Laporan Implementasi Lima Algoritma Kriptografi Klasik

Dosen pengampu: Kodrat Mahatma

Disusun oleh :

Kelompok 8

Khabibah Wiendie Zahra (20123073)

Marshelindo Rizkina Ardhyansyah (20123080)

PROGRAM STUDI INDFORMATIKA

UNIVERSITAS TEKNOLOGI DIGITAL BANDUNG

2025

A. Mekanisme dan Implementasi kode

1. Affine Cipher

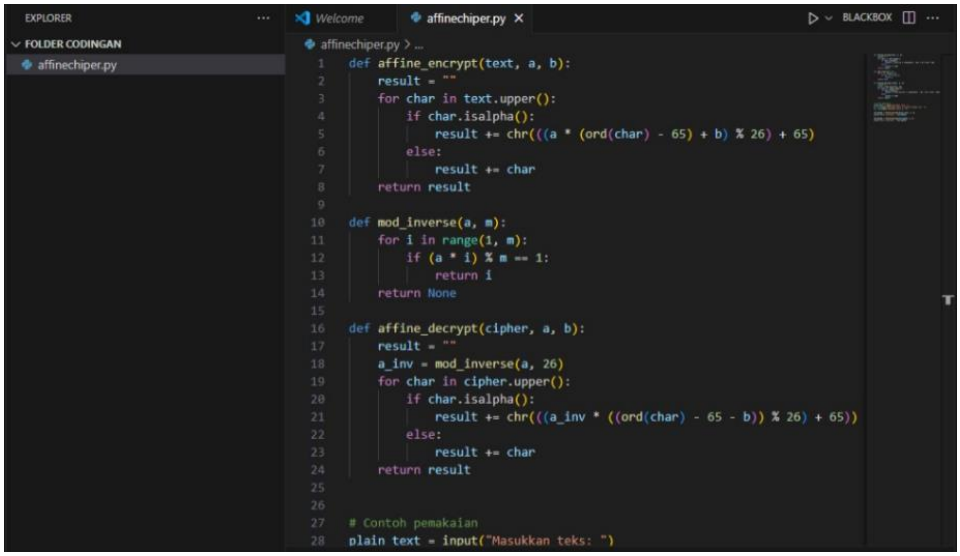
- **Mekanisme :** Menggunakan dua kunci dalam fungsi linier. Ini secara signifikan meningkatkan keragaman pemetaan awal dibandingkan Caesar.
- **Fungsi Kunci** : Pasangan kunci (a, b), di mana a adalah pengali dan b adalah pergeseran.

  Syarat Mutlak: a harus koprima (GCD(a, 26) = 1) agar invers perkalian modular a{-1} ada. Ini menjamin dekripsi dapat dibalik.
- **Rumus matematis :**

  Enkripsi: $C = (aP + b) \mod 26$

  Dekripsi: $P = a^{-1}(C - b) \mod 26$

- **Implementasi kode :**

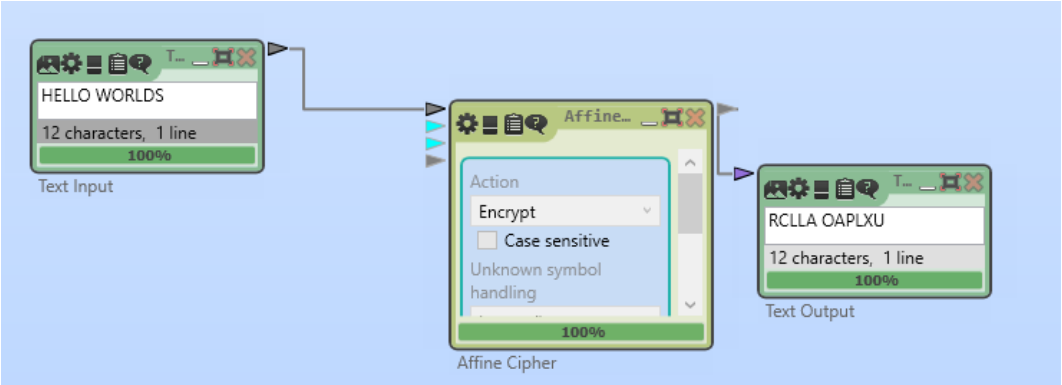



```
plain_text = input("Masukkan teks: ")
a = int(input("Masukkan kunci a (coprime dengan 26): "))
b = int(input("Masukkan kunci b: "))

encrypted = affine_encrypt(plain_text, a, b)
print("Hasil enkripsi:", encrypted)

decrypted = affine_decrypt(encrypted, a, b)
print("Hasil dekripsi:", decrypted)
```

- **Hasil Output :**

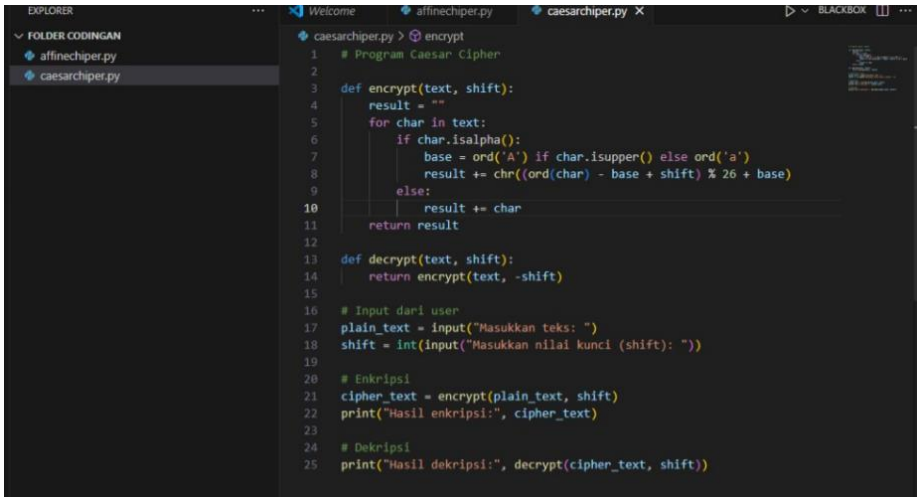
```
Masukkan teks: HELLO WORDS
Masukkan kunci a (coprime dengan 26): 5
Masukkan kunci b: 8
Masukkan kunci a (coprime dengan 26): 5
Masukkan kunci b: 8
Masukkan kunci b: 8
Hasil enkripsi: RCLLA OAPXU
Hasil enkripsi: RCLLA OAPXU
Hasil dekripsi: HELLO WORDS
```

Verifikasi cryptool:



2. Caesar Cipher

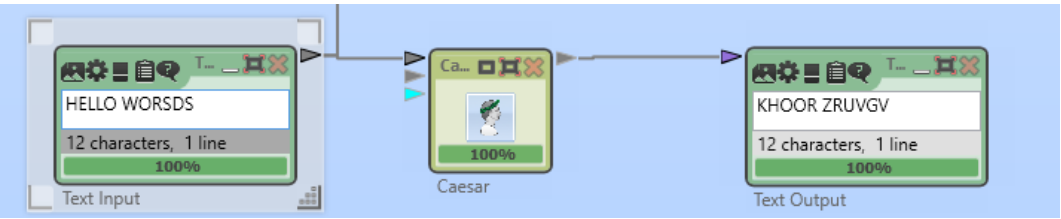
- **Mekanisme :** Sandi ini adalah kasus khusus dari substitusi monoalfabetik. Setiap huruf digeser sejauh n posisi.
- **Fungsi Kunci** : shift (geseran tunggal, 1 < shift > 25).
- **Rumus matematis :**
- $$\text{Enkripsi: } C = (P + k) \mod 26$$
- $$\text{Dekripsi: } P = (C - k) \mod 26$$
- **Implentasi kode :**



- **Hasil Output :**

```
Masukkan teks: HELLO, WORDS
Masukkan nilai kunci (shift): 3
Hasil enkripsi: KHOOR, ZRUGV
Hasil enkripsi: KHOOR, ZRUGV
Hasil dekripsi: HELLO, WORDS
```

Cryptool:



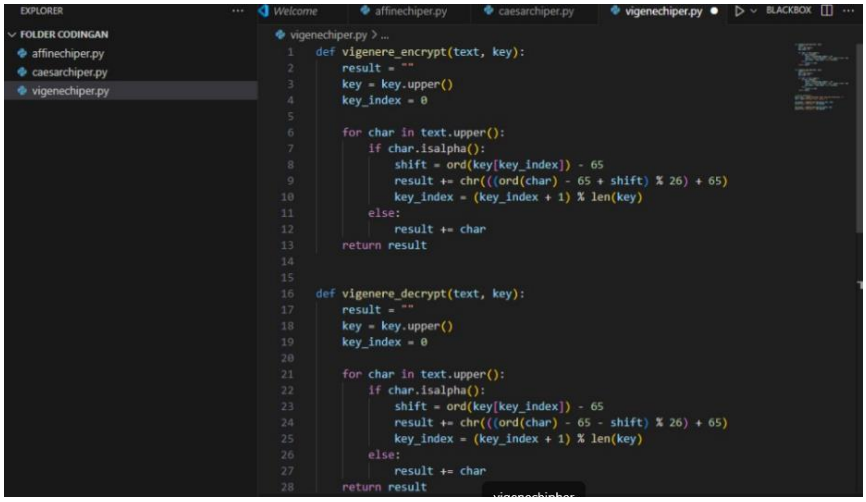
3. Vigenere Chiper (sandi kunci berulang)

- **Mekanisme :** Sandi ini menggunakan *banyak alfabet* pengganti yang berbeda, ditentukan oleh huruf pada kata kunci (K) yang diulang-ulang (perluasan kunci). Ini berhasil menyembunyikan pola frekuensi huruf secara langsung.
- **Fungsi Kunci** : Kata kunci K = k1, k2, …, kL.
- **Rumus matematis :**

  Enkripsi: $C_i = (P_i + K_i) \mod 26$

  Dekripsi: $P_i = (C_i - K_i) \mod 26$

- **Implementasi kode :**

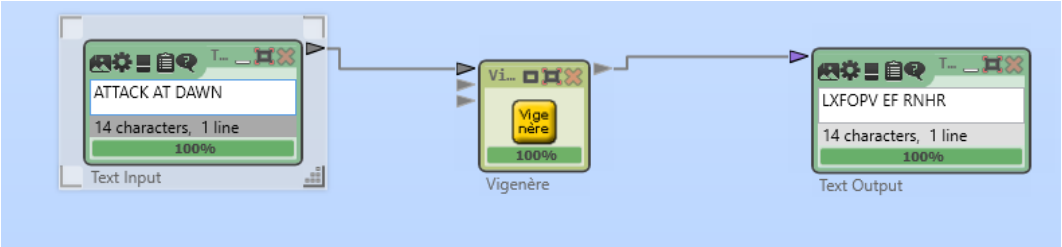


```
    return result


# Contoh penggunaan
plain_text = input("Masukkan teks yang mau dienkripsi: ")
key = input("Masukkan kunci (kata): ")

encrypted = vigenere_encrypt(plain_text, key)
print("Hasil enkripsi:", encrypted)

decrypted = vigenere_decrypt(encrypted, key)
print("Hasil dekripsi:", decrypted)
```

- **Hasil Output :**

```
Masukkan teks yang mau dienkripsi: ATTACK AT DAWN
Masukkan kunci (kata): LEMON
Masukkan kunci (kata): LEMON
Hasil enkripsi: LXFOPV EF RNHR
Hasil dekripsi: ATTACK AT DAWN
```

Cryptool:



4. Playfair Cipher

- **Mekanisme:** Playfair Cipher merupakan **cipher digraph substitusi**, artinya mengenkripsi **dua huruf sekaligus (pasangan huruf)** menggunakan **matriks kunci berukuran 5×5**.

Langkah-langkah utama:

- Membentuk tabel kunci (key table) berukuran 5×5 dari kata kunci (key).
  - Huruf "J" digabung dengan "I" (jadi hanya 25 huruf).
  - Huruf yang berulang diabaikan.
- Membagi plaintext menjadi pasangan dua huruf (bigram).
  - Jika ada huruf ganda (seperti "LL"), sisipkan huruf X di antara mereka.
  - Jika jumlah huruf ganjil, tambahkan X di akhir.
- Mengenkripsi setiap pasangan berdasarkan posisi kedua huruf di tabel.
- **Aturan Enkripsi:**

Kondisi Huruf	Aturan Enkripsi	Penjelasan
Sebaris	Geser ke kanan satu kolom	Jika di ujung kanan, kembali ke awal baris
Sekolom	Geser ke bawah satu baris	Jika di bawah, kembali ke atas kolom
Membentuk persegi	Tukar kolom antar huruf	Bentuk persegi imajiner, ambil huruf di sudut lainnya

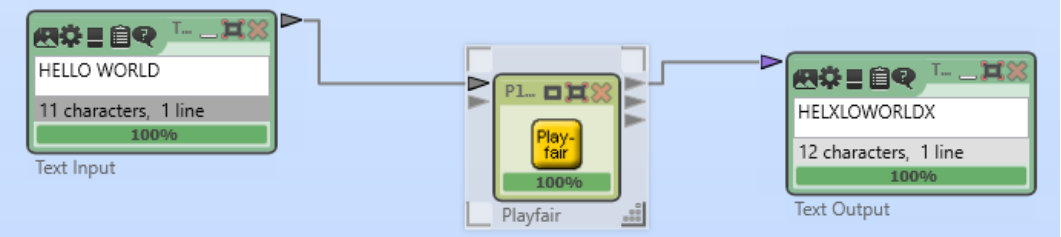
- **Implementasi kode:**

```
def generate_table(key):
    alphabet = 'ABCDEFGHIKLMNOPQRSTUVWXYZ'
    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)
```

- Hasil Output:

```
PS C:\Users\usER\OneDrive\Pictures\Documents\tugaskripto> & C:/Users/usER/Downloads/python.exe
/tugaskripto/playfaircipher.py
Tabel Kunci:
KEYWO
RDABC
FGHIL
MNPQS
TUVXZ

Plaintext : HELLO WORLD
Ciphertext: GYIZSCOKCFBU
Dekripsi : HELXLOWORLDX
```

Cryptool:



5. Hill Cipher

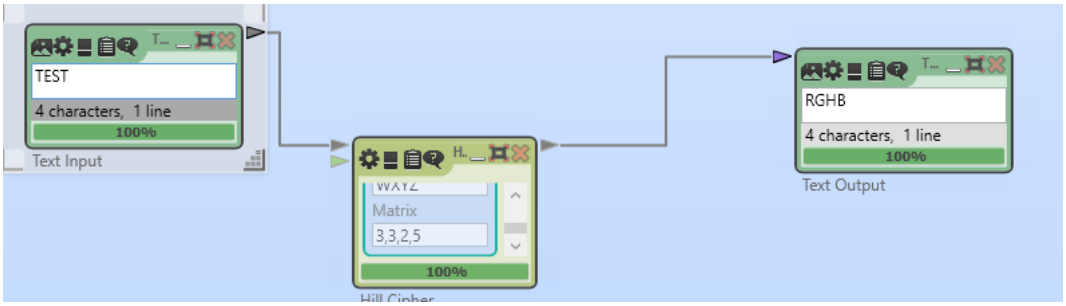
- **Mekanisme**: Hill Cipher adalah algoritma **cipher polialfabetik blok**, yang mengenkripsi plaintext dalam bentuk **kelompok huruf (blok)** menggunakan **operasi matriks linear** di bawah **modulo 26**.
- Kunci pada Hill Cipher berbentuk **matriks bujur sangkar (n × n)** yang berisi bilangan bulat (biasanya hasil konversi dari huruf A–Z → 0–25).
- Fungsi key: Kunci (key) direpresentasikan dalam bentuk **matriks persegi**, contohnya untuk kunci 2×2:
- **Rumus matematis:**
  - Enkripsi: $C = (K \cdot P) \mod 26$
  - Dekripsi: $P = (K^{-1} \cdot C) \mod 26$
- **Implementasi kode:**

```python
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]))
```

- Hasil Output:

```
PS C:\Users\usER\OneDrive\Pictures\Documents\tugaskripto> & C:/Users/usER/Downloads/python.e
/tugaskripto/hillcipher.py
RGHB
```

Cryptool:



II. Analisis kuantitatif dan keamanan

Tingkat keamanan sandi klasik berbanding lurus dengan ruang kunci (keyspace) dan kerentenannya terhadap metode serangan statistika

Sandi	**Ruang Kunci (K)**	**Dasar Kerentanan**	**Keamanan Relatif**
Caesar	25	Jumlah alfabet & K kecil	Sangat Rendah (Pecah Instan)
Affine	312	K kecil & mudah ditebak	Rendah (Cepat Pecah)
Vigenère	Bergantung panjang kunci (misal 26^L)	Kunci pendek & pengulangan	Sedang dengan Kriptanalisis Lanjutan
Playfair	25! (faktor dari 25 huruf unik)	Digram berulang & analisis frekuensi	Sedang (Sulit dipecahkan tanpa konteks)
Hill	Bergantung ukuran matriks (misal 2×2: 26^4)	Matriks dapat dibalik & rentan terhadap analisis linear	Sedang–Tinggi (Tergantung ukuran matriks)

III. Analisis Kelemahan

Cipher	**Kelemahan Utama**
Caesar	Mudah dipecahkan dengan brute-force (hanya 25 kemungkinan).
Vigenère	Masih rentan terhadap analisis frekuensi jika panjang kunci diketahui.

Cipher	Kelemahan Utama
Affine	Jika nilai a diketahui atau salah pilih, cipher dapat terpecahkan.
Playfair	Tidak cocok untuk komputerisasi modern karena hanya bekerja pada huruf.
Hill	Membutuhkan matriks invers untuk dekripsi; sulit jika determinan tidak relatif prima terhadap 26.

IV. Kesimpulan

Lima algoritma cipher klasik memiliki prinsip enkripsi yang berbeda namun bertujuan sama, yaitu menjaga kerahasiaan pesan.

- Caesar: sederhana, cepat, mudah dipecahkan.
- Vigenère: lebih aman, polialfabetik, tapi rentan Kasiski.
- Affine: dua kunci, masih bisa frekuensi.
- Playfair: digraph, lebih aman, analisis digraph tetap memungkinkan.
- Hill: matriks blok, paling aman diantara yang lain, tapi perhitungan lebih rumit.

Link github:

https://github.com/marshelindo/TugasKriptografi.git

https://github.com/khabibah/tugaskriptografi