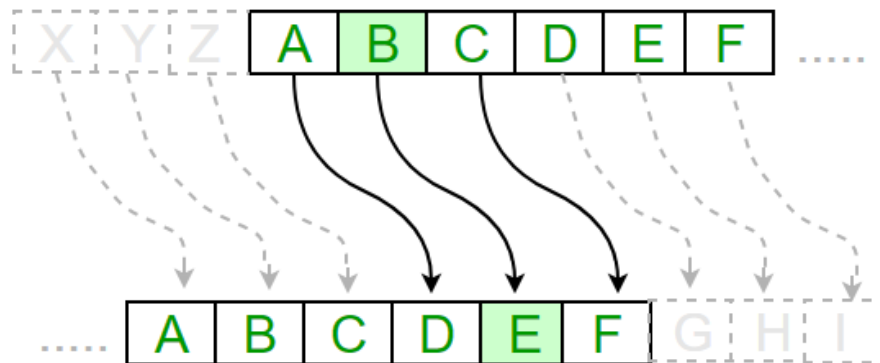


LAPORAN PROJECT 4

Kelompok 1 Big Data D Universitas Udayana.

Pada tugas *project* kali ini, kita akan membuat program kriptografi. Kriptografi sendiri adalah ilmu mengubah pesan yang dapat dipahami ke bentuk yang tidak dapat dipahami, demikian juga sebaliknya. Pada program python ini, teknik kriptografi yang digunakan adalah teknik Caesar Chipper. Algoritma dari teknik ini akan mengubah kalimat yang memiliki karakter alfabet dengan melakukan pergeseran.



Pada program ini juga, kelompok kami memiliki modifikasi khusus yang dianjurkan. Kelompok kami akan melakukan kriptografi dengan menggeser huruf konsonan sebanyak 5 ke kanan dan huruf vocal sebanyak 3 ke kiri. Contoh input nya adalah Av To dengan hasil enkripsi menjadi Xa Yl. Selain fungsi enkripsi, program yang dibangun juga akan memiliki fungsi dekripsi. Enkripsi dilakukan dengan menggeser ke kanan atau menjumlahkan ordinal alphabet dengan konstanta tertentu, sedangkan dekripsi melakukan kebalikannya.

Ketentuan-ketentuan lainnya dari *project* ini diantaranya adalah penggeseran menerapkan konsep rotasi, akan kembali ke huruf awal jika penggeseran ke kanan lebih dari z, atau kembali ke huruf akhir (z) jika ke kiri lebih kecil dari a. Algoritma juga hanya bekerja untuk alfabet, hasil pengubahan pesan tidak mengubah bentuk huruf besar/kecil dari kalimat inputan. Error akan ditampilkan jika ada inputan selain alfabet dan spasi. Program yang dibuat harus menerapkan konsep modul dan menerapkan beberapa fungsi.

Berikut merupakan *code program* beserta dengan penjelasan singkatnya:

1. main.py
#Meng-import module caesarCP untuk dapat dibaca oleh file main.py
import caesarCP
#Menampilkan informasi mengenai kelas serta kelompok
print("""
=====
Kelas : Big Data D
Kelompok : 1
=====
""")
#Menginputkan alfabet yang akan dilakukan kriptografi (enkripsi dan dekripsi)
input = "Av To"

#Assign input alfabet kedalam proses enkripsi

hasil = caesarCP.enkripsi(input)

#Menampilkan input data atau alfabet yang telah ditentukan

print("Ini input data : ", input)

#Menampilkan hasil enkripsi terhadap data atau alfabet yang sudah ditentukan

print("Ini hasil Enkripsi : ", hasil)

#Assign input alfabet kedalam proses dekripsi

res = caesarCP.dekripsi(hasil)

#Menampilkan hasil dekripsi terhadap data atau alfabet yang sudah ditentukan

print("Ini hasil Dekripsi : ", res)

2. caesarCP.py

```
ascii_uppercase = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"]
ascii_lowercase = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]
vokal = ["a", "A", "i", "I", "u", "U", "e", "E", "o", "O"]
tipe = [] # bentuk list global pak

def enkripsi(input_text):
    hasil = ""
    if input_text.isalnum() or " " in input_text:
        None
    else:
        raise NameError
    for huruf in input_text:
        if huruf == " ":
            hasil += huruf
            tipe.append(' ')

        elif huruf in vokal and huruf.isupper():
            pos = ascii_uppercase.index(huruf) - 3
            if pos <= 0: # misal pos = -2
                pos = 26 + pos # 26 - 2
```

```

        pos = ascii_uppercase[pos]
        tipe.append('v')
    else:
        pos = ascii_uppercase[pos]
        tipe.append('v')
    hasil += pos

elif huruf in vokal and huruf.islower():
    # sesuai soal pak vokal -3 konsonan +5
    pos = ascii_lowercase.index(huruf) - 3
    if pos <= 0:
        pos = 26 + pos
        pos = ascii_lowercase[pos]
        tipe.append('v')
    else:
        pos = ascii_lowercase[pos]
        tipe.append('v')
    hasil += pos
elif huruf.islower():
    pos = ascii_lowercase.index(huruf) + 5 # k
onsonan
    if pos >= 26:
        pos = pos - 26
        pos = ascii_lowercase[pos]
        tipe.append('k')
    else: # jika value atau nilai tidak melewa
ti 26
        pos = ascii_lowercase[pos]
        tipe.append('k')
    hasil += pos
elif huruf.isupper():
    pos = ascii_uppercase.index(huruf) + 5
    if pos >= 26:
        pos = pos - 26
        pos = ascii_uppercase[pos]
        tipe.append('k')
    else:
        pos = ascii_uppercase[pos]

```

```

        tipe.append('k')
        hasil += pos

    return hasil

def dekripsi(input_text):
    hasil = ""

    for huruf in input_text:
        if huruf == " ":
            hasil += huruf
        elif tipe[input_text.index(huruf)] == "v" and huruf.isupper():
            pos = ascii_uppercase.index(huruf) + 3
            if pos >= 26:
                pos = pos - 26
            pos = ascii_uppercase[pos]
        else:
            pos = ascii_uppercase[pos]
            hasil += pos

        elif tipe[input_text.index(huruf)] == "v" and huruf.islower():
            pos = ascii_lowercase.index(huruf) + 3
            if pos >= 26:
                pos = pos - 26
            pos = ascii_lowercase[pos]
        else:
            pos = ascii_lowercase[pos]
            hasil += pos

        elif tipe[input_text.index(huruf)] == "k" and huruf.islower():
            pos = ascii_lowercase.index(huruf) - 5
            if pos <= 0:
                pos = 26 + pos
            pos = ascii_lowercase[pos]
        else:
            pos = ascii_lowercase[pos]

```

```

        hasil += pos
    elif tipe[input_text.index(huruf)] == "k" and huruf.isupper():
        pos = ascii_uppercase.index(huruf) - 5
        if pos <= 0:
            pos = 26 + pos
            pos = ascii_uppercase[pos]
        else:
            pos = ascii_uppercase[pos]
        hasil += pos

    return hasil

```

Penjelasan:

Pertama kita inisiasi dulu semua karakter yang termasuk vocal, huruf besar dan huruf kecil untuk digunakan nantinya mengecek inputan dari user.

Kemudian pada enkripsi kita cek apakah inputan sudah merupakan huruf dan spasi saja, jika tidak maka akan memberikan return error. Kemudian untuk karakter spasi apabila terdapat dalam string akan langsung dimasukkan kedalam hasil(string), selanjutnya adalah mengecek karakter termasuk vocal dan huruf, kecil jika iya maka geser 3 karakter ke kiri kemudian masukkan ke dalam variable hasil. Apabila melewati batas yakni kurang dari 0 maka reset dengan 26 – jumlah offset, sehingga karakter akan kembali ke belakang dimulai dari Z kemudian tambahkan pada variable tipe yakni “v”, sebagai memori untuk mengingat urutan vocal. begitu juga untuk karakter vocal Uppercase, dan untuk konsosnon perbedaanya hanya pada pergeserannya diubah ke kanan sebanyak 5 karakter kemudian reset dilakukan dengan offset dikurang 26 karena ada kemungkinan jumlah karakter setelah di offset melewati 26, kemudian save juga pada karakter tipe apabila berbentuk vocal dnegan “v” dan “k” sebagai pertandan huruf konsosnan, begitu seterusnya untuk enkripsi.

Untuk dekripsi tinggal mengecek urutan dari vocal dan konsonan serta huruf besar atau kecil kemudian membalik offset yakni -3 menjadi +3 dan +5 menjadi -5.

3. Hasil Running

```

Ini input data      : Av To
Ini hasil Enkripsi  : Xa Yl
Ini hasil Dekripsi  : Av To
PS C:\Users\Marta\Documents\python_cisco\caesar_cypher> 
.8.8 64-bit ('base': conda) 0 0

```

4. Kesimpulan

Berdasarkan hasil running diatas, dapat disimpulkan bahwa setelah proses enkripsi dan deskripsi yang dilakukan mengikuti ketentuan dengan menggeser huruf konsonan sebanyak 5 ke kanan dan huruf vocal sebanyak 3 ke kiri adalah input alfabet **Av To** akan **mengeluarkan hasil enkripsi Xa Yl**. Begitupun dengan **Xa Yl**, apabila dilakukan dekripsi akan **mengeluarkan hasil Av To**.