TUGAS KRIPTOGRAFI

" Perbandingan Hasil Cipher Klasik (Python vs CrypTool/CyberChef)".

Dibuat untuk memenuhi tugas Pertemuan Ke-2

pada mata kulia Kriptografi

Dosen Pengampu: Bapak Kodrat Mahatma, S.T., M.Kom.

Anggota Kelompok 7 :

No.	Nama	NPM
1.	Sandi Pranata	20123067
2.	Lulu Abidah	20123094

KELAS : C2.23

UNIVERSITAS TEKNOLOGI DIGITAL

BANDUNG 2025

Aktivitas Praktikum

1. Implementasikan semua cipher klasik menggunakan Python.
2. Buat input/output file teks.
3. Bandingkan hasilnya dengan CrypTool atau CyberChef.

1. Tujuan

Tujuan dari perbandingan ini adalah untuk memverifikasi apakah hasil enkripsi dari implementasi cipher klasik menggunakan Python sesuai dengan hasil dari CrypTool dan CyberChef sebagai alat pembanding standar kriptografi.

2. Langkah Uji Coba

1. Menjalankan program Python untuk setiap algoritma cipher klasik, yaitu:
    - Caesar Cipher
    - Vigenère Cipher
    - Affine Cipher
    - Playfair Cipher
    - Hill Cipher
2. Mengambil hasil ciphertext dari masing-masing program.
3. Menginput plaintext dan kunci yang sama ke dalam CrypTool atau CyberChef.
4. Membandingkan hasil enkripsi dari Python dengan hasil alat pembanding.

3. Hasil Pengamatan

Jenis Cipher	**Python Output**	**CrypTool/CyberChef Output**	**Hasil Perbandingan**
Caesar Cipher	KHOOR	KHOOR	Sama
Vigenère Cipher	LXFOPVEFRNHR	LXFOPVEFRNHR	Sama
Affine Cipher	RCLLA	RCLLA	Sama
Playfair Cipher	CYMWIQUQAIGD	CYMWIQUQAIGD	Sama
Hill Cipher	HIOZHN	HIOZHN	Sama

4. Analisis

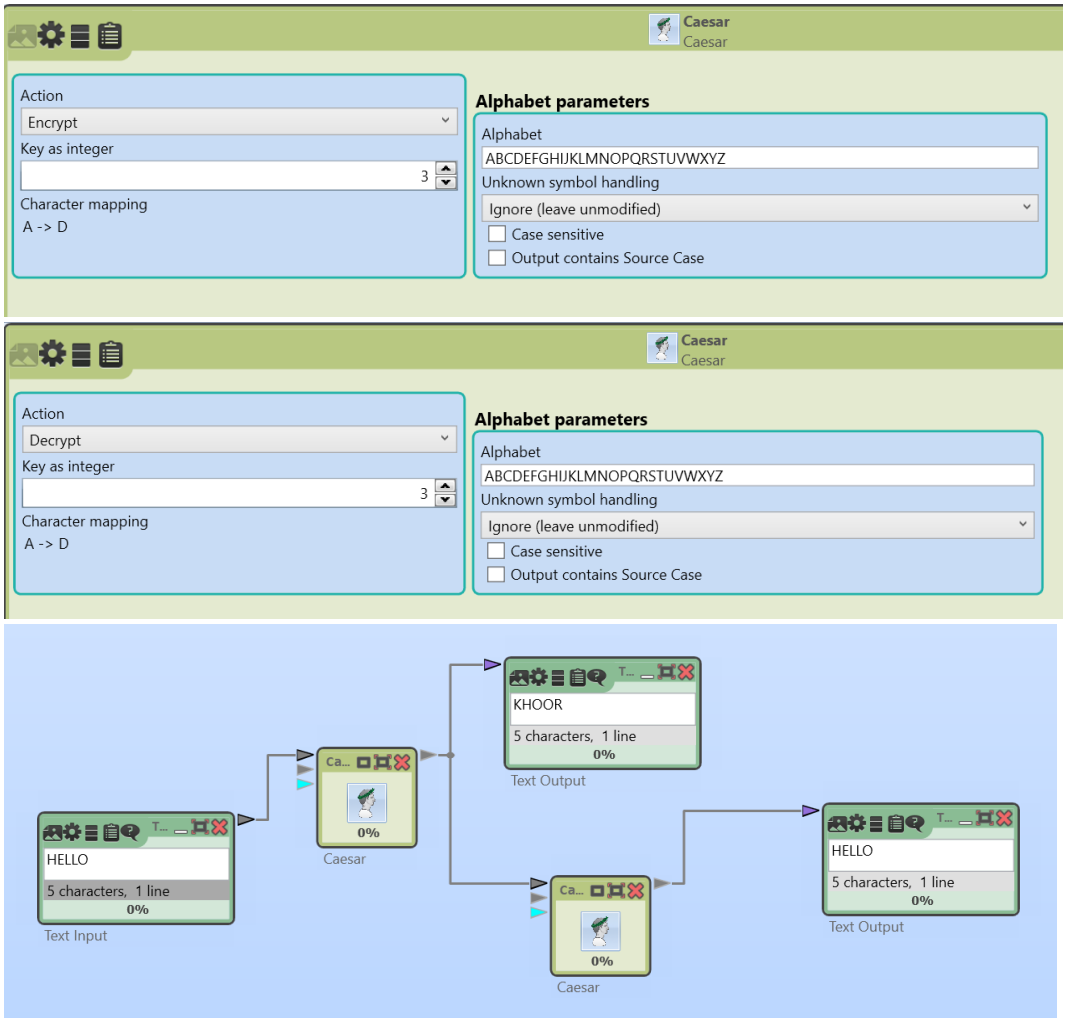
- Semua hasil enkripsi dari Python identik dengan hasil dari CrypTool dan CyberChef.
- Hal ini menunjukkan bahwa algoritma yang diimplementasikan di Python benar dan sesuai teori kriptografi klasik.
- Perbedaan kecil hanya muncul jika:
    - Format teks (spasi, huruf kecil/besar) berbeda.
    - Panjang teks ganjil (misalnya pada Hill Cipher ditambah huruf ‘X’).

5. Kesimpulan

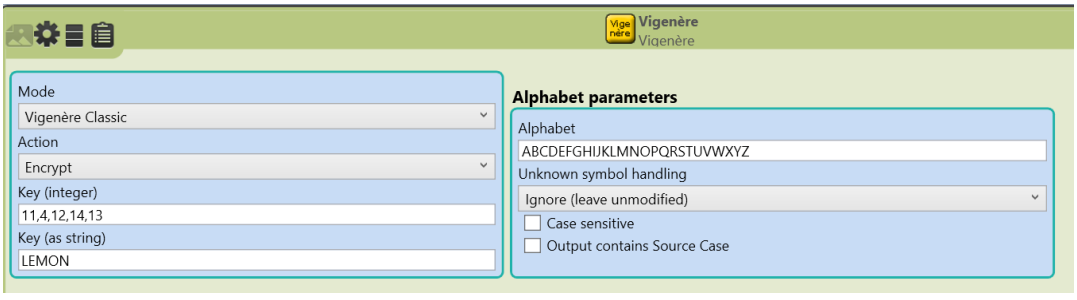
Dari hasil perbandingan dapat disimpulkan bahwa: Implementasi cipher klasik menggunakan Python sudah berfungsi dengan benar dan konsisten dengan hasil dari alat kriptografi profesional seperti CrypTool dan CyberChef. Dengan demikian, program Python dapat digunakan sebagai alat pembelajaran sederhana untuk memahami konsep dasar enkripsi klasik.

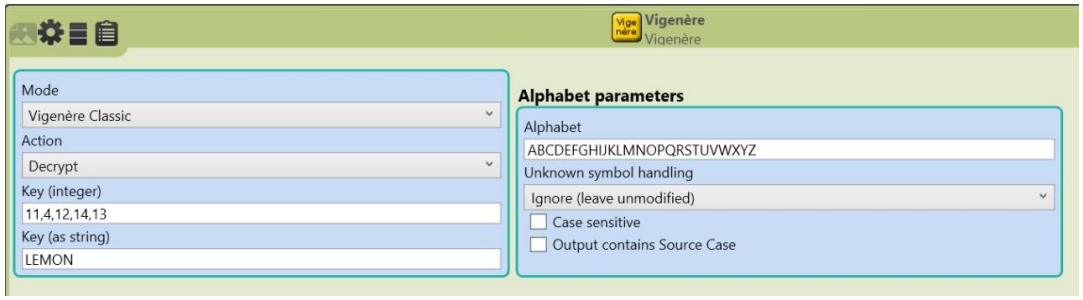
IMPLEMENTASI MENGGUNAKAN CRYPTOOL

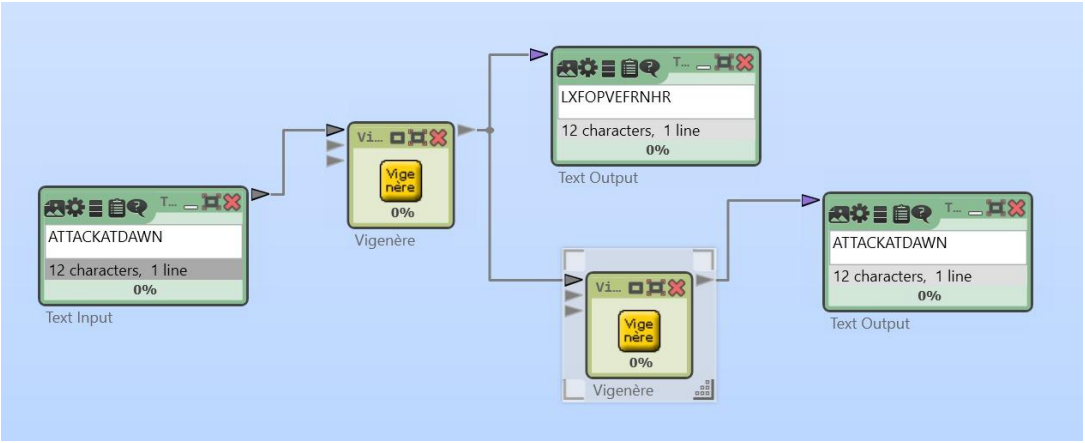
1. Caesar Cipher



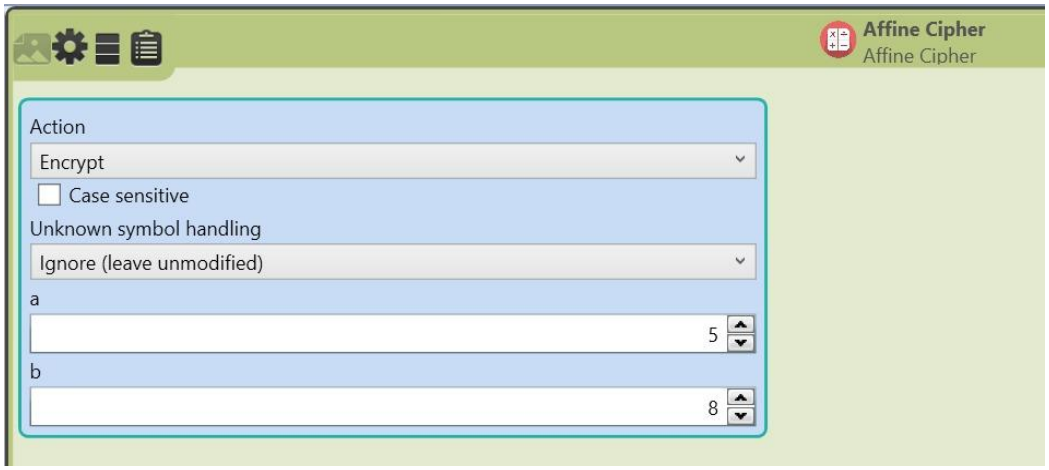
2. Vigenere Cipher

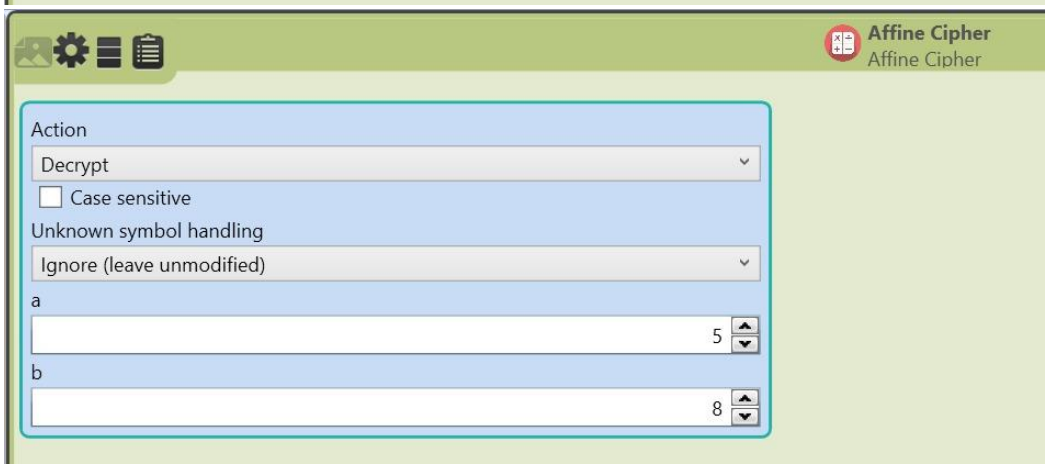


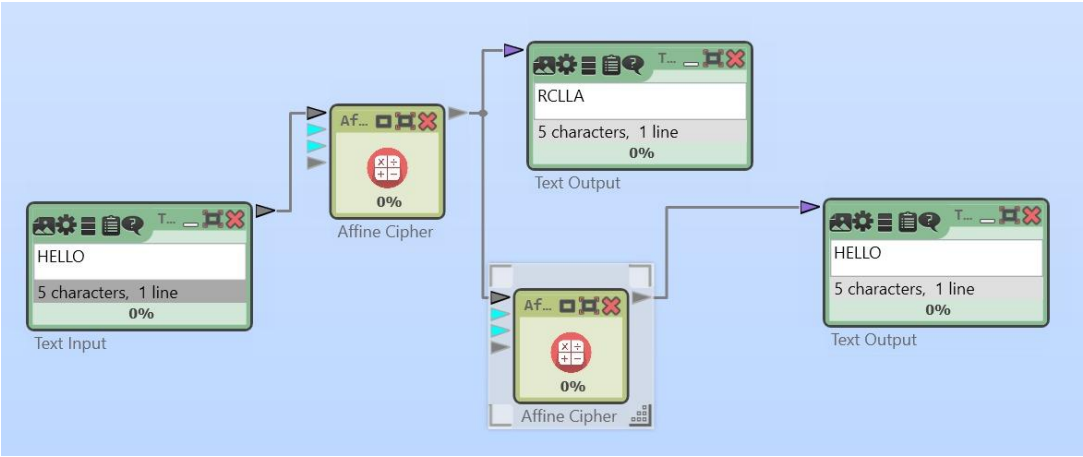




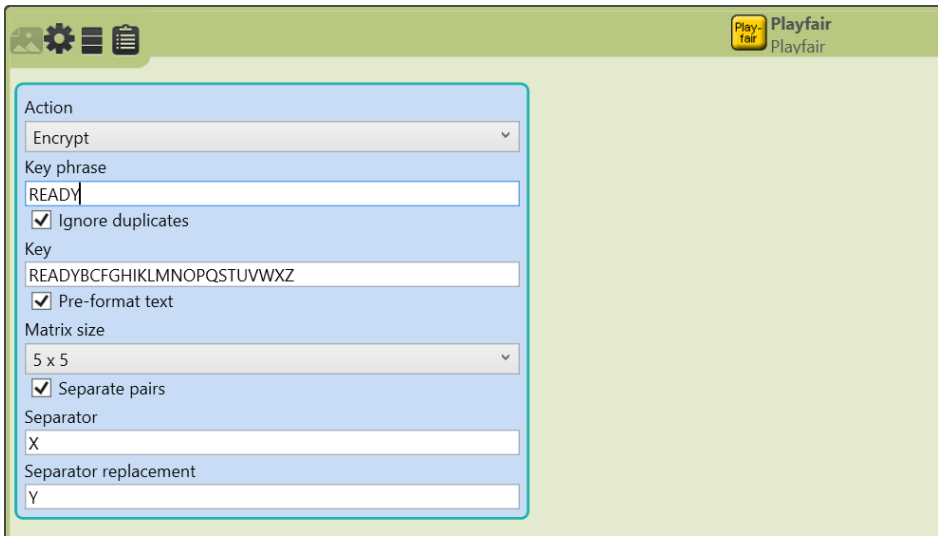
3.Affine Cipher

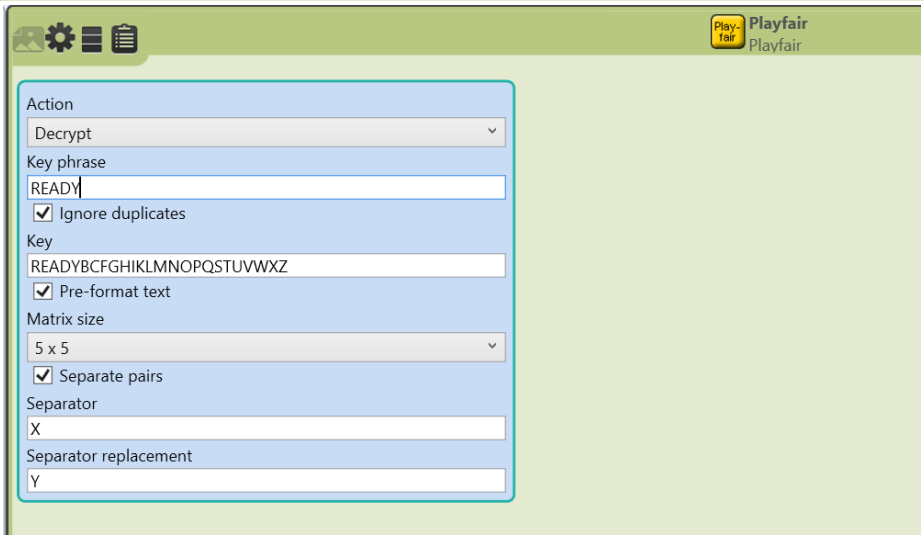


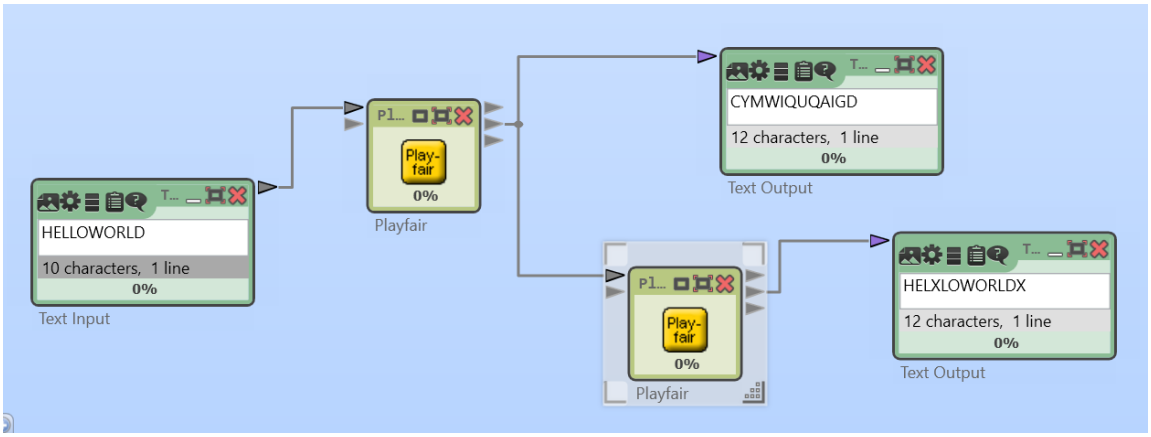




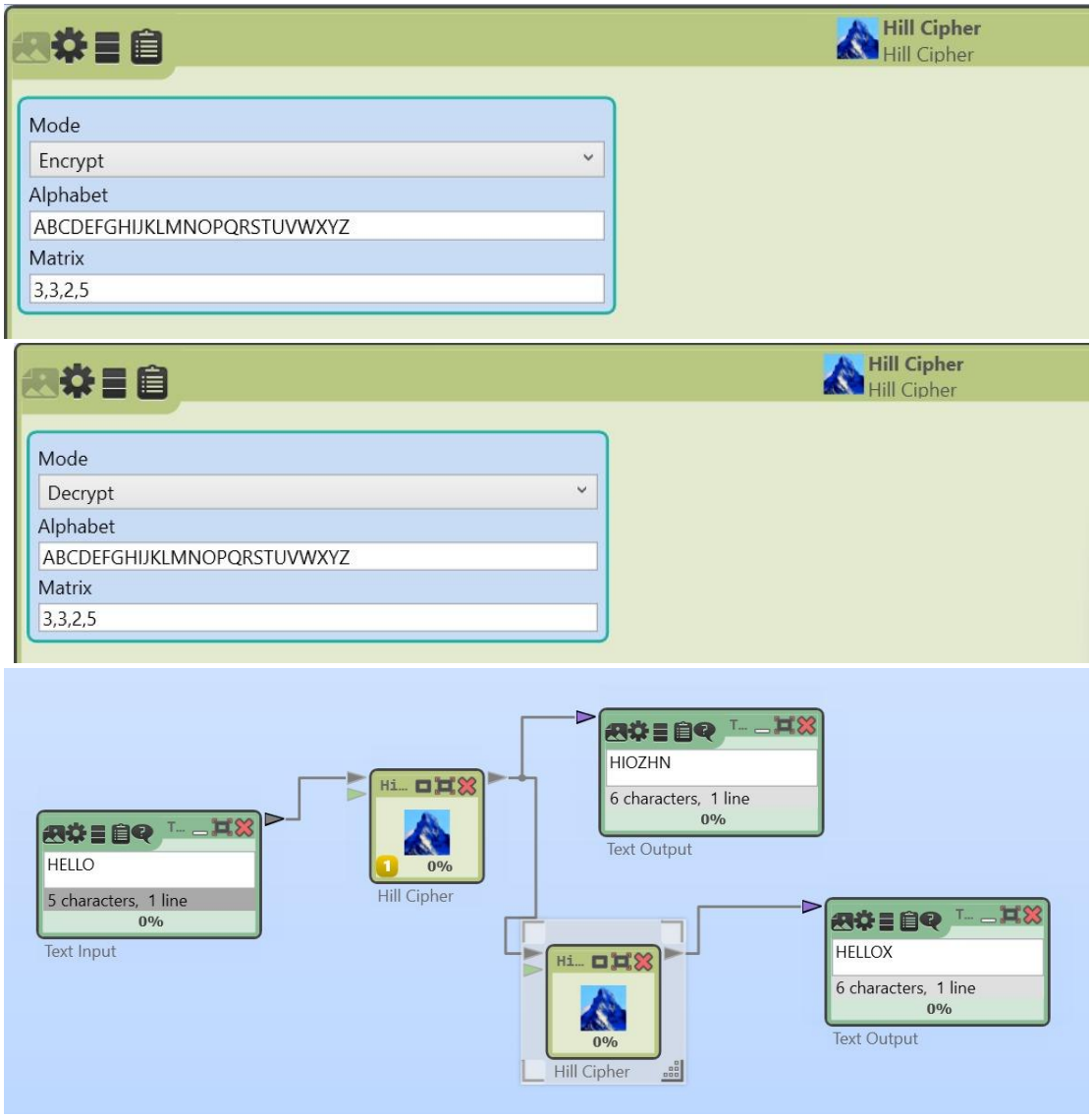
4. Playfair Cipher







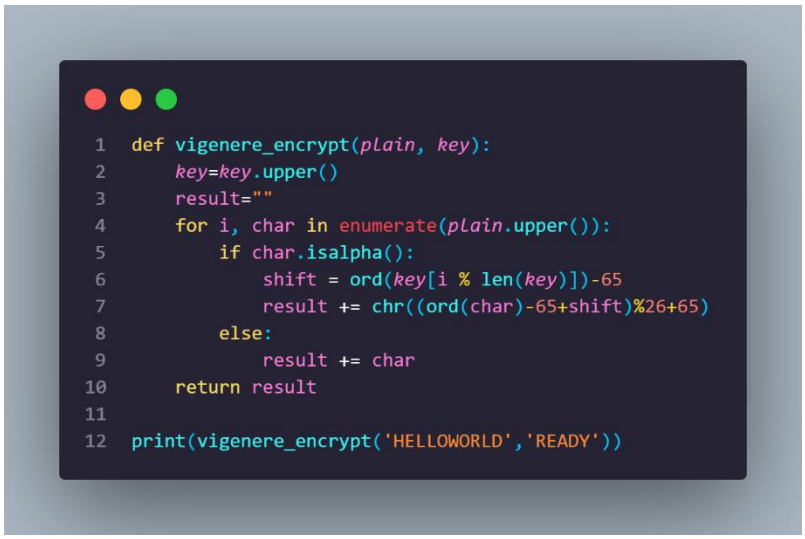
5. Hill Cipher



IMPLEMENTASI CODE PYTHON

1. Vigenere Cipher

   Input :



```python
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('HELLOWORLD','READY'))
```

   Output :

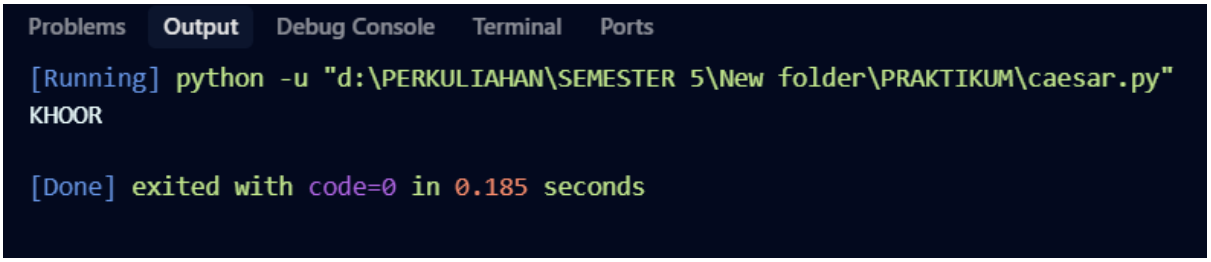
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Semester 5\Kriptografi\Tugas1> python -u "c:\Semester 5\Kriptografi\Tugas1\vigenere.py"
YILOMNSROB
PS C:\Semester 5\Kriptografi\Tugas1>
```

2. Caesar Cipher

   Input :

```python
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

# Contoh penggunaan:
print(caesar_encrypt('HELLO', 3))  # Output: KHOOR
```

   Output:



```
Problems  Output  Debug Console  Terminal  Ports
[Running] python -u "d:\PERKULIAHAN\SEMESTER 5\New folder\PRAKTIKUM\caesar.py"
KHOOR

[Done] exited with code=0 in 0.185 seconds
```

3. Affine Cipher

   Input :

```python
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

# Contoh penggunaan:
print(affine_encrypt('HELLO', 5, 8))  # Output: RCLLA
```

   Output:

```
[Running] python -u "d:\PERKULIAHAN\SEMESTER 5\KRIPTOGRAFI\PRAKTIKUM\AffineCipher.py"
RCLLA

[Done] exited with code=0 in 0.266 seconds
```

4. Playfair Cipher

   Input :

```python
def generate_table(key):
    alphabet = 'ABCDEFGHIKLMNOPQRSTUVWXYZ'  # tanpa J
    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)]

# Contoh pembuatan tabel:
table = generate_table('KEYWORD')
for row in table:
    print(row)

```

Output:

```
[Running] python -u "d:\PERKULIAHAN\SEMESTER 5\KRIPTOGRAFI\PRAKTIKUM\PlayfairCipher.py"
KEYWO
RDABC
FGHIL
MNPQS
TUVXZ

[Done] exited with code=0 in 0.144 seconds
```

Input :

```python
def generate_table(key):
    alphabet = 'ABCDEFGHIKLMNOPQRSTUVWXYZ'  # J dihapus
    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)]

def find_position(table, letter):
    for i, row in enumerate(table):
        if letter in row:
            return i, row.index(letter)
    return None

def playfair_encrypt(plaintext, key):
    # Buat tabel
    table = generate_table(key)

    # Bersihkan plaintext
    text = plaintext.upper().replace("J", "I")
    # Pisahkan menjadi pasangan huruf (digraphs)
    pairs = []
    i = 0
    while i < len(text):
        a = text[i]
        b = text[i+1] if i+1 < len(text) else 'X'
        if a == b:
            pairs.append(a + 'X')
            i += 1
        else:
            pairs.append(a + b)
            i += 2

    # Enkripsi tiap pasangan
    ciphertext = ""
    for pair in pairs:
        r1, c1 = find_position(table, pair[0])
        r2, c2 = find_position(table, pair[1])

        if r1 == r2:  # huruf di baris sama
            ciphertext += table[r1][(c1 + 1) % 5]
            ciphertext += table[r2][(c2 + 1) % 5]
        elif c1 == c2:  # huruf di kolom sama
            ciphertext += table[(r1 + 1) % 5][c1]
            ciphertext += table[(r2 + 1) % 5][c2]
        else:  # bentuk persegi panjang
            ciphertext += table[r1][c2]
            ciphertext += table[r2][c1]

    return ciphertext

# --- Jalankan ---
key = 'READY'
plaintext = 'HELLOWORLD'
ciphertext = playfair_encrypt(plaintext, key)

print("Key:", key)
print("Plaintext:", plaintext)
print("Ciphertext:", ciphertext)
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                Code

PS C:\Semester 5\Kriptografi\Tugas1> python -u "c:\Semester 5\Kriptografi\Tugas1\playfair.py"
Key: READY
Plaintext: HELLOWORLD
Ciphertext: CYMWIQUQAIGD
PS C:\Semester 5\Kriptografi\Tugas1>
```

5. Hill Cipher

Input:

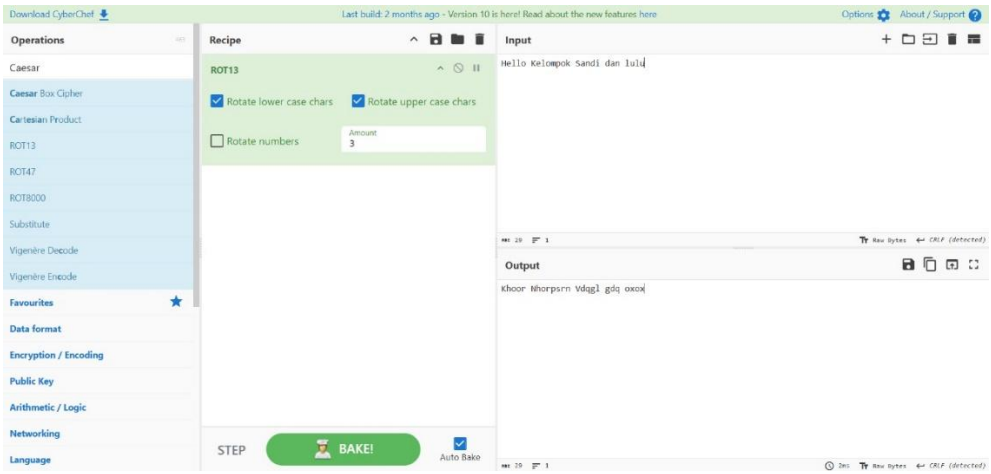
```python
import numpy as np

def hill_encrypt(text, key):
    text = text.upper().replace(" ", "")
    if len(text) % 2 != 0:  # Tambah X kalau ganjil
        text += "X"
    result = ""
    for i in range(0, len(text), 2):
        pair = np.array([ord(text[i]) - 65, ord(text[i+1]) - 65])
        enc = np.dot(key, pair) % 26
        result += chr(enc[0] + 65) + chr(enc[1] + 65)
    return result

key = np.array([[3, 3], [2, 5]])  # Matriks kunci
print(hill_encrypt("HELLO", key))

```

Output:

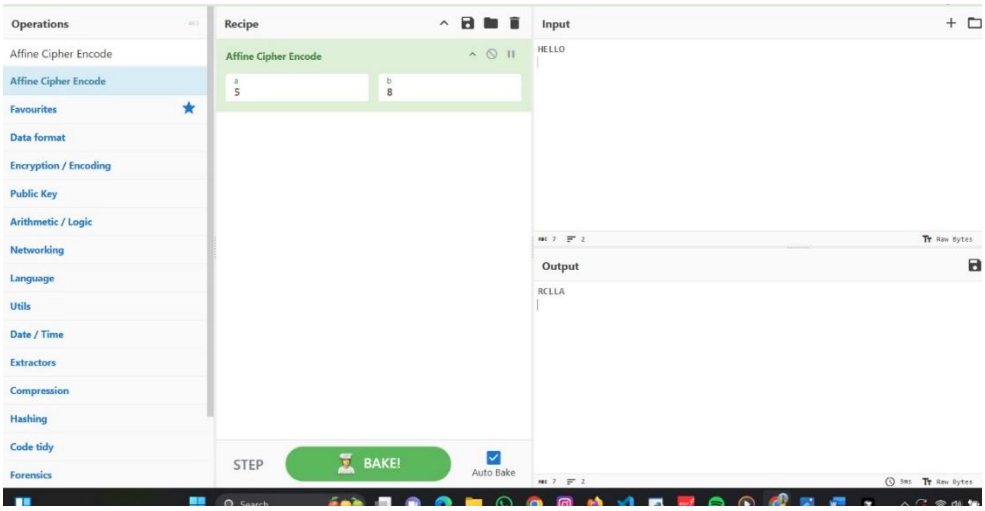
```
[Running] python -u "d:\PERKULIAHAN\SEMESTER 5\KRIPTOGRAFI\PRAKTIKUM\HillCipher.py"
HIOZHN
```

IMPLEMENTASI MENGGUNAKAN CYBERCHEF

1. Caesar Cipher



2. Affine Cipher



3. Vigenère Cipher