Nama : Ibrahim Dafi Iskandar

NPM : 140810210039

Kelas : A

**Tugas 3 Praktikum Kriptografi**

# Nama      : Ibrahim Dafi Iskandar

# NPM       : 140810210039

# Kelas     : A

# Tanggal   : 26 September 2023

# Deskripsi : Program enkripsi dan dekripsi menggunakan metode Hill Cipher

import numpy as np

def char\_to\_number(x):

    x = ord(x)-65

    return x

def number\_to\_char(x):

    x = chr(x+65)

    return x

def mod\_inverse(A, M):

    for X in range(1, M):

        if (((A % M) \* (X % M)) % M == 1):

            return X

    return -1

def input\_key(n):

    key = list(map(int, input("Masukkan nilai key matrix (e.g 1 1 1 1): ").split()))

    key = np.array(key).reshape(n, n) % 26

    print("Key Matrix: ")

    print(key)

    return key

def input\_text(string):

    text = input("Masukkan " + string + ": ")

    text = text.replace(' ', '').upper()

    return text

def hill(method, text, key, n):

    key\_det = np.linalg.det(key).astype(int)

    if(len(text) % n != 0) :

        last\_char = text[-1]

        text = last\_char\*(n - len(text) % n)

    text\_in\_number = list(map(char\_to\_number, list(text)))

    text\_vector = np.array(text\_in\_number).reshape(int(len(text)/n), n)

    result = np.array([], dtype=int)

    if method == 'dekripsi':

        det\_inverse = mod\_inverse(key\_det % 26, 26)

        key = (

            det\_inverse \* np.round(key\_det \* np.linalg.inv(key)).astype(int) % 26

        )

    for i in range(len(text\_vector)):

            temp = np.matmul(key, text\_vector[i].reshape(n, 1)) % 26

            result = np.append(result, temp)

    result = list(map(number\_to\_char, result))

    output = ''.join(result)

    return output

def find\_key(pt, ct, m):

    pt\_in\_number = list(map(char\_to\_number, list(pt)))

    pt\_vector = np.array(pt\_in\_number).reshape(int(len(pt)/m), m)

    p\_matrix = np.array([], dtype=int)

    ct\_in\_number = list(map(char\_to\_number, list(ct)))

    ct\_vector = np.array(ct\_in\_number).reshape(int(len(ct)/m), m)

    c\_matrix = np.array([], dtype=int)

    for i in range(m):

        c\_matrix = np.append(c\_matrix, ct\_vector[i])

        p\_matrix = np.append(p\_matrix, pt\_vector[i])

    c\_matrix = np.transpose(c\_matrix.reshape(m,m))

    p\_matrix = np.transpose(p\_matrix.reshape(m,m))

    p\_det = int(np.linalg.det(p\_matrix))

    if p\_det % 2 == 0 or p\_det == 13 :

        print("Determinan bukan ganjil selain 13. Key tidak ada karena invers tidak ada.")

        return

    p\_det\_inverse = mod\_inverse(p\_det % 26, 26)

    p\_inverse = (

        p\_det\_inverse \* np.round(p\_det \* np.linalg.inv(p\_matrix)).astype(int) % 26

    )

    key = np.matmul(c\_matrix, p\_inverse) % 26

    return key

while True :

    print("\n=== Program Hill Cipher ===")

    print("1. Enkripsi\n2. Dekripsi\n3. Cari Key\n4. Keluar")

    pilihan = input("Pilihan: ")

    if pilihan == '1' or pilihan == '2':

        n = int(input("\nMasukkan ukuran key matrix (n x n): "))

        key = input\_key(n)

        print("\n")

        text = ''

        while(len(text) < n):

            text = input\_text("text")

            if(len(text) < n):

                print("n harus bilangan prima terkecil sebagai faktor dari jumlah karakter")

        if(pilihan == '1'):

            print("\nPlaintext: " + text)

            output = hill("enkripsi", text, key, n)

            print("Ciphertext: ", output)

        elif pilihan == '2':

            print("\nCiphertext: " + text)

            output = hill("dekripsi", text, key, n)

            print("Plaintext: ", output)

    elif pilihan == '3':

        print("\n")

        pt = input\_text("plaintext")

        ct = input\_text("ciphertext")

        m = int(input("\nMasukkan nilai m: "))

        print("\nPlaintext: " + pt + "\nCiphertext: " + ct)

        key = find\_key(pt, ct, m)

        print("key:")

        print(key)

    elif pilihan == '4':

        exit()

    else :

        print("\nInput tidak sesuai.\n")

Penjelasan kode:

def char\_to\_number(*x*) => fungsi untuk mengubah char ke angka yang bersesuaian

def number\_to\_char(x) => fungsi untuk mengubah angka ke char yang bersesuaian

def mod\_inverse(*A*, *M*) => Fungsi untuk mencari modular multiplicative inverse dari suatu angka

def input\_key(*n*) => Fungsi untuk menginput key matrix

def hill(*method*, *text*, *key*, *n*) => Fungsi untuk melakukan operasi hill cipher

def find\_key(*pt*, *ct*, *m*) => Fungsi untuk mencari key dari cipher text hill cipher

Main menu => menu untuk memilih fitur program