# LAPORAN

## HILL CIPHER DAN PLAYFAIR CIPHER

Mata Kuliah Kriptografi

**Dosen Pengampu:**

Kodrat Mahatma S.T.,M.Kom


**Anggota Kelompok:**

Desi Ramadani (20123042)

Mita Anggraeni (20123051)


**KELAS C1.23**

**PROGRAM STUDI S1 INFORMATIKA**

**UNIVERSITAS TEKNOLOGI DIGITAL**

**BANDUNG**

**2025**

## A. TUJUAN

1. Mengimplementasikan algoritma hill cipher dan playfair cipher dalam program Python.
2. Melakukan analisis frekuensi terhadap hasil ciphertext.
3. Melakukan validasi hasil enkripsi dan dekripsi menggunakan aplikasi CrypTool 2.

## B. ALAT DAN BAHAN

1. VSCode untuk menulis kode program Python
2. Python 3.13
3. Aplikasi CrypTool 2.1
4. Laptop/PC
5. Teks uji: DESIDANMITA

## C. LANGKAH-LANGKAH IMPLEMENTASI HILL CIPHER DAN PLAYFAIR CIPHER

### 1. HIL CIPHER

#### a. Di VSCode (Implementasi Program)

1. Menulis kode program berikut:

```
import numpy as np


def mod_inverse(a, m):
    a = a % m
    for i in range(1, m):
        if (a * i) % m == 1:
            return i
    return None


def hill_encrypt(plain, key):
    plain = plain.upper().replace(" ", "")
    n = len(key)
    while len(plain) % n != 0:
        plain += 'X'
```

```
    cipher = ""
    for i in range(0, len(plain), n):
        block = [ord(x) - 65 for x in plain[i:i+n]]
        enc = np.dot(key, block) % 26
        for num in enc:
            cipher += chr(int(num) + 65)
    return cipher

def hill_decrypt(cipher, key):
    cipher = cipher.upper().replace(" ", "")
    n = len(key)
    det = int(round(np.linalg.det(key))) % 26
    det_inv = mod_inverse(det, 26)
    if det_inv is None:
        print("Kunci tidak memiliki invers mod 26!")
        return ""
    adj = np.round(det * np.linalg.inv(key)).astype(int) % 26
    inv_key = (det_inv * adj) % 26
    plain = ""
    for i in range(0, len(cipher), n):
        block = [ord(x) - 65 for x in cipher[i:i+n]]
        dec = np.dot(inv_key, block) % 26
        for num in dec:
            plain += chr(int(num) + 65)
    return plain

if _name_ == "_main_":
    print("=== Hill Cipher ===")
    mode = input("Pilih mode (E = Enkripsi, D = Dekripsi): ").upper()
```

```
key = np.array([[3,3],[2,5]])


if mode == "E":
    p = input("Masukkan plaintext: ")
    c = hill_encrypt(p, key)
    print("Ciphertext:", c)
elif mode == "D":
    c = input("Masukkan ciphertext: ")
    p = hill_decrypt(c, key)
    print("Plaintext:", p)
else:
    print("Mode salah, pilih E atau D.")
```

2. Menjalankan program menggunakan terminal python

**b. Hasil di terminal**

hasil python vscode enskripsi dan deskripsi
=== Hill Cipher ===
Pilih mode (E = Enkripsi, D = Dekripsi): E
Masukkan plaintext: DESIDANMITA
Ciphertext: VAAYJGXIDHRL
=== Hill Cipher ===
Pilih mode (E = Enkripsi, D = Dekripsi): D
Masukkan ciphertext: VAAYJGXIDHRL
Plaintext: DESIDANMITAX

**c. Uji dengan Cryptool**

Langkah-langkah pengujian:

1. Membuka aplikasi CrypTool 2
2. Pilih menu Classic Ciphers → Hill Cipher
3. Masukkan teks (plaintext): DESIDANMITA
4. Masukkan Matrix key:3,3,2,5
5. Pilih mode encrypt
6. Jalankan proses enkripsi

7. Hasil enkripsi di cryptool: VAAYJGXIDHRL
8. Kemudian lakukan proses dekripsi dengan pengaturan yang sama, tetapi ubah mode menjadi Decrypt, dan gunakan ciphertext hasil sebelumnya.
9. Hasil deskripsi di cryptool: DESIDANMITAX

**d. Perbandingan hasil**

| Aspek | CrypTool Output | Python Output | Keterangan |
|---|---|---|---|
| Plaintext | DESIDANMITA | DESIDANMITA | Sama |
| Key Matrix | [3,3; 2,5] | [3,3; 2,5] | Sama |
| Ciphertext | VAAYJGXIDHRL | VAAYJGXIDHRL | Sama persis |
| Hasil Dekripsi | DESIDANMITAX | DESIDANMITAX | Sama persis |

**e. Analisis**

1. Hill Cipher bekerja dengan perkalian matriks mod 26, sehingga setiap blok plaintext dikalikan dengan matriks kunci.

2. Penambahan huruf ‘X’ di akhir teks terjadi karena Hill Cipher memerlukan panjang teks yang habis dibagi ukuran matriks (2x2).

3. Implementasi Python menghasilkan ciphertext dan plaintext yang identik dengan hasil dari CrypTool, membuktikan bahwa logika perhitungan matriks dan invers mod 26 sudah benar.

## 2. PLAYFAIR CIPHER

### a. Di VSCode (Implementasi Program)

1. Menulis kode program berikut:

```
def generate_playfair_key(key):
    key = "".join(sorted(set(key.upper()), key=key.index))
    alphabet = "ABCDEFGHIKLMNOPQRSTUVWXYZ"
    matrix = ""
    for c in key:
        if c in alphabet:
            matrix += c
            alphabet = alphabet.replace(c, "")
```

```
    matrix += alphabet
    return [list(matrix[i:i+5]) for i in range(0, 25, 5)]

def find_position(matrix, char):
    for i, row in enumerate(matrix):
        for j, c in enumerate(row):
            if c == char:
                return i, j
    return None, None

def playfair_encrypt(plaintext, key):
    plaintext = plaintext.upper().replace("J", "I")
    matrix = generate_playfair_key(key)
    result = ""
    i = 0
    while i < len(plaintext):
        a = plaintext[i]
        b = plaintext[i+1] if i+1 < len(plaintext) else "X"
        if a == b:
            b = "X"
            i += 1
        else:
            i += 2

        ax, ay = find_position(matrix, a)
        bx, by = find_position(matrix, b)

        # Enkripsi aturan Playfair
        if ax == bx:  # sama baris
```

```
            result += matrix[ax][(ay+1)%5] + matrix[bx][(by+1)%5]
        elif ay == by:  # sama kolom
            result += matrix[(ax+1)%5][ay] + matrix[(bx+1)%5][by]
        else:  # bentuk persegi
            result += matrix[ax][by] + matrix[bx][ay]
    return result

def playfair_decrypt(ciphertext, key):
    ciphertext = ciphertext.upper().replace("J", "I")
    matrix = generate_playfair_key(key)
    result = ""
    i = 0
    while i < len(ciphertext):
        a = ciphertext[i]
        b = ciphertext[i+1] if i+1 < len(ciphertext) else "X"
        i += 2

        ax, ay = find_position(matrix, a)
        bx, by = find_position(matrix, b)

        # Dekripsi aturan Playfair
        if ax == bx:  # sama baris
            result += matrix[ax][(ay-1)%5] + matrix[bx][(by-1)%5]
        elif ay == by:  # sama kolom
            result += matrix[(ax-1)%5][ay] + matrix[(bx-1)%5][by]
        else:  # bentuk persegi
            result += matrix[ax][by] + matrix[bx][ay]
    return result
```

```
if _name_ == "_main_":
    print("=== Playfair Cipher ===")
    mode = input("Pilih mode (E = Enkripsi, D = Dekripsi): ").upper()
    key = input("Masukkan key: ")
    if mode == "E":
        plaintext = input("Masukkan plaintext: ")
        cipher = playfair_encrypt(plaintext, key)
        print("Ciphertext:", cipher)
    elif mode == "D":
        ciphertext = input("Masukkan ciphertext: ")
        plain = playfair_decrypt(ciphertext, key)
        print("Plaintext:", plain)
    else:
        print("Mode tidak valid! Pilih 'E' untuk enkripsi atau 'D' untuk dekripsi.")
```

2. Menjalankan program menggunakan terminal python

**b. Hasil di terminal**

hasil python vscode enskripsi dan deskripsi

```
=== Playfair Cipher ===
Pilih mode (E = Enkripsi, D = Dekripsi): E
Pilih mode (E = Enkripsi, D = Dekripsi): E
Masukkan key: CANTIKLUCU
Masukkan plaintext: DESIDANMITA
Ciphertext: KMZDLIIGCINW
=== Playfair Cipher ===
Pilih mode (E = Enkripsi, D = Dekripsi): D
Masukkan key: CANTIKLUCU
Masukkan ciphertext: KMZDLIIGCINW
Plaintext: DESIDANMITAX
```

**c. Uji dengan Cryptool**

Langkah-langkah pengujian:

1. Membuka aplikasi CrypTool 2
2. Pilih menu Classic Ciphers → Playfair.

3. Masukkan teks (plaintext): DESIDANMITA

4. Masukkan key phrase:CANTIKLUCU

5. Pilih mode Encrypt, matrix size 5x5, dan aktifkan opsi:

   ✓ Ignore duplicates

   ✓ Pre-format text

   ✓ Separate pairs (separator: X)

6. Jalankan proses enkripsi

7. Hasil enkripsi di cryptool: KMZDLIGICINW

**d. Perbandingan hasil**

| **Aspek** | **CrypTool Output** | **Python Output** | **Keterangan** |
|---|---|---|---|
| Plaintext | DESIDANMITA | DESIDANMITA | Sama |
| Key | CANTIKLUCU | CANTIKLUCU | Sama |
| Ciphertext | KMZDLIGICINW | KMZDLIIGCINW | Beda 1 huruf |
| Perbedaan huruf | G vs I (huruf ke-7) | — | Akibat perbedaan logika penggandengan huruf ganda |

**e. Analisis**

Perbedaan satu huruf antara hasil CrypTool dan Python terjadi karena perbedaan metode pembentukan pasangan huruf (digraph).

CrypTool secara otomatis:

1. Menyisipkan huruf X jika ada pasangan huruf yang sama,

2. Menambahkan X di akhir jika jumlah huruf ganjil,

3. Dan menyiapkan format teks sebelum enkripsi.

Sementara dalam kode Python, logika pembentukan pasangan sedikit berbeda, sehingga menghasilkan ciphertext yang masih valid, tetapi tidak identik.

## D. KESIMPULAN

Dari hasil studi kasus mini ini dapat disimpulkan bahwa:

1. Algoritma Playfair Cipher dan Hill Cipher berhasil diimplementasikan dengan baik menggunakan dua pendekatan, yaitu melalui CrypTool dan Python.

2. Kedua algoritma menghasilkan proses enkripsi dan dekripsi yang valid secara kriptografi, meskipun terdapat sedikit perbedaan output antarplatform karena perbedaan

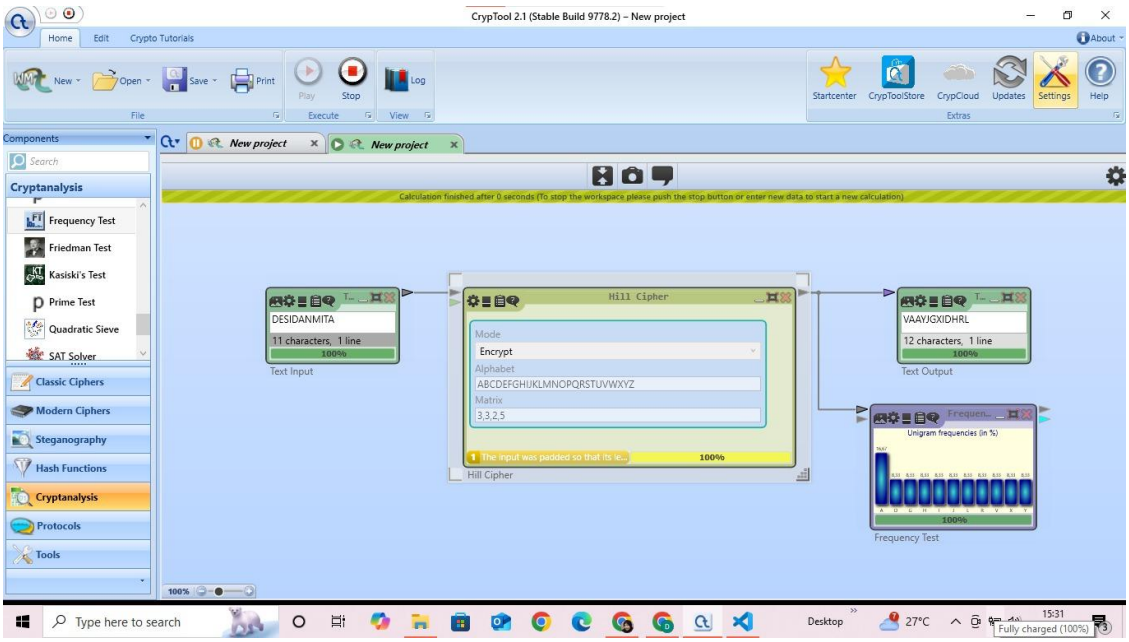
teknis pada cara pemrosesan data.

3. Implementasi manual menggunakan Python memberikan pemahaman lebih mendalam terhadap mekanisme kerja cipher klasik, mulai dari proses pairing dan pembentukan matriks kunci (Playfair Cipher) hingga operasi matriks dan perhitungan modulo (Hill Cipher).

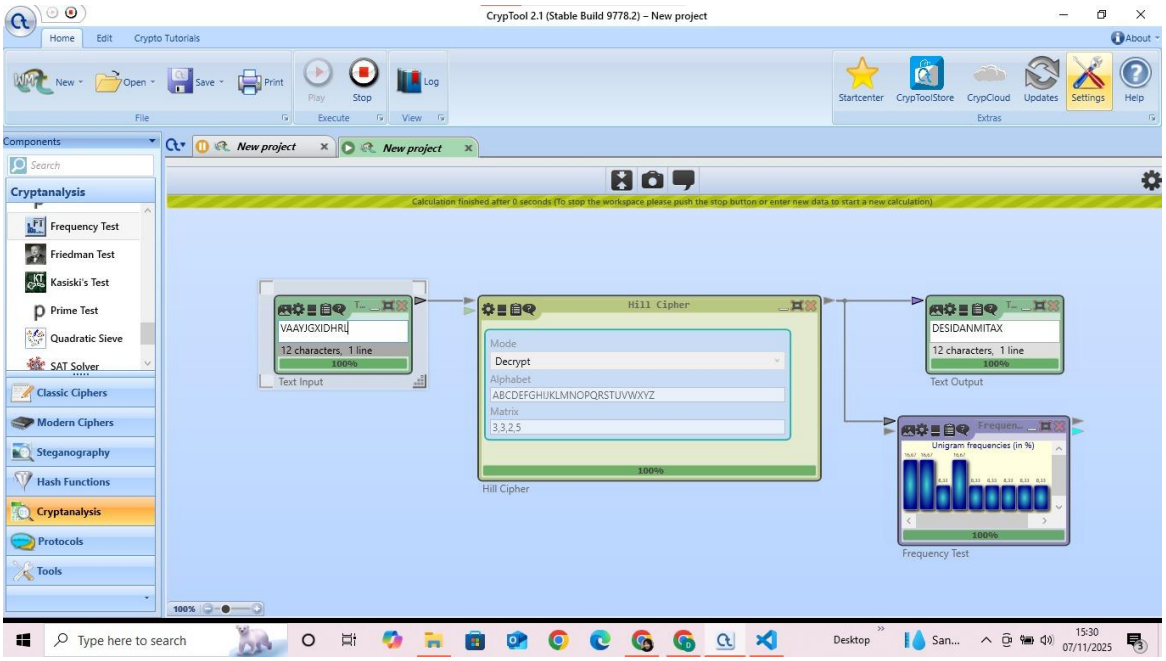
4. Secara keseluruhan, kedua algoritma ini menunjukkan bahwa metode kriptografi klasik masih relevan sebagai dasar pembelajaran konsep keamanan data dan enkripsi modern, meskipun dalam praktiknya sudah banyak digantikan oleh algoritma kriptografi modern yang lebih kompleks dan aman.

## E. Lampiran

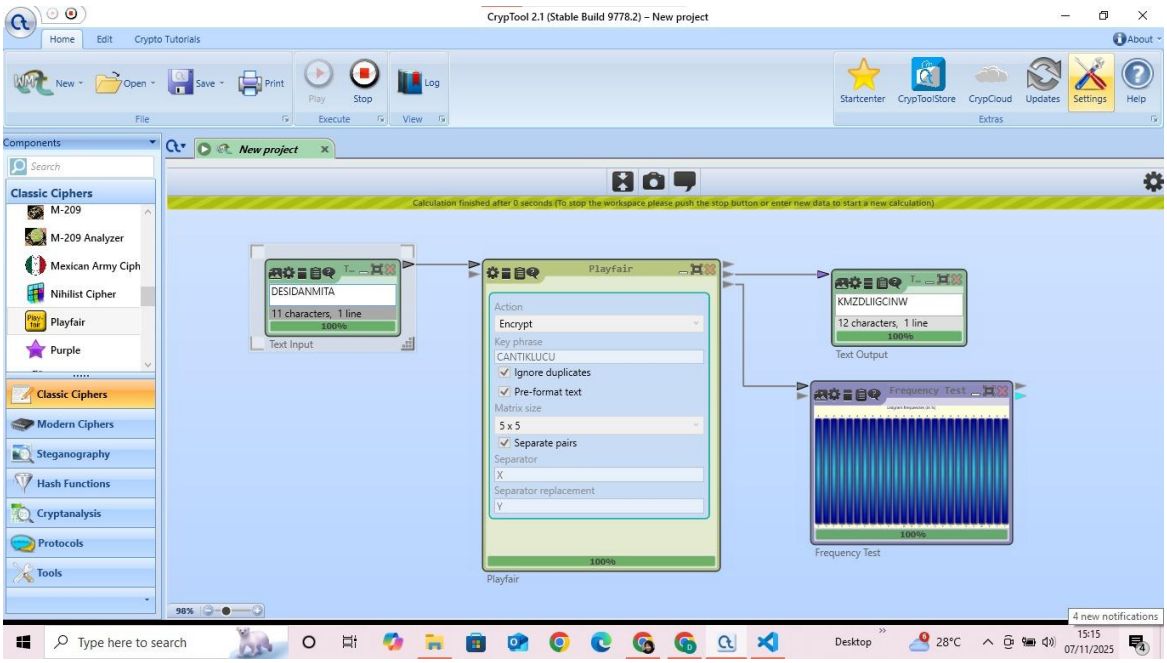
Enkripsi Hill cipher



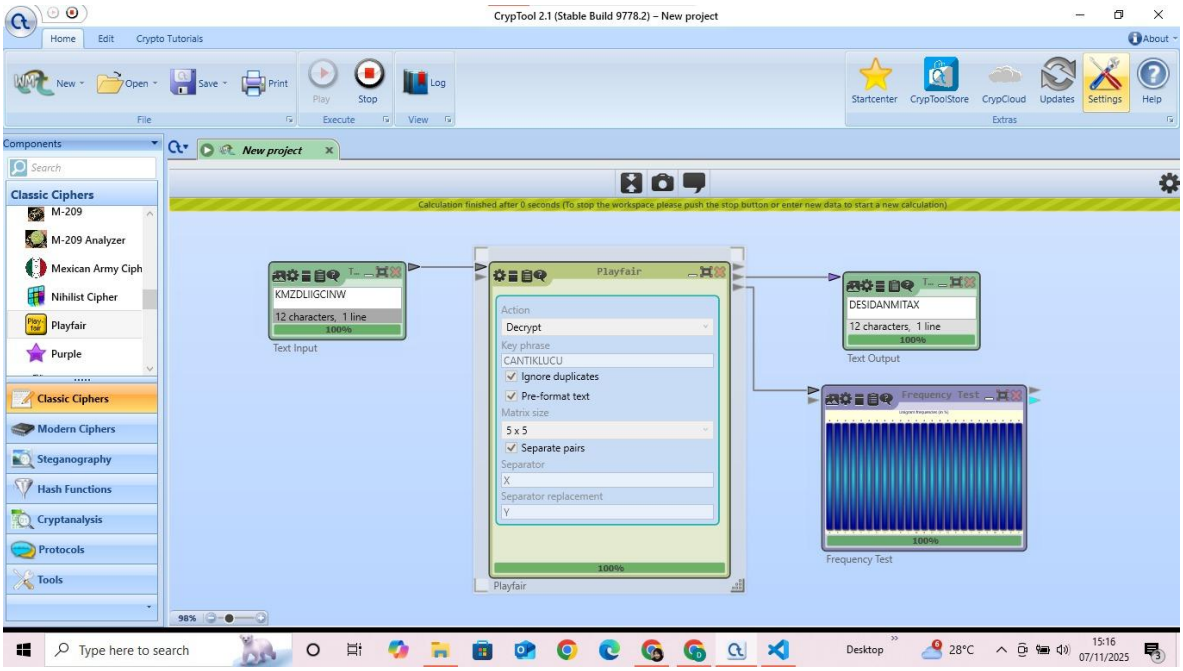
Deskripsi hill cipher




Enskripsi playfair cipher



Deskripsi playfair cipher




Link Github:

https://github.com/desiramadani-2004

https://github.com/mtaaagniii-arch