

LAPORAN ANALISIS KELEMAHAN VIGENERE CIPHER



Universitas Teknologi Digital

Disusun Oleh :

1. Mochammad Rival Sopyan - 20123006
2. Fernanda Syah Putra - 20123019
3. Muhamad Rifky Raihan - 20123021

PROGRAM STUDI S1 INFORMATIKA
UNIVERSITAS TEKNOLOGI DIGITAL
TAHUN AJARAN 2024/2025

VIGENERE CIPHER

Sorcecode

```
1  # Implementasi Vigenère Cipher
2
3  def vigenere_encrypt(plain, key):
4      plain = plain.upper().replace(" ", "")
5      key = key.upper()
6      result = ""
7      for i in range(len(plain)):
8          p = ord(plain[i]) - 65
9          k = ord(key[i % len(key)]) - 65
10         c = (p + k) % 26
11         result += chr(c + 65)
12     return result
13
14
15  def vigenere_decrypt(cipher, key):
16      cipher = cipher.upper().replace(" ", "")
17      key = key.upper()
18      result = ""
19      for i in range(len(cipher)):
20          c = ord(cipher[i]) - 65
21          k = ord(key[i % len(key)]) - 65
22          p = (c - k) % 26
23          result += chr(p + 65)
24     return result
25
26
27  # Contoh uji
28  plaintext = "CRYPTOGRAPHY"
29  key = "LEMON"
30  ciphertext = vigenere_encrypt(plaintext, key)
31  decrypted = vigenere_decrypt(ciphertext, key)
32
33  print("Plaintext :", plaintext)
34  print("Key       :", key)
35  print("Ciphertext:", ciphertext)
36  print("Dekripsi  :", decrypted)
37
```

Hasil Pembuktian

Plaintext : CRYPTOGRAPHY

Key : LEMON

Ciphertext : NVKDGZKDOCSC

Dekripsi : CRYPTOGRAPH

VIGENERE CIPHER FREKUENSI

Sorcecode

```
1  # Implementasi Vigenère Cipher
2
3  def vigenere_encrypt(plain, key):
4      plain = plain.upper().replace(" ", "")
5      key = key.upper()
6      result = ""
7      for i in range(len(plain)):
8          p = ord(plain[i]) - 65
9          k = ord(key[i % len(key)]) - 65
10         c = (p + k) % 26
11         result += chr(c + 65)
12     return result
13
14
15  def vigenere_decrypt(cipher, key):
16      cipher = cipher.upper().replace(" ", "")
17      key = key.upper()
18      result = ""
19      for i in range(len(cipher)):
20          c = ord(cipher[i]) - 65
21          k = ord(key[i % len(key)]) - 65
22          p = (c - k) % 26
23          result += chr(p + 65)
24     return result
25
26
27  # Contoh uji
28  plaintext = "CRYPTOGRAPHY"
29  key = "LEMON"
30  ciphertext = vigenere_encrypt(plaintext, key)
31  decrypted = vigenere_decrypt(ciphertext, key)
32
33  print("Plaintext :", plaintext)
34  print("Key       :", key)
35  print("Ciphertext:", ciphertext)
36  print("Dekripsi  :", decrypted)
37
```

Analisis Frekuensi:

N : 0.17

F : 0.17

X : 0.08

Z : 0.17

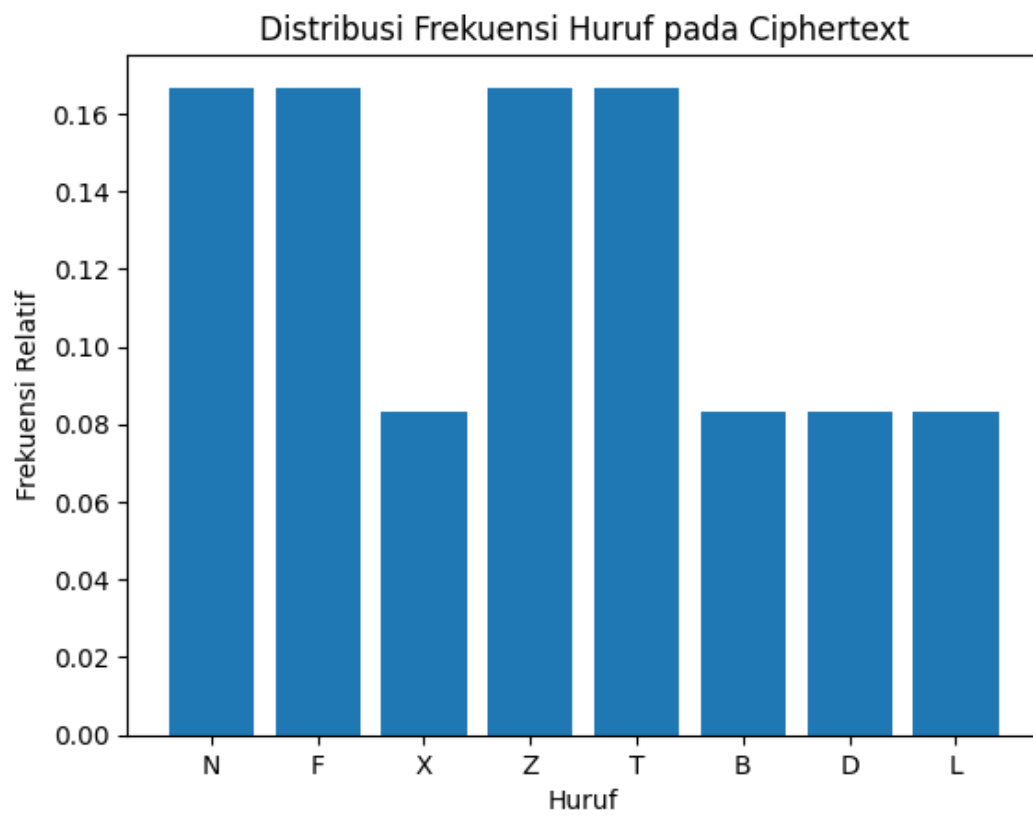
T : 0.17

B : 0.08

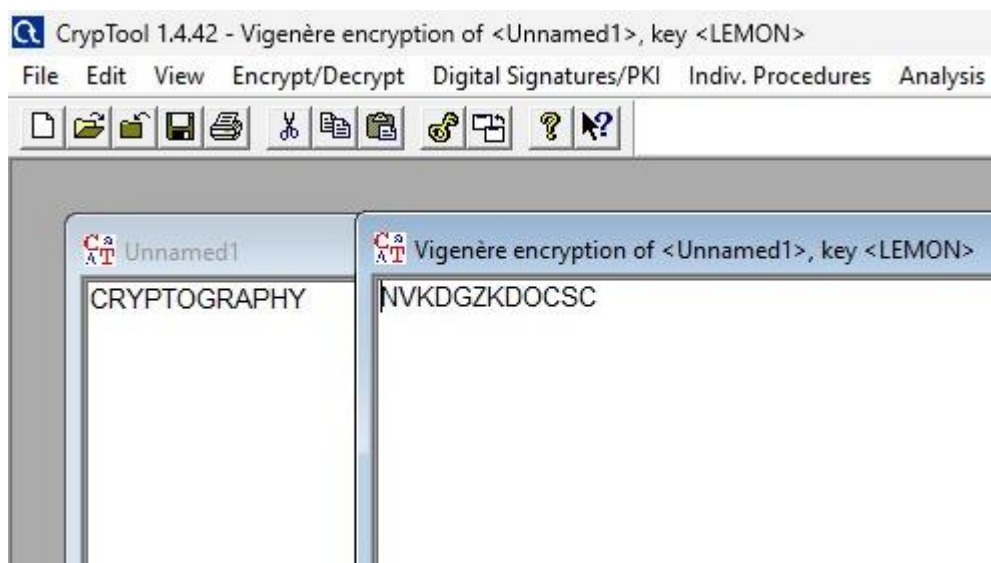
D : 0.08

L : 0.08

Grafik



Validasi CRYTOOL



Hasil validasi menggunakan **CrypTool** menunjukkan ciphertext yang dihasilkan yaitu **“NVKDGZKDOCSC”** dari plaintext **“CRYPTOGRAPHY”** dengan kunci **“LEMON”**. Hasil ini sama persis dengan keluaran dari program Python pada gambar kode implementasi Vigenère Cipher yang digunakan, sehingga dapat dipastikan bahwa proses enkripsi dan dekripsi pada program sudah **berhasil dan akurat**.

Selain itu, grafik distribusi frekuensi huruf pada ciphertext menunjukkan bahwa huruf-huruf seperti **N, F, Z, dan T** memiliki frekuensi relatif yang hampir seimbang, tanpa ada satu huruf yang mendominasi. Pola ini berbeda dengan distribusi frekuensi plaintext bahasa Inggris yang biasanya tidak merata, di mana huruf seperti **E** atau **T** sering muncul lebih sering.

Perbedaan pola ini menegaskan bahwa **algoritma Vigenère Cipher bekerja dengan baik dalam menyamarkan pola huruf asli**, sehingga hasil enkripsi sulit dianalisis hanya dengan metode frekuensi. Dengan demikian, baik dari **hasil ciphertext yang sesuai** maupun dari **pola distribusi frekuensi yang acak**, dapat disimpulkan bahwa **validasi CrypTool mendukung keberhasilan implementasi Vigenère Cipher pada program Python**.