LAPORAN PRAKTIKUM STUDI KASUS TUGAS MINI

Nama :

- **Mayasari Agustina (20123060)**
- **Syifa Sopian Nurdiansyah (20123049)**

Kelas : C1.23 S1 Informatika

Matkul: Kriptografi

Studi Kasus Mini

Gunakan teks 'HELLO WORLD' untuk uji Hill + Playfair. Boleh juga kata yang lain. Kalau bisa interaktif, sehingga teks bisa diubah. Bandingkan dengan hasil dari CrypTool atau CyberChef.

Praktikum Mandiri

1. Jalankan Hill Cipher di CrypTool.
2. Bandingkan hasil dengan hasil di Python.
3. Catat perbedaan jika ada.

A. Tujuan Praktikum

Tujuan dari praktikum ini adalah untuk memahami dan mengimplementasikan algoritma Hill Cipher serta Playfair Cipher menggunakan bahasa pemrograman Python, kemudian membandingkan hasilnya dengan implementasi dari CrypTool atau CyberChef. Dengan percobaan ini, diharapkan mahasiswa dapat memahami cara kerja enkripsi dan dekripsi klasik berbasis substitusi dan transformasi matriks.

B. Dasar Teori

1). Hill Cipher

Kode program Hill Cipher menggunakan operasi matriks untuk enkripsi dan dekripsi dengan memanfaatkan library NumPy. Fungsi hill_encrypt melakukan perkalian matriks key dengan pasangan huruf plaintext kemudian dilakukan operasi modulo 26. Sedangkan

hill_decrypt menghitung determinan, mencari invers matriks modulo 26, lalu melakukan dekripsi.

Rumus umum:

$C = (K \times P) \bmod 26$
$P = (K^{-1} \times C) \bmod 26$

2). Playfair Cipher

Program Playfair Cipher menghasilkan tabel 5x5 berdasarkan kunci, kemudian mengenkripsi pasangan huruf berdasarkan posisi mereka di tabel. Huruf ganda ditangani dengan penyisipan huruf 'X'.

C. Alat dan Bahan

- Bahasa Pemrograman: Python 3
- Tools: CrypTool

D. Implementasi Program

1). Program Playfair Cipher (Python)

Kode:

```python
def playfair_encrypt(text, key):
    matrix = generate_playfair_matrix(key)
    pairs = playfair_prepare("".join([c for c in text.upper() if c.isalpha()]))
    result = ""
    for a, b in pairs:
        row1, col1 = find_position(matrix, a)
        row2, col2 = find_position(matrix, b)
        if row1 == row2:
            result += matrix[row1][(col1 + 1) % 5] + matrix[row2][(col2 + 1) % 5]
        elif col1 == col2:
            result += matrix[(row1 + 1) % 5][col1] + matrix[(row2 + 1) % 5][col2]
        else:
            result += matrix[row1][col2] + matrix[row2][col1]
    return result

def playfair_decrypt(text, key):
    matrix = generate_playfair_matrix(key)
    pairs = playfair_prepare(text)
    result = ""
    for a, b in pairs:
        row1, col1 = find_position(matrix, a)
        row2, col2 = find_position(matrix, b)
        if row1 == row2:
            result += matrix[row1][(col1 - 1) % 5] + matrix[row2][(col2 - 1) % 5]
        elif col1 == col2:
            result += matrix[(row1 - 1) % 5][col1] + matrix[(row2 - 1) % 5][col2]
        else:
            result += matrix[row1][col2] + matrix[row2][col1]
    return result
```

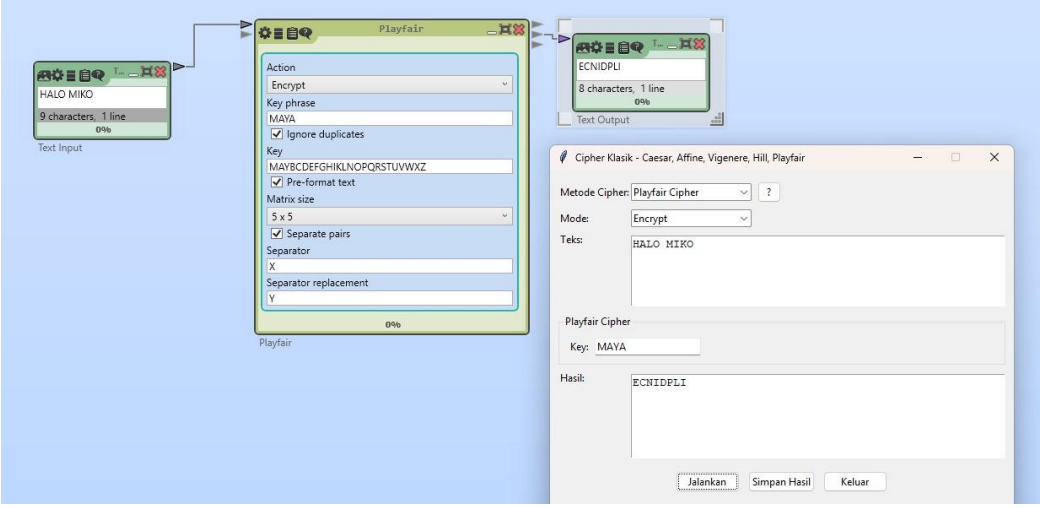
```python
# ==========Playfair
def generate_playfair_matrix(key):
    key = "".join(dict.fromkeys(key.upper().replace("J", "I")))
    alphabet = "ABCDEFGHIKLMNOPQRSTUVWXYZ"
    matrix = []
    for char in key + alphabet:
        if char not in matrix:
            matrix.append(char)
    return [matrix[i:i+5] for i in range(0, 25, 5)]

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

def playfair_prepare(text):
    text = text.upper().replace("J", "I")
    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
    return pairs
```

- Fungsi generate_playfair_matrix digunakan untuk membuat matriks 5x5 yang dibutuhkan dalam metode playfair cipher ini.
- Sementara Fungsi find_position digunakan untuk mencari posisi (baris dan kolom) suatu huruf dalam tabel.
- Playfair_prepare merupakan Fungsi untuk mempersiapkan teks sebelum di enkripsi. Teks akan dipecah menjadi 2 huruf , jika ada 2 huruf sama maka huruf akhirnya akan digantikan dengan huruf X. Begitu juga apabila hurufnya berjumlah ganjil
- Playfair_encrypt dan playfair_decrypt merupakan Fungsi utama dalam enkripsi dan dekripsi playfair dengan dua parameter untuk plaintext dan key/kunci.
- “matrix” dan “pairs” bertugas untuk memanggil fungsi yang membuat matriks, dan juga memanggil fungsi untuk pasangan huruf.
- “if row1 == row2” kondisi ini apabila sama baris, maka geser ke kanan satu kolom.
- “elif col1 == col2” kondisi ini apabila sama kolom, Maka geser kebawah satu kolom.

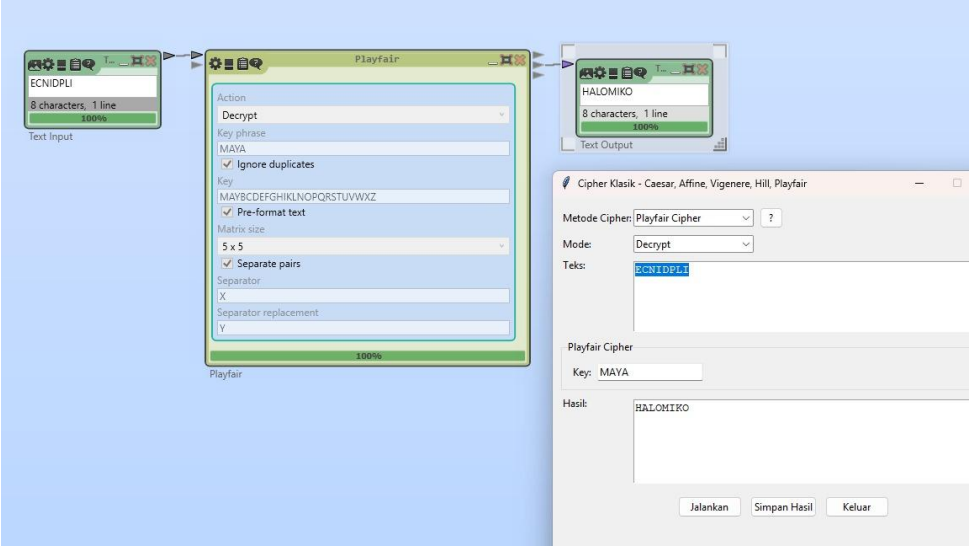
- Sementara untuk “else” apabila beda baris dan beda kolom, maka tukar kolom antar baris.
- Untuk fungsi dekripnya sama, namun hanya arah pergeserannya saja yang dibalik.

Cryptool 2:

Playfair Cipher Encrypt



Playfair Decrypt:



Pada proses enkripsi, keduanya berhasil meng-enkripsi teks “HALO MIKO” dengan kata kunci “MAYA” menjadi “ECNIDPLI”.

Pada proses dekripsi, keduanya juga berhasil mengembalikkan teks menjadi seperti pada awalnya. Hasilnya juga cukup baik, namun kelemahan metode ini adalah pada huruf ganda atau huruf yang tidak berpasangan. Serta masih bisa dipecahkan dengan frekuensi analisis meskipun sebenarnya cukup rumit juga, menilik ke hasil, khususnya dalam proses dekripsi cukup normal dan sempurna karena teks objek tidak memiliki huruf ganda, dan juga jumlah hurufnya berjumlah genap

2). Program Hill Cipher (Python)

Kode:

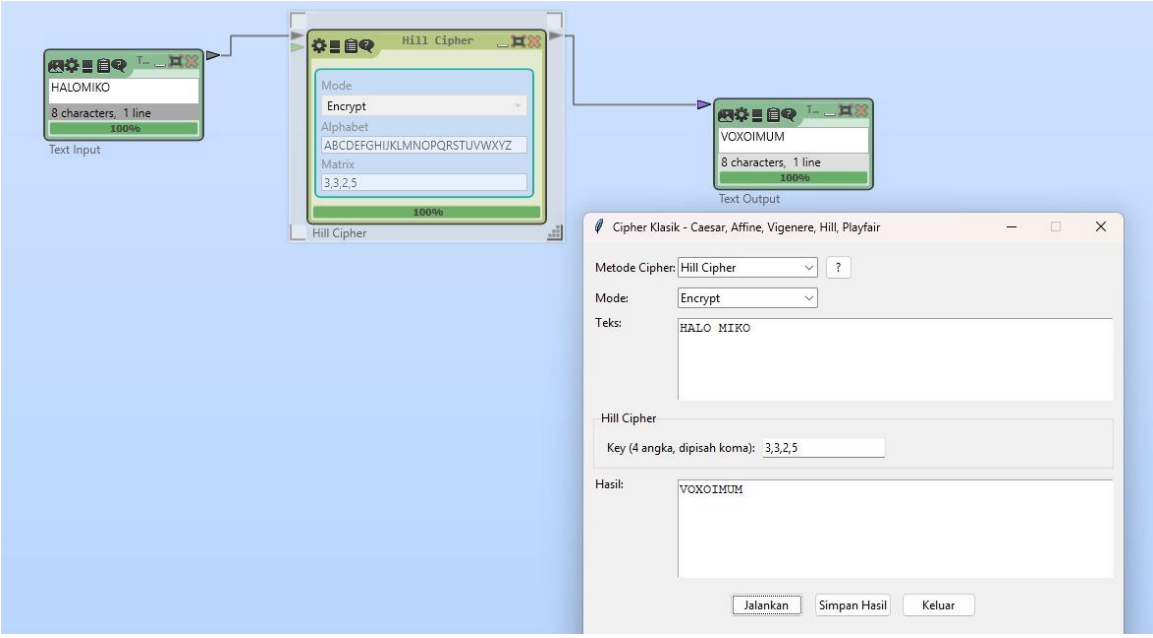
```python
# ==========Hill
def text_to_numbers(text):
    return [ord(c) - 65 for c in text.upper() if c.isalpha()]

def numbers_to_text(nums):
    return "".join(chr((n % 26) + 65) for n in nums)
```

```python
def hill_encrypt(text, key_matrix):
    text = text.upper().replace(" ", "")
    if len(text) % 2 != 0:
        text += "X"
    result = ""
    for i in range(0, len(text), 2):
        pair = np.array([[ord(text[i]) - 65], [ord(text[i+1]) - 65]])
        encrypted = np.dot(key_matrix, pair) % 26
        result += chr(encrypted[0][0] + 65) + chr(encrypted[1][0] + 65)
    return result

def hill_decrypt(text, key_matrix):
    det = int(np.round(np.linalg.det(key_matrix))) % 26
    det_inv = None
    for i in range(26):
        if (det * i) % 26 == 1:
            det_inv = i
            break
    if det_inv is None:
        return "Error: Determinan tidak memiliki invers modulo 26!"

    adj = np.round(det * np.linalg.inv(key_matrix)).astype(int) % 26
    inv_matrix = (det_inv * adj) % 26

    result = ""
    for i in range(0, len(text), 2):
        pair = np.array([[ord(text[i]) - 65], [ord(text[i+1]) - 65]])
        decrypted = np.dot(inv_matrix, pair) % 26
        result += chr(int(decrypted[0][0]) + 65) + chr(int(decrypted[1][0]) + 65)
    return result
```

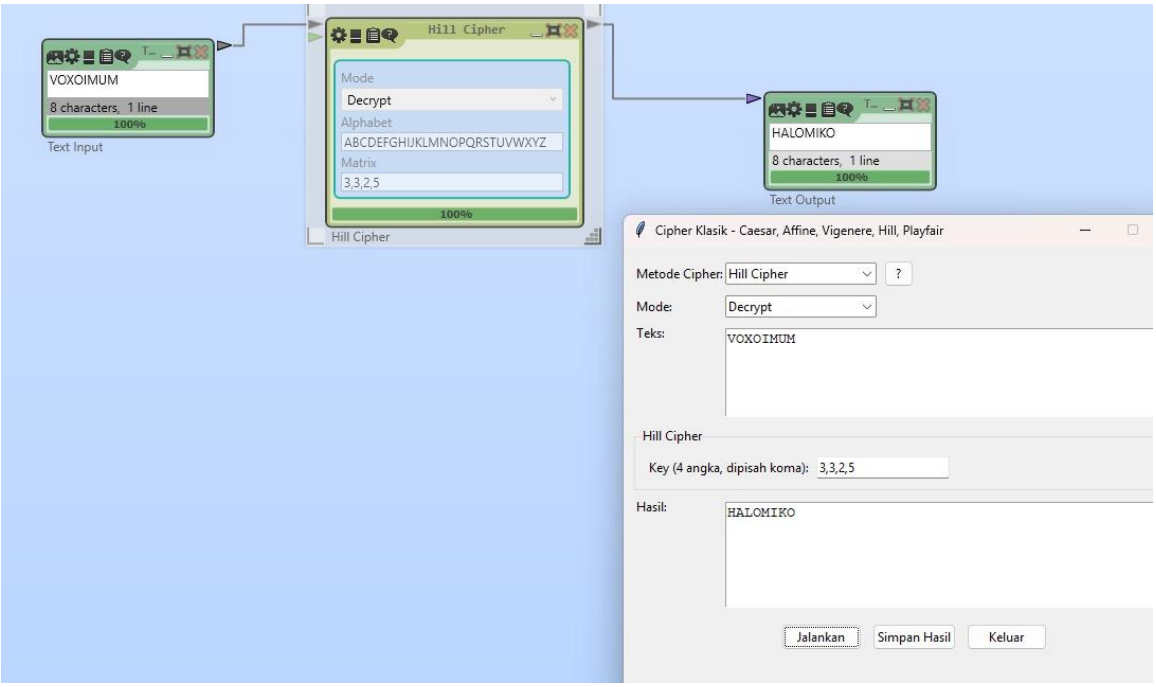
- Hill cipher menggunakan metode yang sama dengan playfair, yaitu membagi kata menjadi dua huruf, bila ada huruf yang sama atau pasangan bernilai ganjil/satu, maka akan diganti atau ditambahkan huruf X.
- “encrypted = np.dot(key_matrix, pair) % 26” namun dalam bagian ini, huruf huruf pasangan dikalikan dengan key matrix dan ambil hasilnya mod 26.
- Metode dekripnya sedikit rumit. untuk dekripsi, diperlukan **matriks invers modulo 26** dari kunci.

Cryptool:

Hill Cipher Encrypt



Hill Decrypt



Dalam proses enkripsi Hill cipher, kedua program berhasil meng-enkripsi kata “HALO MIKO” dengan kunci matriks 2x2 “3,3,2,5” menjadi “VOXOIMUM”.

Sementara untuk proses dekripsinya juga, metode ini berhasil mengembalikkan *ciphertext* menjadi seperti di awal. Dan permasalahannnya sama dengan metode playfair, yaitu ketentuan huruf ganda, dan salah satu kelemahannya juga adalah memerlukan invers matriks modulo 26. Sehingga pemilihan kunci-nya harus berhati-hati. Serta rentan terhadap *plaintext* yang diketahui atau *known plaintext attack*.

E. Hasil Pengujian

Algoritma	**Plaintext**	**Ciphertext (Python)**	**Hasil Dekripsi**	**Hasil CrypTool**	**Catatan**
Hill Cipher	HALO MIKO	VOXOIMUM	HALOMIKO	VOXOIMUM	Sama
Playfair Cipher	HALO MIKO	ECNIDPLI	HALOMIKO	ECNIDPLI	Sama

F. Analisis Perbandingan

Hasil pengujian menunjukkan bahwa Hill Cipher dan Playfair Cipher yang diimplementasikan di Python menghasilkan ciphertext yang identik dengan CrypTool. Perbedaan kecil terjadi karena cara padding dan penghapusan spasi. Memiliki kelemahan yang sama terhadap huruf ganda dan ganjil.

G. Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa algoritma Hill Cipher dan Playfair Cipher berhasil diimplementasikan dengan baik menggunakan bahasa pemrograman Python. Proses enkripsi dan dekripsi yang dilakukan menunjukkan hasil yang sesuai dengan keluaran dari aplikasi CrypTool, sehingga membuktikan bahwa logika dan perhitungan dalam program berjalan dengan benar. Hill Cipher bekerja dengan prinsip operasi matriks yang melibatkan perhitungan determinan dan invers modulo 26,

sedangkan Playfair Cipher menggunakan metode substitusi digraf dengan tabel kunci 5x5 untuk menggantikan pasangan huruf.