah pergeseran tiap huruf pada plaintext. Setiap huruf pada kunci akan menentukan nilai geseran huruf pada teks asli, sehingga pergeserannya tidak tetap seperti Caesar Cipher.

```
def vigenere_encrypt(plain, key):  
    key = key.upper()  
    result = ''  
    for i, char in enumerate(plain.upper()):  
        if char.isalpha():  
            shift = ord(key[i % len(key)]) - 65  
            result += chr((ord(char) - 65 + shift) % 26 + 65)  
        else:  
            result += char  
    return result  
print(vigenere_encrypt('ATTACKATDAWN', 'LEMON'))  
  
LXFOPVEFRNHR
```

Kode diatas membaca setiap huruf pada plaintext, lalu menentukan jumlah pergeseran berdasarkan huruf pada kunci (key). Jika kuncinya adalah LEMON, maka huruf pertama plaintext digeser sesuai huruf L, huruf kedua sesuai E, dan seterusnya.

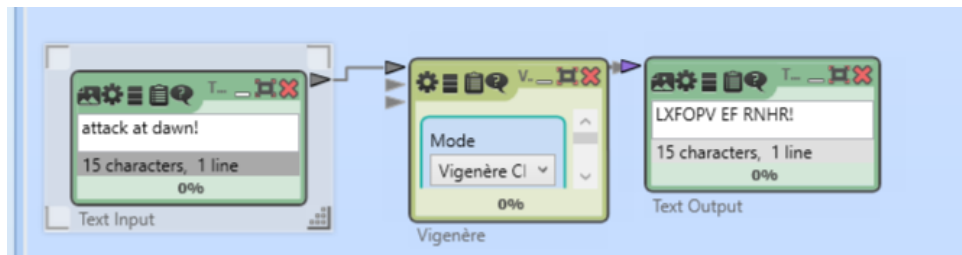
Proses enkripsi:

Plaintext : ATTACKATDAWN

Key : LEMONLEMONLE

Dengan perhitungan:

- $A (0) + L (11) = L (11)$
- $T (19) + E (4) = X (23)$
- $T (19) + M (12) = F (5)$
- $A (0) + O (14) = O (14)$
- $C (2) + N (13) = P (15)$
- Dan seterusnya....



Gambar diatas menunjukkan bahwa hasil enkripsi pada Cryptool 2 menghasilkan Ciphertext LXFOPVEFRNHR, dimana hasilnya sama dengan program python yang dikerjakan di google collab.

3. Affine Cipher

Affine chipper adalah salah satu bentuk substitusi Cipher yang menggunakan fungsi matematis untuk mengenkripsi huruf pada teks. Setiap huruf diubah menjadi angka, dikalikan dengan parameter a, ditambah b, kemudian hasilnya diubah menjadi huruf dengan operasi modulo 26.

```

1  def affine_encrypt(text, a, b):
    result = ''
    for char in text.upper():
        if char.isalpha():
            result += chr(((a * (ord(char) - 65) + b) % 26) + 65)
        else:
            result += char
    return result
    print(affine_encrypt('HELLO', 5, 8))

```

RCLLA

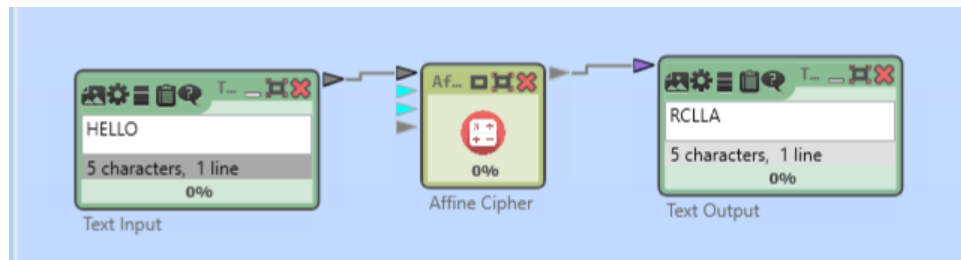
Pada kode diatas, setiap huruf plaintext dikonversi ke nilai angka dengan urutan A=0, B=1, C=2, dan seterusnya.

Selanjutnya dilakukan perhitungan dengan rumus:

$$C = (a \times P + b) \bmod 26$$

Di mana:

- C adalah huruf hasil enkripsi (Ciphertext)
- P adalah huruf asli (plaintext)
- A dan b adalah kunci enkripsi



Gambar diatas merupakan hasil cryptool 2 yang telah dilakukan, hasilnya menunjukkan enkripsi yang sama dengan yang dilakukan pada program google collab yaitu, RCLLA, sehingga dapat disimpulkan bahwa implementasi program python bekerja dengan benar.

4. Playfair Cipher

Playfair Cipher adalah salah satu algoritma kriptografi klasik yang digunakan untuk mengamankan pesan dengan cara melakukan substitusi pasangan huruf (digraph), bukan per-huruf seperti Caesar atau Vigenère. Oleh karena itu chipr ini dianggap lebih kuat dari Caesar Cipher karena melakukan enkripsi berdasarkan dua huruf sekaligus sehingga pola frekuensi hurufnya lebih sulit dianalisis.

```
def generate_table(key):
    alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    table = ''
    for c in key.upper() + alphabet:
        if c not in table:
            table += c
    return [table[i:i+5] for i in range(0,25,5)]
table = generate_table('KEYWORD')
for row in table: print(row)
```

```
KEYWO
RDABC
FGHIL
MNPQS
TUVXZ
```

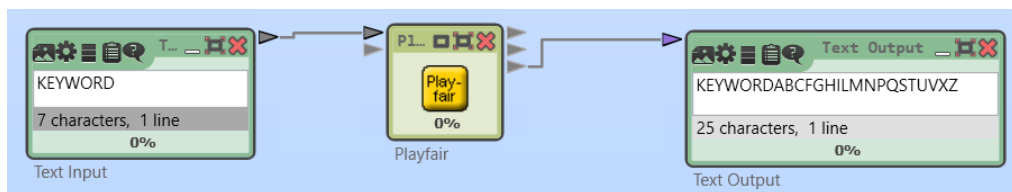
Proses pertama Playfair Cipher adalah menyusun tabel 5×5 berdasarkan kata kunci (keyword). Kata kunci dituliskan terlebih dahulu ke dalam tabel, kemudian diikuti huruf alfabet yang belum muncul dalam keyword. Di Cipher

ini huruf I dan J digabungkan menjadi satu entitas karena tabel hanya memiliki 25 kotak, bukan 26.

Adapun kata kunci yang digunakan yaitu 'KEYWORD', maka menghasilkan tabel Playfair:

- K E Y W O
- R D A B C
- F G H I L
- M N P Q S
- T U V X Z

Hasil diatas sesuai dengan implementasi yang dilakukan di Ccryptool 2, dimana urutan hasilnya sesuai walaupun yang ditampilkan dalam bentuk satu baris.



5. Hill Cipher

Hill Cipher merupakan salah satu teknik kriptografi klasik berbasis aljabar linear. Hill Cipher bekerja dengan cara mengubah teks menjadi bentuk vektor angka, kemudian melakukan operasi perkalian matriks menggunakan kunci (key matrix), dan hasilnya dikonversi kembali menjadi huruf.

```
import numpy as np
def hill_encrypt(text, key):
    text = text.upper().replace(' ', '')
    n = int(len(key)**0.5)
    key = np.array(key).reshape(n, n)
    result = ''
    for i in range(0, len(text), n):
        block = [ord(c) - 65 for c in text[i:i+n]]
        cipher = np.dot(key, block) % 26
        result += ''.join(chr(c + 65) for c in cipher)
    return result
print(hill_encrypt('TEST', [3,3,2,5]))
```

RGHB

Dalam percobaan ini teks yang digunakan yaitu 'TEST' dengan kata kunci berbentuk matriks 2x2. Sebelum dilakukan enkripsi, setiap huruf pada plaintext diubah ke angka sesuai posisinya dalam alfabet (A = 0, B = 1, ..., Z = 25. Maka: T = 19, E = 4, S = 18, T = 19. Kemudian teks dibagi menjadi blok berukuran 2 huruf (sesuai ukuran matriks kunci) sehingga [19,4] dan [18, 19].

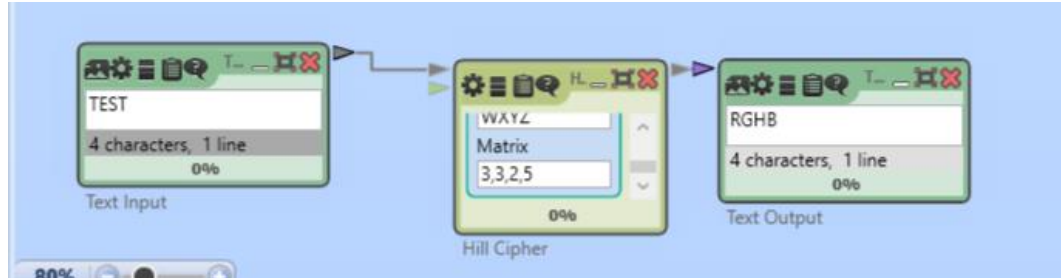
Selanjutnya dilakukan operasi enkripsi dengan rumus:

$$C = K \cdot P \text{ mod } 26$$

Di mana:

- C = Ciphertext (hasil)
- K = Key matrix
- P = Plaintext dalam bentuk vektor angka

Sehingga menampilkan hasil Ciphertext: RGHB.



Sesuai dengan validasi penggunaan CrypTool 2, proses enkripsi bisa dikatakan berhasil dan benar.