

Nama : Fazar Rizwanul Ikhlas

NPM : 20123065

Kelas : C.2.23

Kriptografi – Penugasan

Implementasikan Vigenère Cipher dan analisis frekuensinya.

1. Tujuan

- a. Mengimplementasikan algoritma Vigenère Cipher untuk proses enkripsi dan dekripsi.
- b. Melakukan analisis frekuensi terhadap ciphertext untuk mengidentifikasi pola distribusi huruf.
- c. Melakukan validasi hasil enkripsi dan dekripsi menggunakan aplikasi CrypTool

2. Dasar Teori

Cipher polialfabetik merupakan algoritma kriptografi klasik yang menggunakan lebih dari satu alfabet substitusi dalam proses enkripsi. Salah satu cipher polialfabetik paling terkenal adalah Vigenère Cipher.

Vigenère Cipher menggunakan kunci berupa kata untuk menentukan pergeseran huruf pada plaintext. Setiap huruf plaintext digeser berdasarkan nilai huruf kunci.

Rumus matematis:

a. Enkripsi

$$C_i = (P_i + K_i) \bmod 26$$

b. Deskripsi

$$P_i = (C_i - K_i) \bmod 26$$

Keterangan:

- C_i = ciphertext
- P_i = plaintext
- K_i = kunci ($A=0, B=1, \dots, Z=25$)

Contoh:

Plaintext: ATTACKATDAWN

Key: LEMON

Ciphertext: LXFOPVEFRNHR

3. Implementasi Program

Kode Program Python

```
1 # -----
2 # VIGENERE CIPHER IMPLEMENTATION
3 # -----
4
5 # Fungsi Enkripsi
6 def vigenere_encrypt(plain, key):
7     plain = plain.upper().replace(" ", "")
8     key = key.upper()
9     res = ''
10    for i in range(len(plain)):
11        res += chr(((ord(plain[i]) - 65 + ord(key[i % len(key)])) - 65) % 26) + 65)
12    return res
13
14 # Fungsi Dekripsi
15 def vigenere_decrypt(cipher, key):
16     cipher = cipher.upper().replace(" ", "")
17     key = key.upper()
18     res = ''
19     for i in range(len(cipher)):
20         res += chr(((ord(cipher[i]) - ord(key[i % len(key)]))) % 26) + 65)
21     return res
22
23 # Uji Program
24 plain = "HELLO WORLD"
25 key = "LEMON"
26
27 cipher = vigenere_encrypt(plain, key)
28 print("Plaintext :", plain)
29 print("Key      :", key)
30 print("Ciphertext:", cipher)
31
32 decrypted = vigenere_decrypt(cipher, key)
33 print("Dekripsi  :", decrypted)
34
35 from collections import Counter
36 import matplotlib.pyplot as plt
37
38 def frequency_analysis(text):
39     text = text.upper().replace(" ", "")
40     count = Counter(text)
41     print("\n==== Analisis Frekuensi Huruf ===")
42     for k, v in count.items():
43         print(f"{k}: {v/len(text):.2f}")
44
45     # Visualisasi
46     plt.bar(count.keys(), [v/len(text) for v in count.values()])
47     plt.title("Analisis Frekuensi Ciphertext")
48     plt.xlabel("Huruf")
49     plt.ylabel("Frekuensi Relatif")
50     plt.show()
51
52 # Panggil fungsi untuk ciphertext yang dihasilkan
53 frequency_analysis(cipher)
54
```

4. Hasil Pengujian

Input

- Plaintext : HELLO WORLD
- Key : LEMON

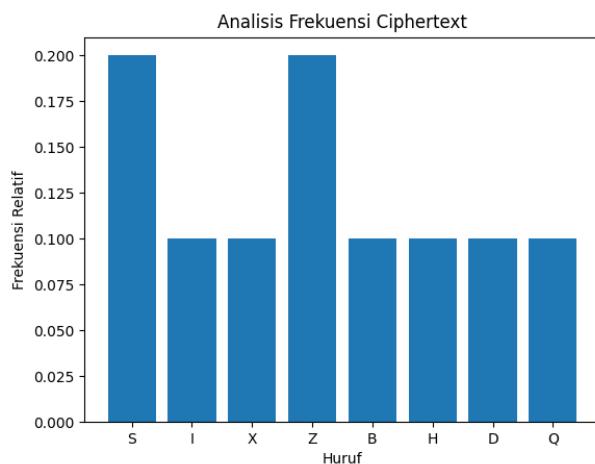
Output Program

```
(base) C:\Users\LENOVO\Documents\SEMESTER 4\Kriptografi\PT-3>python vigenere_cipher.py
Plaintext : HELLO WORLD
Key       : LEMON
Ciphertext: SDXZBHSQDZQ
Dekripsi  : HELLOWORLD

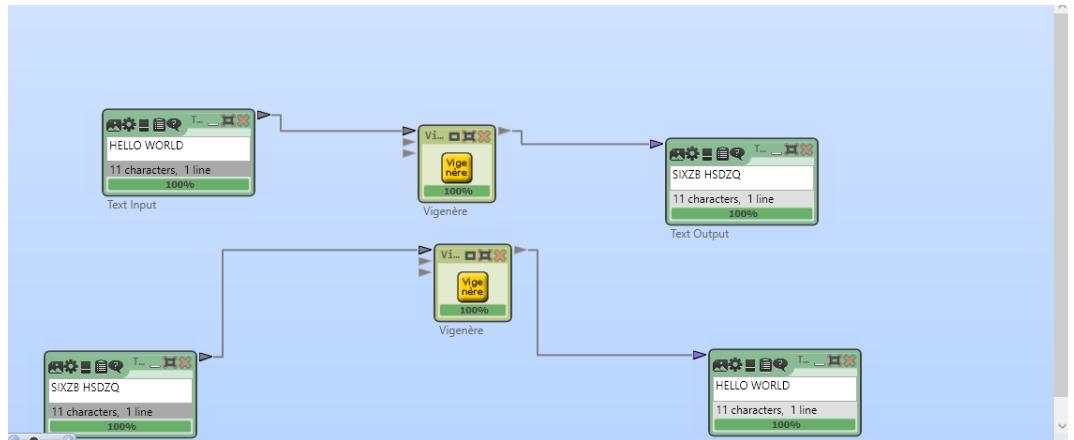
==== Analisis Frekuensi Huruf ===
S: 0.20
I: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
(base) C:\Users\LENOVO\Documents\SEMESTER 4\Kriptografi\PT-3>python vigenere_cipher.py
Plaintext : HELLO WORLD
Key       : LEMON
Ciphertext: SDXZBHSQDZQ
Dekripsi  : HELLOWORLD

==== Analisis Frekuensi Huruf ===
S: 0.20
I: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
Key       : LEMON
Ciphertext: SDXZBHSQDZQ
Dekripsi  : HELLOWORLD
```

```
==== Analisis Frekuensi Huruf ===
S: 0.20
I: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
S: 0.20
I: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
S: 0.20
I: 0.10
X: 0.10
Z: 0.20
B: 0.10
H: 0.10
D: 0.10
H: 0.10
D: 0.10
D: 0.10
Q: 0.10
```



5. Validasi Menggunakan Cryptool



6. Kesimpulan

- Vigenère Cipher merupakan algoritma cipher polialfabetik yang lebih aman dibanding Caesar Cipher karena menggunakan pergeseran dinamis berdasarkan key.
- Ciphertext yang dihasilkan memiliki frekuensi huruf yang lebih merata, sehingga lebih sulit diretas menggunakan analisis frekuensi biasa.
- Hasil program Python sesuai dan tervalidasi dengan CrypTool.