

TUGAS 1

KEAMANAN KOMPUTER/KRIPTOGRAFI



NAMA : Muh. Najwar Ramadhan

NIM : 202018

KELAS : 5TKKO-E

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

2022/2023

DAFTAR ISI

Source Code Program

Tampilan Antar Muka Program (Print Screen)

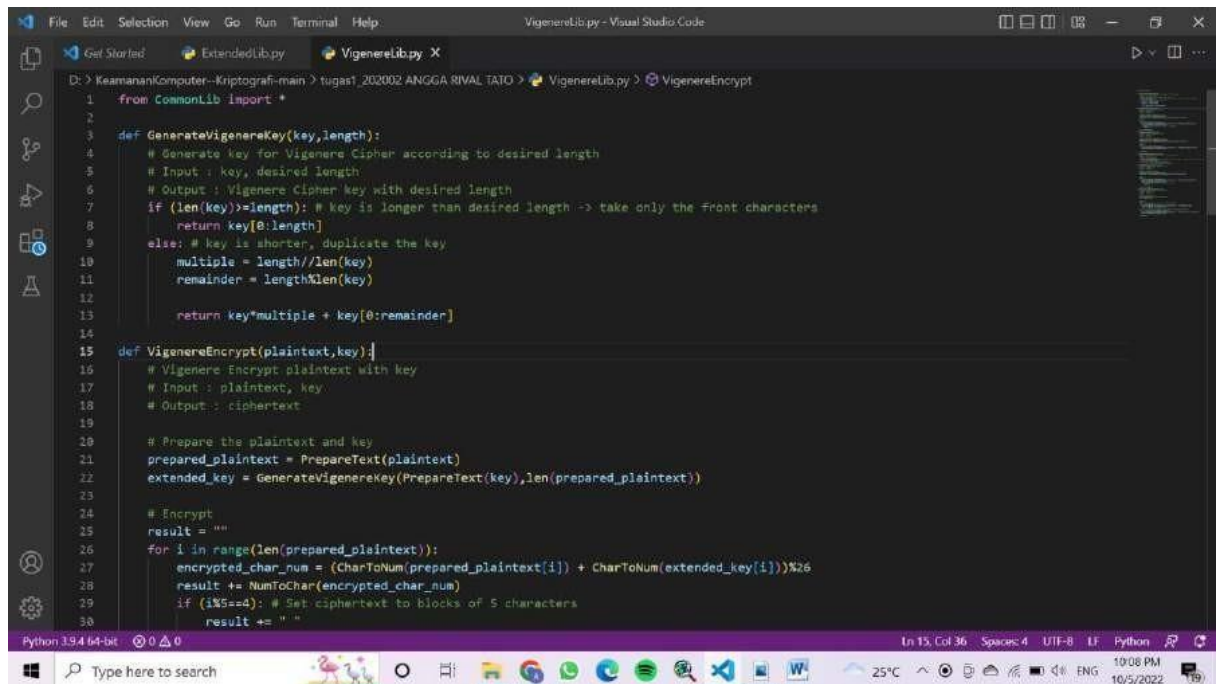
Contoh plainteks dan cipherteks (text, gambar, file database, audio, video)

Link ke github atau google drive yang berisi kode program

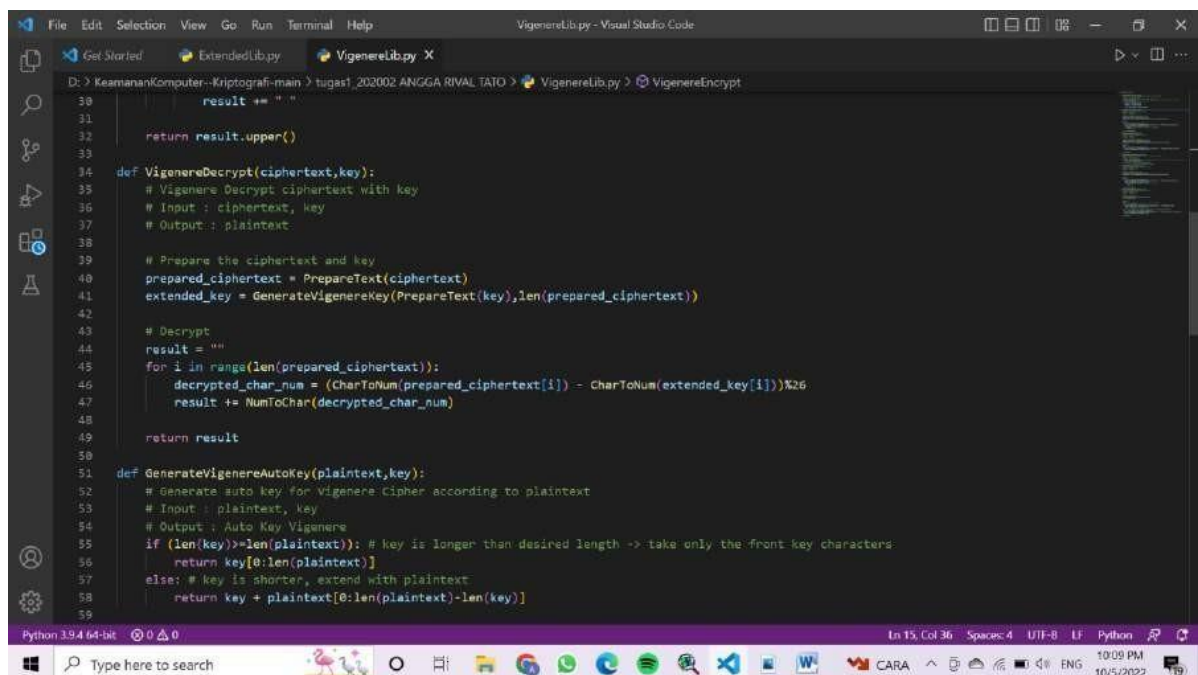
1. Source Code Program

Pembuatan source code di lakukan dalam bahasa python. Pembuatan GUI di lakukan dalam tkinter yang terdapat pada bahasa perograman python. Terdapat 5 file yang ber extension py (python) yang akan di build pada text editor visual studio code.

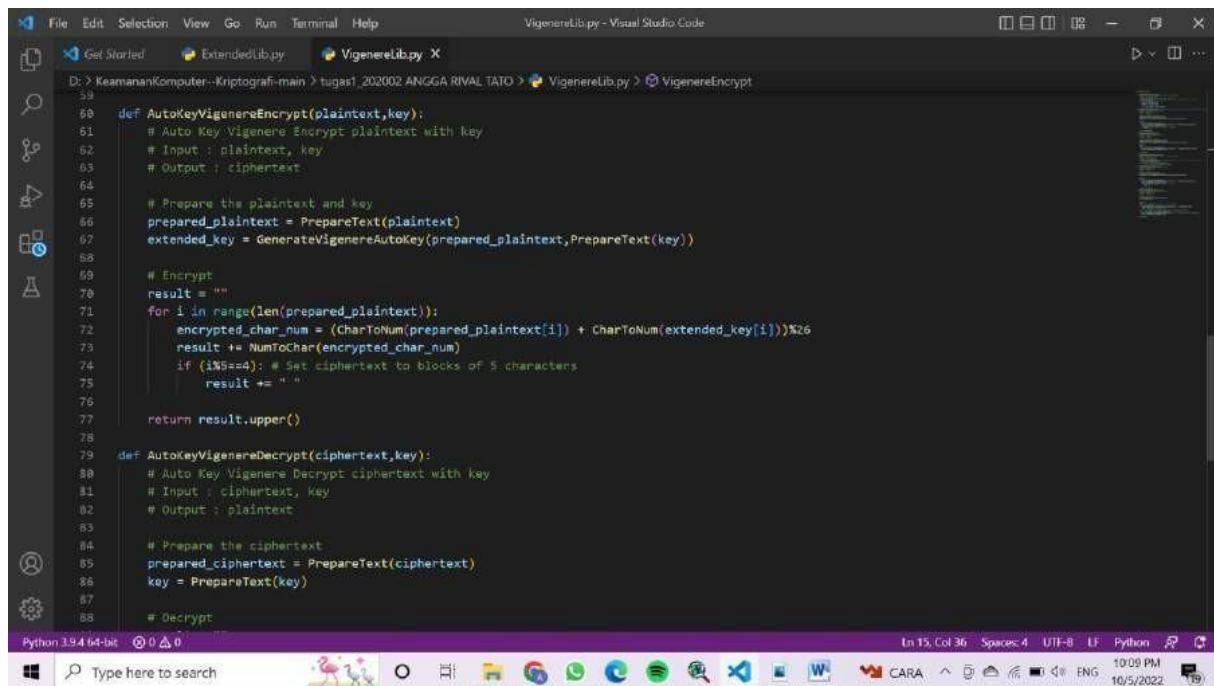
Vigenere Cipher.py



```
1 from CommonLib import *
2
3 def GenerateVigenereKey(key,length):
4     # Generate key for Vigenere Cipher according to desired length
5     # Input : key, desired length
6     # Output : Vigenere Cipher key with desired length
7     if (len(key)>=length): # key is longer than desired length -> take only the front characters
8         return key[0:length]
9     else: # key is shorter, duplicate the key
10         multiple = length//len(key)
11         remainder = length%len(key)
12
13         return key*multiple + key[0:remainder]
14
15 def VigenereEncrypt(plaintext,key):
16     # Vigenere Encrypt plaintext with key
17     # Input : plaintext, key
18     # Output : ciphertext
19
20     # Prepare the plaintext and key
21     prepared_plaintext = PrepareText(plaintext)
22     extended_key = GenerateVigenereKey(PrepareText(key),len(prepared_plaintext))
23
24     # Encrypt
25     result = ""
26     for i in range(len(prepared_plaintext)):
27         encrypted_char_num = (CharToNum(prepared_plaintext[i]) + CharToNum(extended_key[i]))%26
28         result += NumToChar(encrypted_char_num)
29         if (i%5==4): # Set ciphertext to blocks of 5 characters
30             result += " "
```



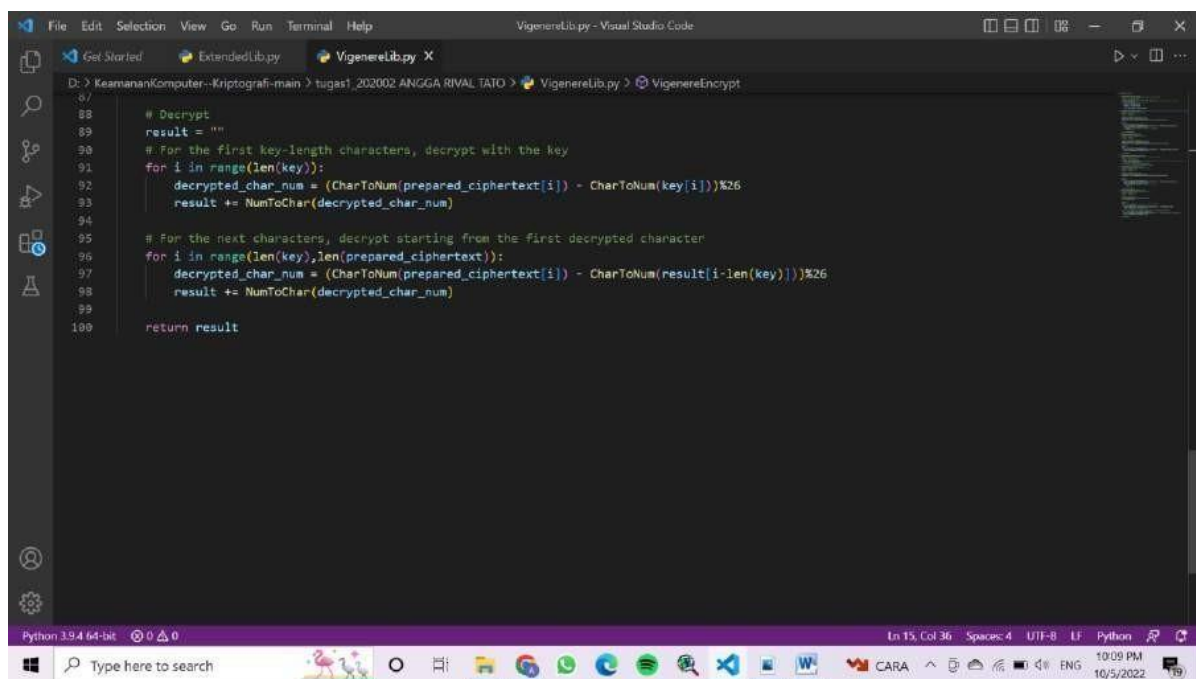
```
30 result += " "
31
32 return result.upper()
33
34 def VigenereDecrypt(ciphertext,key):
35     # Vigenere Decrypt ciphertext with key
36     # Input : ciphertext, key
37     # Output : plaintext
38
39     # Prepare the ciphertext and key
40     prepared_ciphertext = PrepareText(ciphertext)
41     extended_key = GenerateVigenereKey(PrepareText(key),len(prepared_ciphertext))
42
43     # Decrypt
44     result = ""
45     for i in range(len(prepared_ciphertext)):
46         decrypted_char_num = (CharToNum(prepared_ciphertext[i]) - CharToNum(extended_key[i]))%26
47         result += NumToChar(decrypted_char_num)
48
49     return result
50
51 def GenerateVigenereAutoKey(plaintext,key):
52     # Generate auto key for Vigenere Cipher according to plaintext
53     # Input : plaintext, key
54     # Output : Auto Key Vigenere
55     if (len(key)>len(plaintext)): # key is longer than desired length -> take only the front key characters
56         return key[0:len(plaintext)]
57     else: # key is shorter, extend with plaintext
58         return key + plaintext[0:len(plaintext)-len(key)]
59
```



```
File Edit Selection View Go Run Terminal Help
VigenereLib.py - Visual Studio Code

D:\> Keamanankomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > VigenereLib.py > VigenereEncrypt

59
60 def AutoKeyVigenereEncrypt(plaintext,key):
61     # Auto Key Vigenere Encrypt plaintext with key
62     # Input : plaintext, key
63     # Output : ciphertext
64
65     # Prepare the plaintext and key
66     prepared_plaintext = PrepareText(plaintext)
67     extended_key = GenerateVigenereAutoKey(prepared_plaintext,PrepareText(key))
68
69     # Encrypt
70     result = ""
71     for i in range(len(prepared_plaintext)):
72         encrypted_char_num = (CharToNum(prepared_plaintext[i]) + CharToNum(extended_key[i]))%26
73         result += NumToChar(encrypted_char_num)
74         if (i%5==4): # Set ciphertext to blocks of 5 characters
75             result += " "
76
77     return result.upper()
78
79 def AutoKeyVigenereDecrypt(ciphertext,key):
80     # Auto Key Vigenere Decrypt ciphertext with key
81     # Input : ciphertext, key
82     # Output : plaintext
83
84     # Prepare the ciphertext
85     prepared_ciphertext = PrepareText(ciphertext)
86     key = PrepareText(key)
87
88     # Decrypt
```



```
File Edit Selection View Go Run Terminal Help
VigenereLib.py - Visual Studio Code

D:\> Keamanankomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > VigenereLib.py > VigenereDecrypt

87
88     # Decrypt
89     result = ""
90     # For the first key-length characters, decrypt with the key
91     for i in range(len(key)):
92         decrypted_char_num = (CharToNum(prepared_ciphertext[i]) - CharToNum(key[i]))%26
93         result += NumToChar(decrypted_char_num)
94
95     # For the next characters, decrypt starting from the first decrypted character
96     for i in range(len(key),len(prepared_ciphertext)):
97         decrypted_char_num = (CharToNum(prepared_ciphertext[i]) - CharToNum(result[i-len(key)]))%26
98         result += NumToChar(decrypted_char_num)
99
100     return result
```

Extended vigenere cipher.py

```
File Edit Selection View Go Run Terminal Help ExtendedLib.py - Visual Studio Code
Get Started ExtendedLib.py X VigenereLib.py
D:\> KeamanankKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > ExtendedLib.py > ...
1 from VigenereLib import *
2
3 def ExtendedEncrypt(plaintext,key):
4     # Extended Vigenere Encrypt plaintext with key
5     # Input : plaintext, key
6     # Output : ciphertext
7
8     # Prepare the key
9     extended_key = GenerateVigenereKey(key,len(plaintext))
10
11     # Encrypt
12     result = ""
13     for i in range(len(plaintext)):
14         encrypted_char_num = (ord(plaintext[i]) + ord(extended_key[i]))%256
15         result += chr(encrypted_char_num)
16     return result
17
18 def ExtendedDecrypt(ciphertext,key):
19     # Extended Vigenere Decrypt ciphertext with key
20     # Input : ciphertext, key
21     # Output : plaintext
22
23     # Prepare the key
24     extended_key = GenerateVigenereKey(key,len(ciphertext))
25
26     # Decrypt
27     result = ""
28     for i in range(len(ciphertext)):
29         decrypted_char_num = (ord(ciphertext[i]) - ord(extended_key[i]))%256
30         result += chr(decrypted_char_num)
31     return result
32
Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces: 4 UTF-8 LF Python 10:10 PM 10/5/2022
Type here to search
```

```
File Edit Selection View Go Run Terminal Help ExtendedLib.py - Visual Studio Code
Get Started ExtendedLib.py X VigenereLib.py
D:\> KeamanankKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > ExtendedLib.py > ...
4     # Extended Vigenere Encrypt plaintext with key
5     # Input : plaintext, key
6     # Output : ciphertext
7
8     # Prepare the key
9     extended_key = GenerateVigenereKey(key,len(plaintext))
10
11     # Encrypt
12     result = ""
13     for i in range(len(plaintext)):
14         encrypted_char_num = (ord(plaintext[i]) + ord(extended_key[i]))%256
15         result += chr(encrypted_char_num)
16     return result
17
18 def ExtendedDecrypt(ciphertext,key):
19     # Extended Vigenere Decrypt ciphertext with key
20     # Input : ciphertext, key
21     # Output : plaintext
22
23     # Prepare the key
24     extended_key = GenerateVigenereKey(key,len(ciphertext))
25
26     # Decrypt
27     result = ""
28     for i in range(len(ciphertext)):
29         decrypted_char_num = (ord(ciphertext[i]) - ord(extended_key[i]))%256
30         result += chr(decrypted_char_num)
31     return result
32
Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces: 4 UTF-8 LF Python 10:11 PM 10/5/2022
Type here to search
```

Playfair cipher.py

```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code

D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

1 from CommonLib import *
2
3 def GeneratePlayfairKeyMatrix(key):
4     # Create Playfair Key Matrix
5     # Input : key(string)
6     # Output : Playfair Key Matrix from key
7
8     result = ""
9
10    # delete non alphabet characters and 'j'
11    for c in key.lower():
12        if ((c not in result) and (c!='j') and (c.isalpha())):
13            result+=c
14
15    # extend the key with the other characters not in string according to alphabet order (except 'j')
16    for c in range(ord('a'),ord('z')+1):
17        if ((chr(c) not in result) and (chr(c)!='j')):
18            result+=chr(c)
19
20    # create the matrix
21    result_matr = []
22    for i in range(5):
23        new_array = []
24        for j in range(5):
25            new_array.append(result[i*5+j])
26        result_matr.append(new_array)
27
28    return result_matr
29
30 def PlayfairPlaintextBigram(plaintext):
```

```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code

D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

30 def PlayfairPlaintextBigram(plaintext):
31     # Create bigram from plaintext
32     # Input : plaintext
33     # Output : array of playfair bigrams
34
35     # Prepare the plaintext
36     prepared_plaintext = PrepareText(plaintext)
37
38     # Replace j with i
39     replaced_plaintext = ""
40     for c in prepared_plaintext:
41         if (c=='j'):
42             replaced_plaintext += 'i'
43         else:
44             replaced_plaintext += c
45
46     # Create the bigram array
47     bigram_array = []
48     i = 0
49     while (i < len(replaced_plaintext)):
50         if (i==len(replaced_plaintext)-1): # for the last character with no pair, add 'x'
51             bigram = replaced_plaintext[i] + 'x'
52             i = i + 1
53         elif (replaced_plaintext[i]==replaced_plaintext[i+1]): # for bigram with same characters, set 'x' as the first character pair
54             bigram = replaced_plaintext[i] + 'x'
55             i = i + 1
56         else: # create bigram with the next character
57             bigram = replaced_plaintext[i] + replaced_plaintext[i+1]
58             i = i + 2
59     return bigram_array
```



```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code

D:\> KeenanKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

60     bigram_array.append(bigram)
61
62     return bigram_array
63
64 def PlayfairCiphertextBigram(ciphertext):
65     # Create bigram from ciphertext
66     # Input : ciphertext
67     # Output : array of playfair bigrams
68
69     # Prepare the ciphertext
70     prepared_ciphertext = PrepareText(ciphertext)
71
72     # Create the bigram array
73     bigram_array = []
74     i = 0
75     while (i < len(prepared_ciphertext)):
76         # pair every two characters
77         # for playfair ciphertext length, it is always expected to have even length
78         bigram = prepared_ciphertext[i] + prepared_ciphertext[i+1]
79         bigram_array.append(bigram)
80         i = i + 2
81
82     return bigram_array
83
84 def FindPlayfairIndex(bigram, key_matr):
85     # Find bigram index position in key matrix
86     # Input : bigram to be searched, key matrix
87     # Output : (x,y) index of bigram characters
88     found0 = False
89     found1 = False
90
91     for i in range(5):
92         for j in range(5):
93             if (key_matr[i][j] == bigram[0]):
94                 x0 = j
95                 y0 = i
96                 found0 = True
97             if (key_matr[i][j] == bigram[1]):
98                 x1 = j
99                 y1 = i
100                 found1 = True
101             if ((found0) and (found1)):
102                 break
103         if ((found0) and (found1)):
104             break
105
106     return x0,y0,x1,y1
107
108 def PlayfairEncrypt(plaintext, key):
109     # Playfair Encrypt plaintext with key
110     # Input : plaintext, key
111     # Output : ciphertext
112
113     # Prepare the bigram and key matrix
114     plaintext_bigram = PlayfairPlaintextBigram(plaintext)
115     playfair_key = GeneratePlayfairKeyMatrix(key)
116
117     # Encrypt
118     encrypted_text = ""
```

```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code

D:\> KeenanKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

88     found0 = False
89     found1 = False
90     for i in range(5):
91         for j in range(5):
92             if (key_matr[i][j] == bigram[0]):
93                 x0 = j
94                 y0 = i
95                 found0 = True
96             if (key_matr[i][j] == bigram[1]):
97                 x1 = j
98                 y1 = i
99                 found1 = True
100             if ((found0) and (found1)):
101                 break
102         if ((found0) and (found1)):
103             break
104
105     return x0,y0,x1,y1
106
107 def PlayfairEncrypt(plaintext, key):
108     # Playfair Encrypt plaintext with key
109     # Input : plaintext, key
110     # Output : ciphertext
111
112     # Prepare the bigram and key matrix
113     plaintext_bigram = PlayfairPlaintextBigram(plaintext)
114     playfair_key = GeneratePlayfairKeyMatrix(key)
115
116     # Encrypt
117     encrypted_text = ""
```

```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code
Get Started ExtendedLib.py PlayfairLib.py X VigenereLib.py
D:\> KeenanKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

116 # Encrypt
117 encrypted_text = ""
118 for bigram in plaintext_bigram:
119     x0,y0,x1,y1 = FindPlayfairIndex(bigram,playfair_key)
120     if (x0==x1): # same row -> take the next characters
121         encrypted_bigram = playfair_key[(y0+1)%5][x0] + playfair_key[(y1+1)%5][x1]
122     elif (y0==y1): # same column -> take the next characters
123         encrypted_bigram = playfair_key[y0][(x0+1)%5] + playfair_key[y1][(x1+1)%5]
124     else: # different row and column -> take the rectangle angle characters
125         encrypted_bigram = playfair_key[y0][x1] + playfair_key[y1][x0]
126
127     encrypted_text += encrypted_bigram
128
129 # Split into blocks of 5 characters
130 result = ""
131 for i in range(len(encrypted_text)):
132     result += encrypted_text[i].upper()
133     if (i%5==4):
134         result += ' '
135
136 return result
137
138 def PlayfairDecrypt(ciphertext,key):
139     # Playfair Decrypt ciphertext with key
140     # Input : ciphertext, key
141     # Output : plaintext
142
143     # Prepare the bigram and key matrix
144     ciphertext_bigram = PlayfairCiphertextBigram(ciphertext)
145     playfair_key = GeneratePlayfairKeyMatrix(key)
```

```
File Edit Selection View Go Run Terminal Help
PlayfairLib.py - Visual Studio Code
Get Started ExtendedLib.py PlayfairLib.py X VigenereLib.py
D:\> KeenanKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > PlayfairLib.py > ...

146 # Decrypt
147 decrypted_text = ""
148 for bigram in ciphertext_bigram:
149     x0,y0,x1,y1 = FindPlayfairIndex(bigram,playfair_key)
150     if (x0==x1): # same row -> take the previous character
151         decrypted_bigram = playfair_key[(y0-1)%5][x0] + playfair_key[(y1-1)%5][x1]
152     elif (y0==y1): # same column -> take the previous character
153         decrypted_bigram = playfair_key[y0][(x0-1)%5] + playfair_key[y1][(x1-1)%5]
154     else: # different row and column -> take the rectangle angle characters
155         decrypted_bigram = playfair_key[y0][x1] + playfair_key[y1][x0]
156
157     decrypted_text += decrypted_bigram
158
159 # Delete the inserted 'x'
160 result = ""
161 for i in range(len(decrypted_text)):
162     if (decrypted_text[i]=='x'):
163         if (i==(len(decrypted_text)-1)): # skip the last 'x' character
164             pass
165         elif (decrypted_text[i-1]==decrypted_text[i+1]): # skip the 'x' character between two same characters
166             pass
167         else:
168             result += decrypted_text[i]
169     else:
170         result += decrypted_text[i]
171
172 return result
173
```

Enigma cipher.py


```
File Edit Selection View Go Run Terminal Help
AffineLib.py - Visual Studio Code
Get Started ExtendedLib.py AffineLib.py X FullVigenerLib.py GUI.py main.py PlayfairLib.py CommonLib.py
D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > AffineLib.py > ...
1 from CommonLib import *
2
3 def AffineEncrypt(plaintext,multiple,offset):
4     # Affine Encrypt
5     # Input : plaintext, multiple, offset
6     # Output : ciphertext
7
8     # Prepare the plaintext
9     prepared_plaintext = PrepareText(plaintext)
10
11     # Encrypt
12     result = ""
13     for i in range(len(prepared_plaintext)):
14         encrypted_char_num = (multiple*CharToNum(prepared_plaintext[i]) + offset)%26
15         result += NumToChar(encrypted_char_num)
16         if (i%5==4): # Set ciphertext to blocks of 5 characters
17             result += " "
18     return result.upper()
19
20
21 def AffineDecrypt(ciphertext,multiple,offset):
22     # Affine Decrypt
23     # Input : ciphertext, multiple, offset
24     # Output : plaintext
25
26     # Prepare the ciphertext
27     prepared_ciphertext = PrepareText(ciphertext)
28
29     # Find modulo inverse
30     for i in range(1,26):
```

```
File Edit Selection View Go Run Terminal Help
AffineLib.py - Visual Studio Code
Get Started ExtendedLib.py AffineLib.py X FullVigenerLib.py GUI.py main.py PlayfairLib.py CommonLib.py
D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > AffineLib.py > ...
29     # Find modulo inverse
30     for i in range(1,26):
31         if ((multiple*i)%26==1):
32             inverse_modulo = i
33             break
34
35     # Decrypt
36     result = ""
37     for i in range(len(prepared_ciphertext)):
38         encrypted_char_num = inverse_modulo*(CharToNum(prepared_ciphertext[i]) - offset)%26
39         result += NumToChar(encrypted_char_num)
40
41     return result
```

One-time pad.py

```
File Edit Selection View Go Run Terminal Help
FullVigenerLib.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenerLib.py X GUI.py PlayfairLib.py CommonLib.py Components.py VigenerLib.py

D:\> Keenanankomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > FullVigenerLib.py > ...
1 from CommonLib import *
2 from VigenerLib import *
3
4 def MatrixFullVigener(key):
5     arr_key = []
6     for i in range(len(key)):
7         num_key = CharToNum(key[i])
8         arr_key.append(i)
9
10    abjadUrut = []
11    for i in range(26):
12        abjadUrut.append(i)
13
14    X = 0
15    tempAbjad = abjadUrut.copy()
16    matVig = []
17    for i in range(26):
18        rowVig = []
19        for j in range(25):
20            X = X + arr_key[j%len(arr_key)]
21            X = X%len(tempAbjad)
22            rowVig.append(tempAbjad[X])
23            tempAbjad.pop(X)
24            X = tempAbjad[0]
25            rowVig.append(tempAbjad[0])
26            tempAbjad.pop(0)
27            tempAbjad = rowVig.copy()
28            matVig.append(rowVig)
29    return matVig
30
```

Python 3.9.4 64-bit 0 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 10:20 PM 10/5/2022

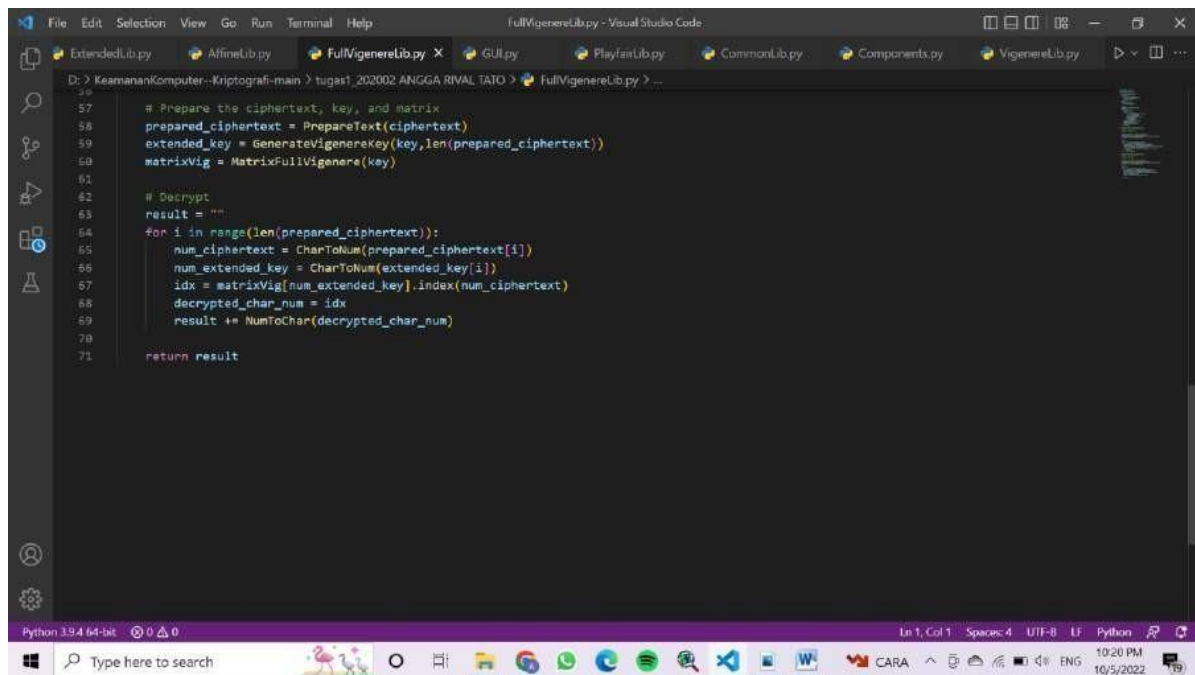
```
File Edit Selection View Go Run Terminal Help
FullVigenerLib.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenerLib.py X GUI.py PlayfairLib.py CommonLib.py Components.py VigenerLib.py

D:\> Keenanankomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > FullVigenerLib.py > ...
31 def FullVigenerEncrypt(plaintext,key):
32     # Vigener Encrypt plaintext with key
33     # Input : plaintext, key
34     # Output : ciphertext
35
36     # Prepare the plaintext, key, and matrix
37     prepared_plaintext = PrepareText(plaintext)
38     extended_key = GenerateVigenerKey(key,len(prepared_plaintext))
39     matrixVig = MatrixFullVigener(key)
40
41     # Encrypt
42     result = ""
43     for i in range(len(prepared_plaintext)):
44         num_plaintext = CharToNum(prepared_plaintext[i])
45         num_extended_key = CharToNum(extended_key[i])
46         encrypted_char_num = matrixVig[num_extended_key][num_plaintext]
47         result += NumToChar(encrypted_char_num)
48         if (i%5==4): # Set ciphertext to blocks of 5 characters
49             result += " "
50     return result.upper()
51
52 def FullVigenerDecrypt(ciphertext,key):
53     # Vigener Decrypt ciphertext with key
54     # Input : ciphertext, key
55     # Output : plaintext
56
57     # Prepare the ciphertext, key, and matrix
58     prepared_ciphertext = PrepareText(ciphertext)
59     extended_key = GenerateVigenerKey(key,len(prepared_ciphertext))
60     matrixVig = MatrixFullVigener(key)

```

Python 3.9.4 64-bit 0 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 10:20 PM 10/5/2022



GUI.py

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py x PlayfairLib.py CommonLib.py Components.py VigenereLib.py

D:\KeamananKomputer-Kriptografi-main> tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

1 import tkinter as tk
2 import tkinter.scrolledtext as st
3 import tkinter.filedialog as fd
4 import random
5 import math
6 import pickle
7
8 from Components import *
9
10 from CommonLib import *
11 from VigenereLib import *
12 from PlayfairLib import *
13 from AffineLib import *
14 from FullVigenereLib import *
15 from ExtendedLib import *
16
17 class GUI:
18     def __init__(self, parent):
19         #--- init ---#
20         self.parent = parent
21         parent.title("Kriptografi")
22         self.mode = "Vigenere"
23
24         #--- define grid ---#
25         parent.columnconfigure([0,1,2,3], weight=1)
26         parent.rowconfigure([0,1,2], weight=1, minsize=100)
27
28         #--- plaintext ---#
29         self.plaintext = TextFrame(
30             title="Plaintext",
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py x PlayfairLib.py CommonLib.py Components.py VigenereLib.py

D:\KeamananKomputer-Kriptografi-main> tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

28 #--- plaintext ---#
29 self.plaintext = TextFrame(
30     title="Plaintext",
31     width=50,
32     height=5
33 )
34 self.plaintext.frame.grid(row=0, column=0, columnspan=3)
35
36 #--- encrypt button ---#
37 self.encrypt_button = tk.Button(text="Encrypt", command=self.Encrypt)
38 self.encrypt_button.grid(row=1, column=0, padx=10, pady=10)
39
40 #--- keyframe ---#
41 self.keyframe = KeyFrame(title="Key", width=34)
42 self.keyframe.button.bind("<Button-1>", self.RandomizeKey)
43 self.keyframe.frame.grid(row=1, column=1)
44
45 #--- affinekeyframe ---#
46 self.affinekeyframe = AffineKeyFrame(width=6)
47 # is created only but not packed
48
49 #--- decrypt button ---#
50 self.decrypt_button = tk.Button(text="Decrypt", command=self.Decrypt)
51 self.decrypt_button.grid(row=1, column=2, padx=10, pady=10)
52
53 #--- ciphertext ---#
54 self.ciphertext = TextFrame(
55     title="Ciphertext",
56     width=50,
57     height=5
```



```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
72
73 #--- ciphertext ---#
74 self.ciphertext = TextFrame(
75     title="Ciphertext",
76     width=50,
77     height=5
78 )
79 self.ciphertext.frame.grid(row=2,column=0,columnspan=3)
80
81 #--- cipher method button list ---#
82 cipher_method_list = ["Vigenere","Full Vigenere","Auto-Key Vigenere","Extended Vigenere","Playfair","Affine"]
83 self.cipher_method_frame = ButtonListFrame(
84     title = "Method : Vigenere",
85     labels = cipher_method_list,
86     width = 25
87 )
88 for button in self.cipher_method_frame.button_list:
89     button.bind("<Button-1>",lambda event,mode=button["text"]: self.ChangeMode(event,mode))
90 self.cipher_method_frame.frame.grid(row=0,column=3,rowspan=3,sticky="ns")
91
92 #--- file frame ---#
93 file_method_list = ["Open Plaintext Textfile","Open Ciphertext Textfile","Save Plaintext to Textfile","Save Ciphertext to Textfile"]
94 self.file_frame = ButtonListFrame(
95     title = "File",
96     labels = file_method_list,
97     width = 25
98 )
99 self.file_frame.button_list[0].bind("<Button-1>",lambda event,text="plaintext": self.OpenFileText(event,text))
100 self.file_frame.button_list[1].bind("<Button-1>",lambda event,text="ciphertext": self.OpenFileText(event,text))
101 self.file_frame.button_list[2].bind("<Button-1>",lambda event,text="plaintext": self.SaveFileText(event,text))
102 self.file_frame.button_list[3].bind("<Button-1>",lambda event,text="ciphertext": self.SaveFileText(event,text))
103
Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces:4 UTF-8 LF Python 10:21 PM 10/5/2022
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
78
79 self.file_frame.button_list[0].bind("<Button-1>",lambda event,text="plaintext": self.OpenFileText(event,text))
80 self.file_frame.button_list[1].bind("<Button-1>",lambda event,text="ciphertext": self.OpenFileText(event,text))
81 self.file_frame.button_list[2].bind("<Button-1>",lambda event,text="plaintext": self.SaveFileText(event,text))
82 self.file_frame.button_list[3].bind("<Button-1>",lambda event,text="ciphertext": self.SaveFileText(event,text))
83 self.file_frame.button_list[4].bind("<Button-1>",self.BinaryFileDialog)
84 self.file_frame.frame.grid(row=2,column=3)
85
86 def ChangeMode(self,event,mode):
87     # Event handler when button in cipher button list is pressed
88     # Change cipher mode and the text
89     if (mode=="Affine" and self.mode=="Affine"): # Change to Affine from not Affine, change the frame
90         self.keyframe.frame.grid_forget()
91         self.affinekeyframe.frame.grid(row=1,column=1)
92     elif (mode!="Affine" and self.mode=="Affine"): # Clicked not Affine From Affine, change the frame
93         self.affinekeyframe.frame.grid_forget()
94         self.keyframe.frame.grid(row=1,column=1)
95
96     self.mode = mode
97     self.cipher_method_frame.label["text"] = "Method : " + mode
98
99 def RandomizeKey(self,event):
100     # Create randomized key according to input length
101
102     # Take input length
103     length = self.keyframe.random_entry.get()
104
105     if (len(length)==0): # No length is inputted
106         self.AlertWindow("Please insert randomizer length")
107     elif (not length.isnumeric()): # Inputted length is not a number
108
Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces:4 UTF-8 LF Python 10:22 PM 10/5/2022
```



```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
102 # Take input length
103 length = self.keyframe.random_entry.get()
104
105 if (len(length)==0): # No length is inputted
106     self.AlertWindow("Please insert randomizer length")
107 elif (not length.isnumeric()): # Inputted length is not a number
108     self.AlertWindow("Please insert randomizer length in number")
109 else:
110     # Generate random string
111     randomizer = ""
112     for i in range(int(length)):
113         randomizer += NumToChar(random.randint(0,26))
114
115     self.keyframe.entry.delete(0,tk.END)
116     self.keyframe.entry.insert(0,randomizer)
117
118
119 def Encrypt(self):
120     # Event handler when encrypt button is pressed
121     # Encrypt plaintext and key
122
123     if (self.mode=="Affine"): # for methods other than Affine
124         # Take the plaintext and key from the field
125         plaintext = self.plaintext.entry.get("1.0",tk.END)[:1]
126         key = self.keyframe.entry.get()
127
128         # Check for validity
129         if (len(plaintext)==0): # Empty plaintext
130             self.AlertWindow("Please insert plaintext")
131         elif (len(key)==0): # Empty key
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
127
128 # Check for validity
129 if (len(plaintext)==0): # Empty plaintext
130     self.AlertWindow("Please insert plaintext")
131 elif (len(key)==0): # Empty key
132     self.AlertWindow("Please insert key")
133 else:
134     # Encrypt
135     if (self.mode=="Vigenere"): # Vigenere
136         ciphertext = VigenereEncrypt(plaintext,key)
137     elif (self.mode=="Full Vigenere"): # Full Vigenere
138         ciphertext = FullVigenereEncrypt(plaintext,key)
139     elif (self.mode=="Auto-Key Vigenere"): # Auto key
140         ciphertext = AutoKeyVigenereEncrypt(plaintext,key)
141     elif (self.mode=="Extended Vigenere"): # Extended
142         ciphertext = ExtendedEncrypt(plaintext,key)
143     elif (self.mode=="Playfair"): # Playfair
144         ciphertext = PlayfairEncrypt(plaintext,key)
145
146     # Insert into ciphertext field
147     self.ciphertext.entry.delete("1.0",tk.END)
148     self.ciphertext.entry.insert("1.0",ciphertext)
149
150 else: # Affine
151     # Take the plaintext and parameters from the field
152     plaintext = self.plaintext.entry.get("1.0",tk.END)[:1]
153     multiple = self.affinekeyframe.multiple_entry.get()
154     offset = self.affinekeyframe.offset_entry.get()
155
156     # Check for validity
```

```
File Edit Selection View Go Run Terminal Help
GUILpy - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUILpy X PlayfairLib.py CommonLib.py Components.py VigenereLib.py

D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUILpy > ...

149
150 else: # Affine
151     # Take the plaintext and parameters from the field
152     plaintext = self.plaintext.entry.get("1.0",tk.END)[-1:]
153     multiple = self.affinekeyframe.multiple_entry.get()
154     offset = self.affinekeyframe.offset_entry.get()
155
156     # Check for validity
157     if (len(plaintext)==0): # Empty plaintext
158         self.AlertWindow("Please insert plaintext")
159     elif (not multiple.isnumeric() or not offset.isnumeric()): # Non numeric multiple and offset
160         self.AlertWindow("Multiple and offset is a number")
161     else:
162         # Encrypt
163         multiple = int(multiple)
164         offset = int(offset)
165         if (math.gcd(multiple,26)!=1):
166             self.AlertWindow("Multiple is not relative prime of 26")
167         else:
168             # Insert into ciphertext field
169             ciphertext = AffineEncrypt(plaintext,multiple,offset)
170             self.ciphertext.entry.delete("1.0",tk.END)
171             self.ciphertext.entry.insert("1.0",ciphertext)
172
173
174 def Decrypt(self):
175     # Event handler when decrypt button is pressed
176     # Decrypt ciphertext and key
177
178     if (self.mode!="Affine"): # for methods other than Affine
```

```
File Edit Selection View Go Run Terminal Help
GUILpy - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUILpy X PlayfairLib.py CommonLib.py Components.py VigenereLib.py

D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUILpy > ...

173
174 def Decrypt(self):
175     # Event handler when decrypt button is pressed
176     # Decrypt ciphertext and key
177
178     if (self.mode!="Affine"): # for methods other than Affine
179         # Take the ciphertext and key from the field
180         key = self.keyframe.entry.get()
181         ciphertext = self.ciphertext.entry.get("1.0",tk.END)[-1:]
182
183         # Check for validity
184         if (len(ciphertext)==0): # Empty ciphertext
185             self.AlertWindow("Please insert ciphertext")
186         elif (len(key)==0): # Empty key
187             self.AlertWindow("Please insert key")
188         else:
189             # Decrypt
190             if (self.mode=="Vigenere"): # Vigenere
191                 plaintext = VigenereDecrypt(ciphertext,key)
192             elif (self.mode=="Full Vigenere"): # Full Vigenere
193                 plaintext = FullVigenereDecrypt(ciphertext,key)
194             elif (self.mode=="Auto-Key Vigenere"): # Auto Key
195                 plaintext = AutoKeyVigenereDecrypt(ciphertext,key)
196             elif (self.mode=="Extended Vigenere"): # Extended
197                 plaintext = ExtendedDecrypt(ciphertext,key)
198             elif (self.mode=="Playfair"): # Playfair
199                 plaintext = PlayfairDecrypt(ciphertext,key)
200
201             # Insert into plaintext field
202             self.plaintext.entry.delete("1.0",tk.END)
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayanLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

200
201     # Insert into plaintext field
202     self.plaintext.entry.delete("1.0",tk.END)
203     self.plaintext.entry.insert("1.0",plaintext)
204
205 else: # Affine
206     # Take the plaintext and parameters from the field
207     ciphertext = self.ciphertext.entry.get("1.0",tk.END)[:1]
208     multiple = self.affinekeyframe.multiple_entry.get()
209     offset = self.affinekeyframe.offset_entry.get()
210
211     # Check for validity
212     if (len(ciphertext)==0): # Empty plaintext
213         self.AlertWindow("Please insert plaintext")
214     elif (not multiple.isnumeric()) or not offset.isnumeric(): # Non numeric multiple and offset
215         self.AlertWindow("Multiple and offset is a number")
216     else:
217         # Decrypt
218         multiple = int(multiple)
219         offset = int(offset)
220         if (math.gcd(multiple,26)!=1):
221             self.AlertWindow("Multiple and is not relative prime of 26")
222         else:
223             # Insert into ciphertext field
224             plaintext = AffineDecrypt(ciphertext,multiple,offset)
225             self.plaintext.entry.delete("1.0",tk.END)
226             self.plaintext.entry.insert("1.0",plaintext)
227
228 def OpenFileText(self,event,text):
229     # Open file using open file dialog
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayanLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

227
228 def OpenFileText(self,event,text):
229     # Open file using open file dialog
230
231     # Take filename
232     filename = fd.askopenfilename(
233         initialdir = "/",
234         title = "Select " + text + " file",
235         filetypes = [("Text files (.txt)","*.txt")]
236     )
237
238     if (filename!=""): # If filename is chosen
239         file = open(filename,"rt")
240         content = file.read()
241
242         file.close()
243
244         if (text=="plaintext"): # For plaintext, insert to plaintext field
245             self.plaintext.entry.delete("1.0",tk.END)
246             self.plaintext.entry.insert("1.0",content)
247         elif (text=="ciphertext"): # For ciphertext, insert to ciphertext field
248             self.ciphertext.entry.delete("1.0",tk.END)
249             self.ciphertext.entry.insert("1.0",content)
250
251         return "break"
252
253 def SaveFileText(self,event,text):
254     # Save file using save file dialog
255
256     # Take filename
```



```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
251     return "break"
252
253 def SavefileText(self,event,text):
254     # Save file using save file dialog
255
256     # Take filename
257     filename = fd.asksaveasfilename(
258         initialdir = "/",
259         title = "Select " + text + " file",
260         filetypes = [("Text files (.txt)","*.txt")],
261         defaulttextextension = [("Text files (.txt)","*.txt")]
262     )
263
264     if (filename!=""): # If File name is chosen
265         file = open(filename,"wt")
266         if (text=="plaintext"): # For plaintext, insert the plaintext
267             plaintext = self.plaintext.entry.get("1.0",tk.END)[:1]
268             file.write(plaintext)
269         elif (text=="ciphertext"): # For ciphertext, insert the ciphertext
270             ciphertext = self.ciphertext.entry.get("1.0",tk.END)[:1]
271             file.write(ciphertext)
272
273         file.close()
274
275     return "break"
276
277 def AlertWindow(self,text):
278     # Create new window for alert
279     # Components : Label with input text and dismiss button
280     alert_window = tk.Toplevel(self.parent)
281     alert_window.title("Alert")
282
283     tk.Label(master=alert_window,text=text).pack(padx=120,pady=20)
284     tk.Button(master=alert_window,text="Ok",width=10,command=lambda:alert_window.destroy()).pack(pady=10)
285
286     alert_window.grab_set()
287
288 def BinaryFileWindow(self,event):
289     # Create new window for file encrypt/decrypt
290     # Components : Label, key entry, and buttons
291     new_window = tk.Toplevel(self.parent)
292     new_window.title("Binary File")
293
294     self.binary_file = ""
295
296     self.file_label = tk.Label(master=new_window,text="File : " + self.binary_file)
297     self.file_label.grid(row=0,column=0,columnspan=2,sticky="we",padx=120,pady=2)
298     self.key_label = tk.Label(master=new_window,text="Key : ")
299     self.key_label.grid(row=1,column=0,columnspan=2,pady=2)
300     self.key_entry = tk.Entry(master=new_window,width=15)
301     self.key_entry.grid(row=2,column=0,columnspan=2,pady=2)
302     tk.Button(master=new_window,text="Choose File",width=20,command=self.ChooseFileBinary).grid(row=3,column=0,columnspan=2,pady=2)
303     tk.Button(master=new_window,text="Encrypt and Save",width=20,command=self.SaveEncryptedFile).grid(row=4,column=0,columnspan=2,pady=2)
304     tk.Button(master=new_window,text="Decrypt and Save",width=20,command=self.SaveDecryptedFile).grid(row=5,column=0,columnspan=2,pady=2)
305
Python 3.9.4 64-bit
Type here to search
10:23 PM 10/5/2022
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...
275     return "break"
276
277 def AlertWindow(self,text):
278     # Create new window for alert
279     # Components : Label with input text and dismiss button
280     alert_window = tk.Toplevel(self.parent)
281     alert_window.title("Alert")
282
283     tk.Label(master=alert_window,text=text).pack(padx=120,pady=20)
284     tk.Button(master=alert_window,text="Ok",width=10,command=lambda:alert_window.destroy()).pack(pady=10)
285
286     alert_window.grab_set()
287
288 def BinaryFileWindow(self,event):
289     # Create new window for file encrypt/decrypt
290     # Components : Label, key entry, and buttons
291     new_window = tk.Toplevel(self.parent)
292     new_window.title("Binary File")
293
294     self.binary_file = ""
295
296     self.file_label = tk.Label(master=new_window,text="File : " + self.binary_file)
297     self.file_label.grid(row=0,column=0,columnspan=2,sticky="we",padx=120,pady=2)
298     self.key_label = tk.Label(master=new_window,text="Key : ")
299     self.key_label.grid(row=1,column=0,columnspan=2,pady=2)
300     self.key_entry = tk.Entry(master=new_window,width=15)
301     self.key_entry.grid(row=2,column=0,columnspan=2,pady=2)
302     tk.Button(master=new_window,text="Choose File",width=20,command=self.ChooseFileBinary).grid(row=3,column=0,columnspan=2,pady=2)
303     tk.Button(master=new_window,text="Encrypt and Save",width=20,command=self.SaveEncryptedFile).grid(row=4,column=0,columnspan=2,pady=2)
304     tk.Button(master=new_window,text="Decrypt and Save",width=20,command=self.SaveDecryptedFile).grid(row=5,column=0,columnspan=2,pady=2)
305
Python 3.9.4 64-bit
Type here to search
10:23 PM 10/5/2022
```

```
File Edit Selection View Go Run Terminal Help
GUL.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUL.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUL.py > ...

296 self.file_label = tk.Label(master=new_window, text="File : " + self.binary_file)
297 self.file_label.grid(row=8, column=0, columnspan=2, sticky="we", padx=120, pady=2)
298 self.key_label = tk.Label(master=new_window, text="Key : ")
299 self.key_label.grid(row=2, column=0, columnspan=2, pady=2)
300 self.key_entry = tk.Entry(master=new_window, width=15)
301 self.key_entry.grid(row=2, column=0, columnspan=2, pady=2)
302 tk.Button(master=new_window, text="Choose File", width=20, command=self.ChooseFileBinary).grid(row=3, column=0, columnspan=2, pady=2)
303 tk.Button(master=new_window, text="Encrypt and Save", width=20, command=self.SaveEncryptedFile).grid(row=4, column=0, columnspan=2, pady=2)
304 tk.Button(master=new_window, text="Decrypt and Save", width=20, command=self.SaveDecryptedFile).grid(row=5, column=0, columnspan=2, pady=2)
305 tk.Button(master=new_window, text="Unselect File", width=20, command=self.UnselectFile).grid(row=6, column=0, columnspan=2, pady=2)
306
307 new_window.grab_set()
308
309 def ChooseFileBinary(self):
310     # Take filename
311     filename = fd.askopenfilename(
312         initialdir = "/",
313         title = "Select file",
314         filetypes = [("Text files (.txt)", "*.txt"), ("Binary files (.bin)", "*.bin"), ("All files", "*.*")],
315     )
316
317     if (filename!=""):
318         self.file_label["text"] = "File : " + filename
319         self.binary_file = filename
320
321 def SaveEncryptedFile(self):
322     # buka file di self.binary_file
323     if (self.binary_file==""):
324         self.AlertWindow("Please choose a file")
325
326 Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces: 4 UTF-8 LF Python 10:23 PM 10/5/2022
```

```
File Edit Selection View Go Run Terminal Help
GUL.py - Visual Studio Code

ExtendedLib.py AffineLib.py FullVigenereLib.py GUL.py X PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUL.py > ...

317 if (filename!=""):
318     self.file_label["text"] = "File : " + filename
319     self.binary_file = filename
320
321 def SaveEncryptedFile(self):
322     # buka file di self.binary_file
323     if (self.binary_file==""):
324         self.AlertWindow("Please choose a file")
325     else:
326         #encrypt
327         key = self.key_entry.get()
328         if (len(key)==0):
329             self.AlertWindow("Please insert key")
330         else:
331             file = open(self.binary_file, "r")
332             readPlaintext = file.read()
333             encrptChipertext = ExtendedEncrypt(readPlaintext, key)
334             file.close()
335             # save
336             filename = fd.asksaveasfilename(
337                 initialdir = "/",
338                 title = "Save file",
339                 filetypes = [("Binary files (.bin)", "*.bin"), ("All files", "*.*")],
340                 defaultextension = [("Binary files (.bin)", "*.bin"), ("All files", "*.*")]
341             )
342             if (filename!=""):
343                 output_file = open(filename, "wb")
344                 pickle.dump(encrptChipertext, output_file)
345                 output_file.close()
346
347 Python 3.9.4 64-bit 0 0 0 In 1, Col 1 Spaces: 4 UTF-8 LF Python 10:24 PM 10/5/2022
```



```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py x PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

346
347
348 def SaveDecryptedFile(self):
349     # buka file di self.binary_file
350     if (self.binary_file==""):
351         self.AlertWindow("Please choose a file")
352     else:
353         # decrypt
354         key = self.key_entry.get()
355         if (len(key)==0):
356             self.AlertWindow("Please insert key")
357         else:
358             if (self.binary_file[-4:]!=".bin"):
359                 file = open(self.binary_file,"rb")
360                 readChiptext = pickle.load(file)
361                 decrypPlaintext = ExtendedDecrypt(readChiptext, key)
362                 file.close()
363                 # sama kayak atasnya
364                 filename = fd.asksaveasfilename(
365                     initialdir = "/",
366                     title = "Save file",
367                     filetypes = [("Text files (.txt)","*.txt"),("All files","*.*")],
368                     defaultextension = [("Text files (.txt)","*.txt"),("All files","*.*")]
369                 )
370                 if (filename!=""):
371                     output_file = open(filename, "w")
372                     output_file.write(decrypPlaintext)
373                     output_file.close()
374             else:
375                 self.AlertWindow("Ekstensi file yang mau didekripsi harus .bin")
```

```
File Edit Selection View Go Run Terminal Help
GUI.py - Visual Studio Code
ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py x PlayfairLib.py CommonLib.py Components.py VigenereLib.py
D:\> KeamananKomputer--Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > GUI.py > ...

374
375         self.AlertWindow("Ekstensi file yang mau didekripsi harus .bin")
376
377 def UnselectFile(self):
378     self.binary_file = ""
379     self.file_label["text"] = "File : " + self.binary_file
380
```

Componets.py

```
File Edit Selection View Go Run Terminal Help Components.py - Visual Studio Code
Get Started ExtendedLib.py AffineLib.py FullVigenerLib.py GUI.py PlayfairLib.py CommonLib.py Components.py X
D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > Components.py > ...
1 import tkinter as tk
2 import tkinter.scrolledtext as st
3
4 class TextFrame:
5     def __init__(self, title, width=50, height=5):
6         # Constructor for text frame
7         # Components : title and input field (big text)
8         # Input :
9         #     title(string) for title
10        #     width(int) and height(int) for input field dimensions
11        self.frame = tk.Frame()
12
13        self.label = tk.Label(master=self.frame, text=title)
14        self.label.pack()
15
16        self.entry = st.ScrolledText(master=self.frame, width=width, height=height)
17        self.entry.pack()
18
19 class KeyFrame:
20     def __init__(self, title, width=30):
21         # Constructor for key frame
22         # Components : title, entry, randomizer
23         # Input :
24         #     title(string) for title
25         #     width(int) for input field width
26        self.frame = tk.Frame()
27
28        self.label = tk.Label(master=self.frame, text=title)
29        self.label.pack()
30
```

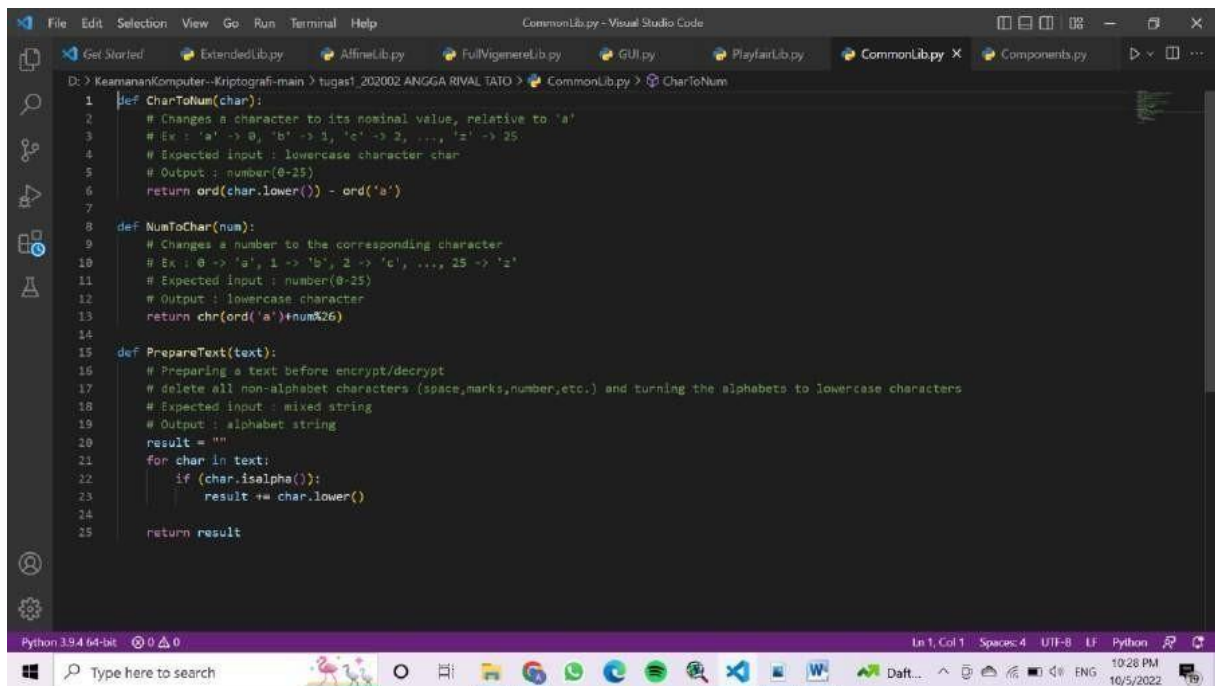
```
File Edit Selection View Go Run Terminal Help Components.py - Visual Studio Code
Get Started ExtendedLib.py AffineLib.py FullVigenerLib.py GUI.py PlayfairLib.py CommonLib.py Components.py X
D:\KeamananKomputer-Kriptografi-main > tugas1_202002 ANGGA RIVAL TATO > Components.py > ...
31 self.entry = tk.Entry(master=self.frame, width=width)
32 self.entry.pack()
33
34 self.random_label = tk.Label(master=self.frame, text="Randomizer length")
35 self.random_label.pack(padx=2, pady=2, side="left", anchor="center")
36
37 self.random_entry = tk.Entry(master=self.frame, width=4)
38 self.random_entry.pack(padx=2, pady=2, side="left", anchor="center")
39
40 self.button = tk.Button(master=self.frame, width=10, text="Randomize")
41 self.button.pack(padx=2, pady=2, side="left", anchor="center")
42
43 class AffineKeyFrame:
44     def __init__(self, width=10):
45         # Constructor for affine key frame
46         # Components : two titles and two input fields (one line)
47         # Input :
48         #     width(int) for input field width
49        self.frame = tk.Frame()
50
51        self.multiple_label = tk.Label(master=self.frame, text="Multiple (m)")
52        self.multiple_label.pack(padx=2, pady=2, side="left")
53
54        self.multiple_entry = tk.Entry(master=self.frame, width=width)
55        self.multiple_entry.pack(padx=2, pady=2, side="left")
56
57        self.offset_label = tk.Label(master=self.frame, text="Offset (b)")
58        self.offset_label.pack(padx=2, pady=2, side="left")
59
60        self.offset_entry = tk.Entry(master=self.frame, width=width)
```

```
File Edit Selection View Go Run Terminal Help Components.py - Visual Studio Code
Get Started ExtendedLib.py AffineLib.py FullVigenereLib.py GUI.py PlayfairLib.py CommonLib.py Components.py x
D:\KeamananKomputer-Kriptografi-main> tugas1_202002 ANGGA RIVAL TATO > Components.py > ...

56
57 self.offset_label = tk.Label(master=self.frame,text="Offset (b)")
58 self.offset_label.pack(padx=2,pady=2,side="left")
59
60 self.offset_entry = tk.Entry(master=self.frame,width=width)
61 self.offset_entry.pack(padx=2,pady=2,side="left")
62
63 class ButtonListFrame:
64     def __init__(self,title,labels,width=20):
65         # Constructor for list of buttons frame
66         # Components : title and button list
67         # Input :
68         #     title(string) for title
69         #     labels(list of strings) for button label
70         #     width(int) for button width
71         self.frame = tk.Frame()
72
73         self.label = tk.Label(master=self.frame,text=title)
74         self.label.pack()
75
76         self.button_list = []
77         for label in labels:
78             new_button = tk.Button(master=self.frame,text=label,width=width)
79             new_button.pack(padx=2,pady=2)
80             self.button_list.append(new_button)
```

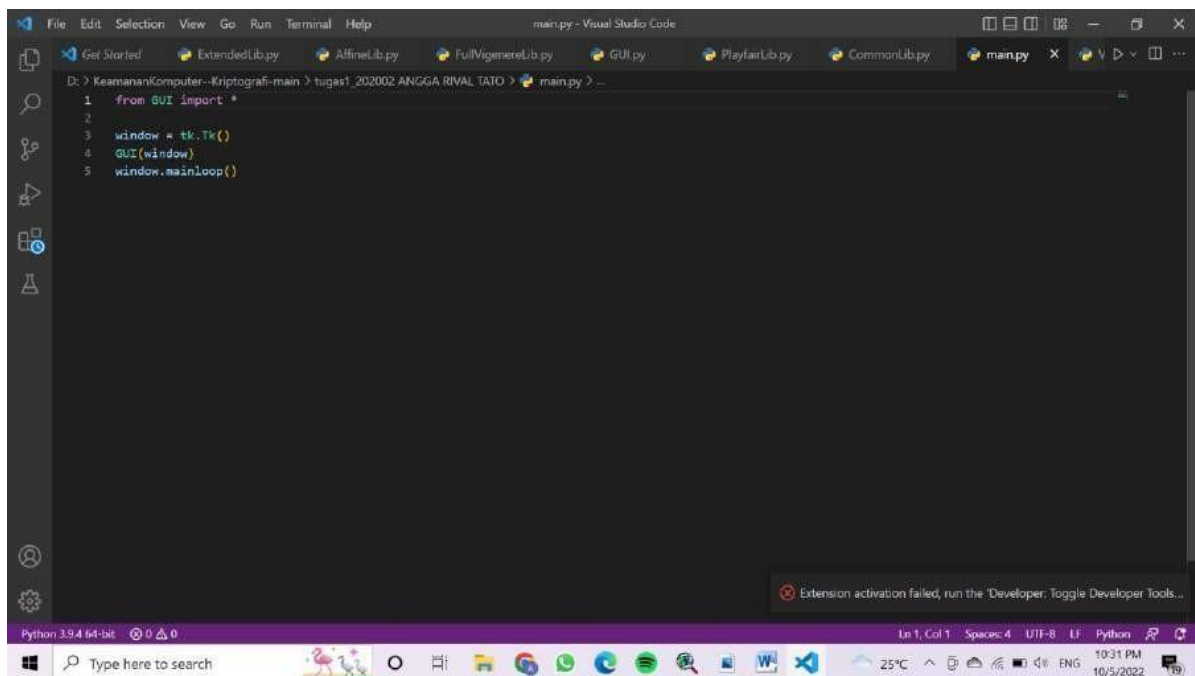
Python 3.9.4 64-bit 0 0 0 Lin 1, Col 1 Spaces 4 UTF-8 LF Python 10:27 PM 10/5/2022

CommonLib.py



```
1 def CharToNum(char):
2     # Changes a character to its nominal value, relative to 'a'
3     # Ex : 'a' -> 0, 'b' -> 1, 'c' -> 2, ..., 'z' -> 25
4     # Expected input : lowercase character char
5     # Output : number(0-25)
6     return ord(char.lower()) - ord('a')
7
8 def NumToChar(num):
9     # Changes a number to the corresponding character
10    # Ex : 0 -> 'a', 1 -> 'b', 2 -> 'c', ..., 25 -> 'z'
11    # Expected input : number(0-25)
12    # Output : lowercase character
13    return chr(ord('a') + num%26)
14
15 def PrepareText(text):
16    # Preparing a text before encrypt/decrypt
17    # delete all non-alphabet characters (space,marks,number,etc.) and turning the alphabets to lowercase characters
18    # Expected input : mixed string
19    # Output : alphabet string
20    result = ""
21    for char in text:
22        if (char.isalpha()):
23            result += char.lower()
24
25    return result
```

main.py

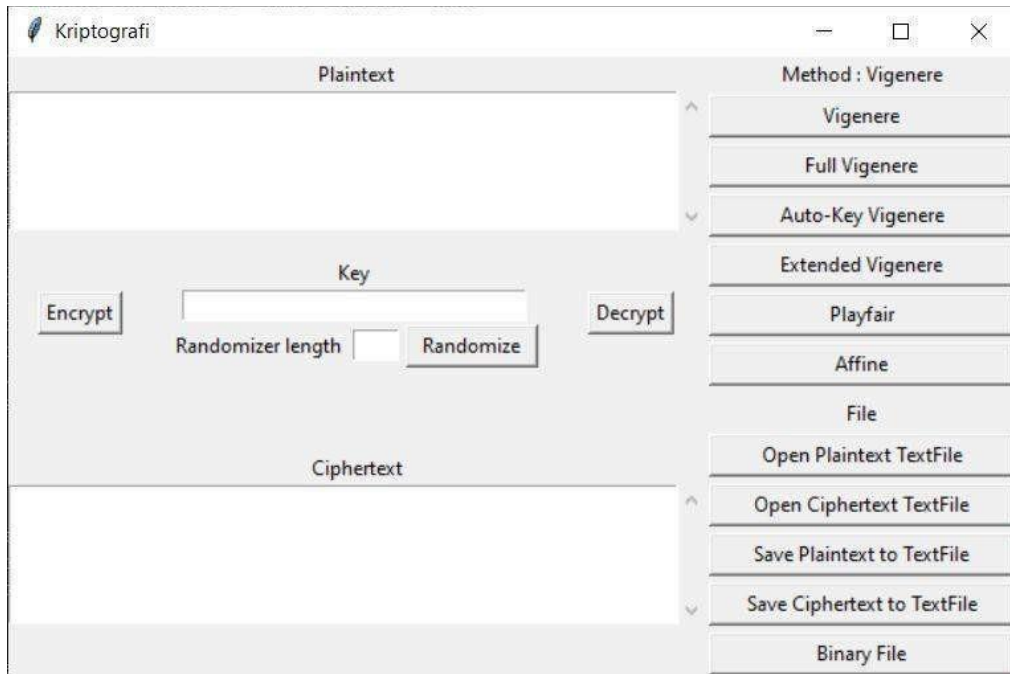


```
1 from GUI import *
2
3 window = tk.Tk()
4 GUI(window)
5 window.mainloop()
```

2. Tampilan antarmuka program (print screen)

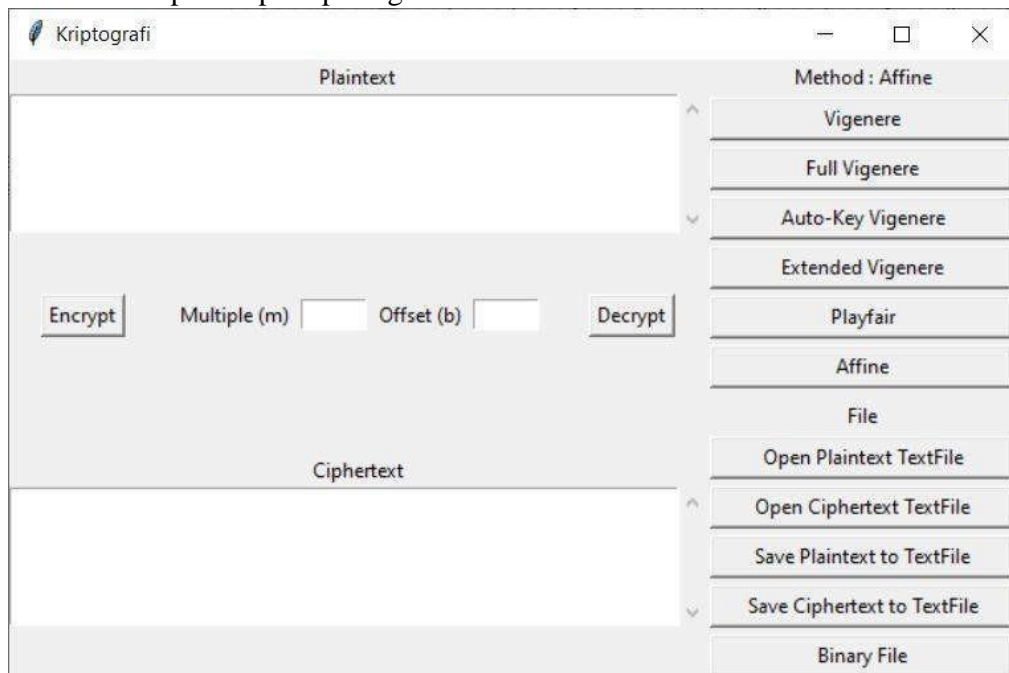
Antarmuka program dibuat dengan menggunakan tkinter pada dengan komponen pada Components.py dan digabungkan pada GUI.py.

Berikut adalah tampilan dari antarmuka program yang dibuat.



Gambar 1. Tampilan Antarmuka Program

Untuk metode Affine Cipher yang tidak menggunakan key, tampilan dari KeyFrame akan berubah ketika dipilih metode Affine Cipher. Pada tampilan ini, diminta input parameter m dan b untuk melakukan Affine Cipher seperti pada gambar berikut.



Gambar 2. Tampilan Antarmuka Program pada Affine Cipher

Untuk handling file binary, ketika button “Binary File” di pojok kanan bawah ditekan akan muncul window sebagai berikut.

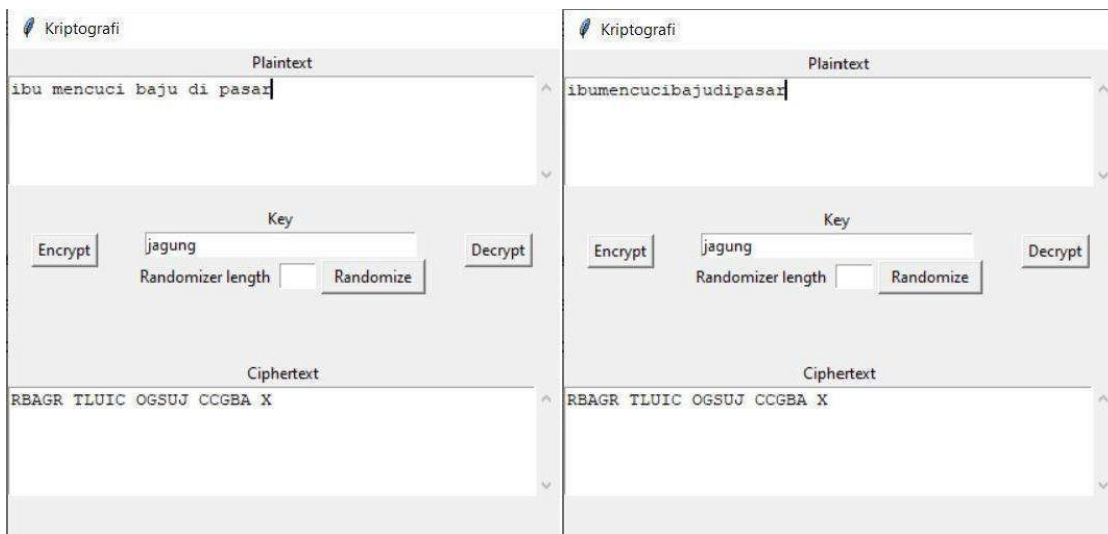


Gambar 3. Tampilan Antarmuka Program untuk Binary File

3. Contoh plainteks dan cipherteks (text, gambar, file database, audio, video)

Pengujian Vigenere Cipher

Pada pengujian pertama, digunakan plaintext “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Vigenere Cipher kemudian didekripsi lagi.



Gambar 4. Pengujian Vigenere Cipher 1

Selanjutnya, digunakan input plaintext “bunga bunga bambu terbang” dengan key “apel” yang memberikan hasil sebagai berikut.

Kriptografi	
Plaintext	
bunga bunga bambu terbang	
Key	
Encrypt	ape
Randomizer length <input type="text"/> Randomize	
Decrypt	
Ciphertext	
BJRRA QYYGP FLMQY EEGFL NV	

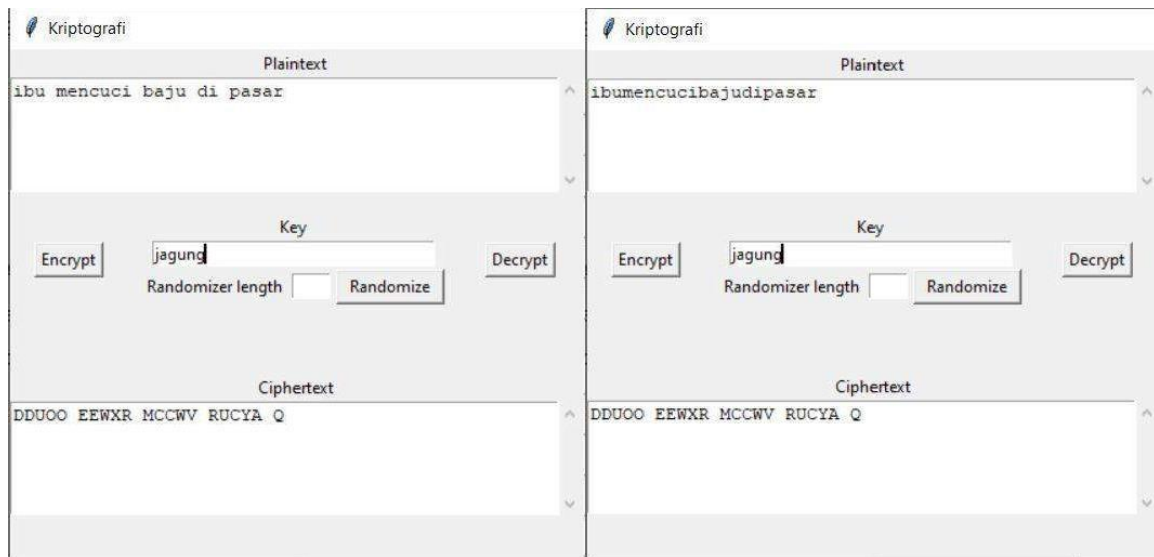
Kriptografi	
Plaintext	
bungabungabambuterbang	
Key	
Encrypt	ape
Randomizer length <input type="text"/> Randomize	
Decrypt	
Ciphertext	
BJRRA QYYGP FLMQY EEGFL NV	

Gambar 5. Pengujian Vigenere Cipher 2

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-dekripsi yang terjadi berjalan dengan baik, hanya saja terjadi kehilangan informasi pada teks selain alfabet (seperti spasi). Hal ini sudah sesuai dengan spesifikasi yang diharapkan.

Pengujian Full Vigenere Cipher

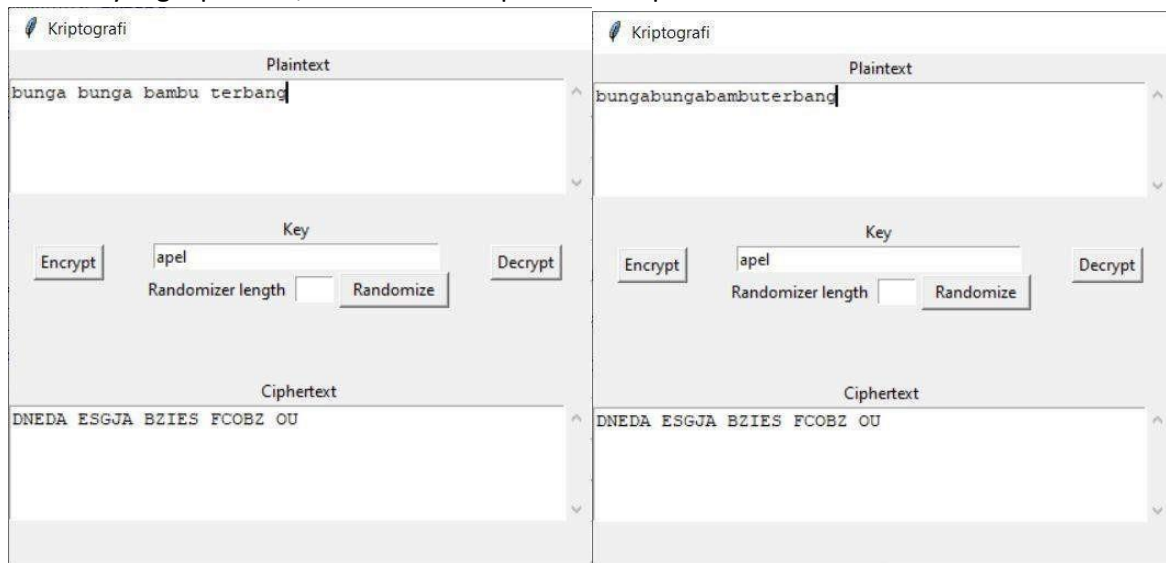
Pada pengujian pertama, digunakan plaintext “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Full Vigenere Cipher kemudian didekripsi lagi.



Gambar 6. Pengujian Full Vigenere Cipher 1

Selanjutnya, digunakan input plaintext “bunga bunga bambu terbang” dengan key “apel” yang memberikan hasil sebagai berikut.

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-

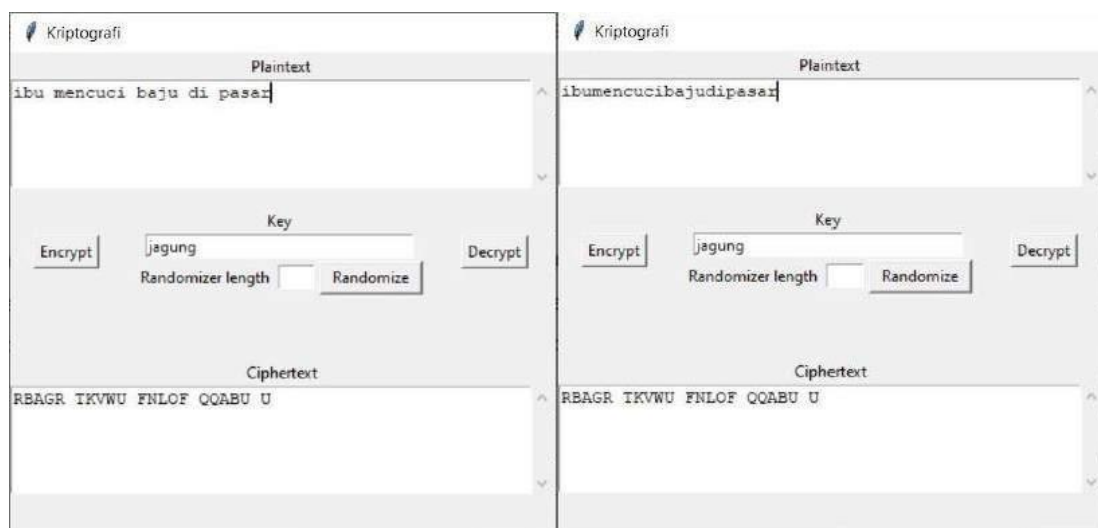


Gambar 7. Pengujian Full Vigenere Cipher 2

dekripsi yang terjadi berjalan dengan baik, hanya saja terjadi kehilangan informasi pada teks selain alfabet (seperti spasi). Hal ini sudah sesuai dengan spesifikasi yang diharapkan.

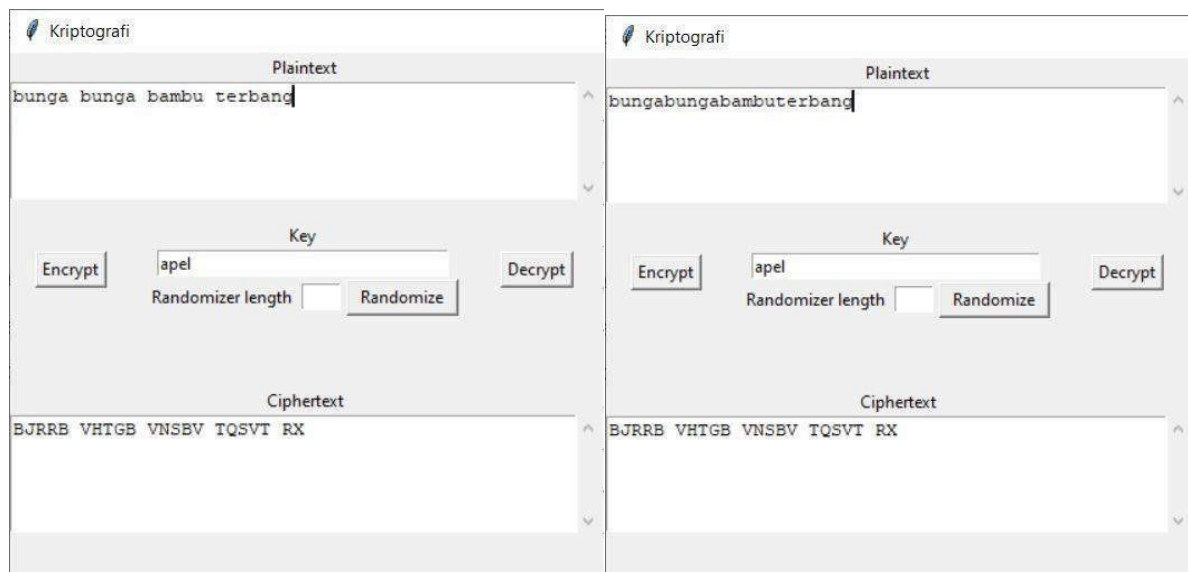
Pengujian Auto Key Vigenere Cipher

Pada pengujian pertama, digunakan plaintext “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Auto Key Vigenere Cipher kemudian didekripsi lagi.



Gambar 8. Pengujian Auto Key Vigenere Cipher 1

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-dekripsi yang terjadi berjalan dengan Selanjutnya, digunakan input plaintext “bunga bunga bambu terbang” dengan key “apel” yang memberikan hasil sebagai berikut.

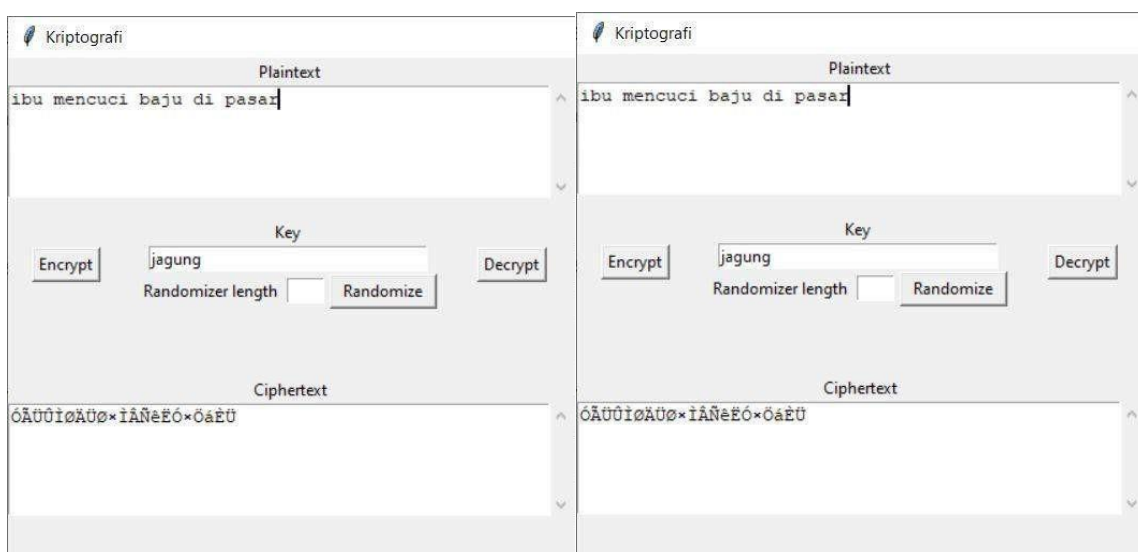


Gambar 10. Pengujian Auto Key Vigenere Cipher 2

baik, hanya saja terjadi kehilangan informasi pada teks selain alfabet (seperti spasi). Hal ini sudah sesuai dengan spesifikasi yang diharapkan.

Pengujian Extended Vigenere Cipher

Untuk pengujian Extended Vigenere Cipher pertama dilakukan dengan melakukan input pada GUI program yang dibuat, dengan mengisi bagian plaintext dan key kemudian melakukan enkripsi-dekripsi seperti biasa. Digunakan plaintext “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Extended Vigenere Cipher kemudian didekripsi lagi.



Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-

Gambar 10. Pengujian Extended Vigenere Cipher 1

Selanjutnya, dilakukan pengujian pada file binary. Digunakan plaintext yang sama yaitu “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Extended Vigenere Cipher kemudian didekripsi lagi.



Gambar 11. Pengujian Extended Vigenere Cipher 2

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-dekripsi yang terjadi berjalan cukup baik, masih terdapat sedikit masalah pada enkripsi-dekripsi biasa dimana terdapat 32 karakter ASCII hasil enkripsi yang tidak dapat ditampilkan (*not printable*) sehingga memengaruhi proses

dekripsi yang terjadi berjalan dengan

dekripsi (rentan salah). Keunggulan cipher ini adalah proses enkripsi tidak menghilangkan karakter selain abjad, sehingga karakter seperti spasi dan tanda baca dapat terbaca (dekripsi) dengan baik.

Pengujian Playfair Cipher

Pada pengujian pertama, digunakan plaintext “ibu mencuci baju di pasar” dengan key “jagung” yang dienkripsi dengan Playfair Cipher kemudian didekripsi lagi.

The image shows two side-by-side screenshots of a web application titled "Kriptografi". The application has three main sections: Plaintext, Key, and Ciphertext. In the first screenshot, the Plaintext field contains "ibu mencuci baju di pasar", the Key field contains "jagung", and the Ciphertext field displays "OANLF UEAI P AGLAC KVC PN XU". In the second screenshot, the Plaintext field contains "ibumencucibaiudipasar" (spaces removed), the Key field still contains "jagung", and the Ciphertext field displays "OANLF UEAI P AGLAC KVC PN XU". Both screenshots include "Encrypt" and "Decrypt" buttons, as well as a "Randomizer length" input and a "Randomize" button.

Gambar 12. Pengujian Playfair Cipher 1

Selanjutnya, digunakan input plaintext “bunga bunga bambu terbang” dengan key “apel” yang memberikan hasil sebagai berikut.

The image shows two side-by-side screenshots of a web application titled "Kriptografi". The application has three main sections: Plaintext, Key, and Ciphertext. In the first screenshot, the Plaintext field contains "bunga bunga bambu terbang", the Key field contains "apel", and the Ciphertext field displays "H2TNP ATOCL APOEQ UPSAP TN". In the second screenshot, the Plaintext field contains "bungabungabambuterbang" (spaces removed), the Key field still contains "apel", and the Ciphertext field displays "H2TNP ATOCL APOEQ UPSAP TN". Both screenshots include "Encrypt" and "Decrypt" buttons, as well as a "Randomizer length" input and a "Randomize" button.

Gambar 13. Pengujian Playfair Cipher 2

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-baik, hanya saja terjadi kehilangan informasi pada teks selain alfabet (seperti spasi), dan kehilangan informasi mengenai huruf 'j' pada plainteks (diganti dengan huruf 'i'). Hal ini sudah sesuai dengan spesifikasi yang diharapkan.

Pengujian Affine Cipher

Pada pengujian pertama, digunakan plaintext “ibu mencuci baju di pasar” dengan $m = 3$ dan $b = 10$ yang dienkripsi dengan Affine Cipher kemudian didekripsi lagi.

The image shows two side-by-side screenshots of a web application titled "Kriptografi". Each screenshot displays the encryption and decryption process for the Affine Cipher.

Left Screenshot:

- Plaintext:** ibu mencuci baju di pasar
- Encryption Settings:** Multiple (m) = 3, Offset (b) = 10
- Ciphertext:** INSUW XQSQI NKLST IDKMK J

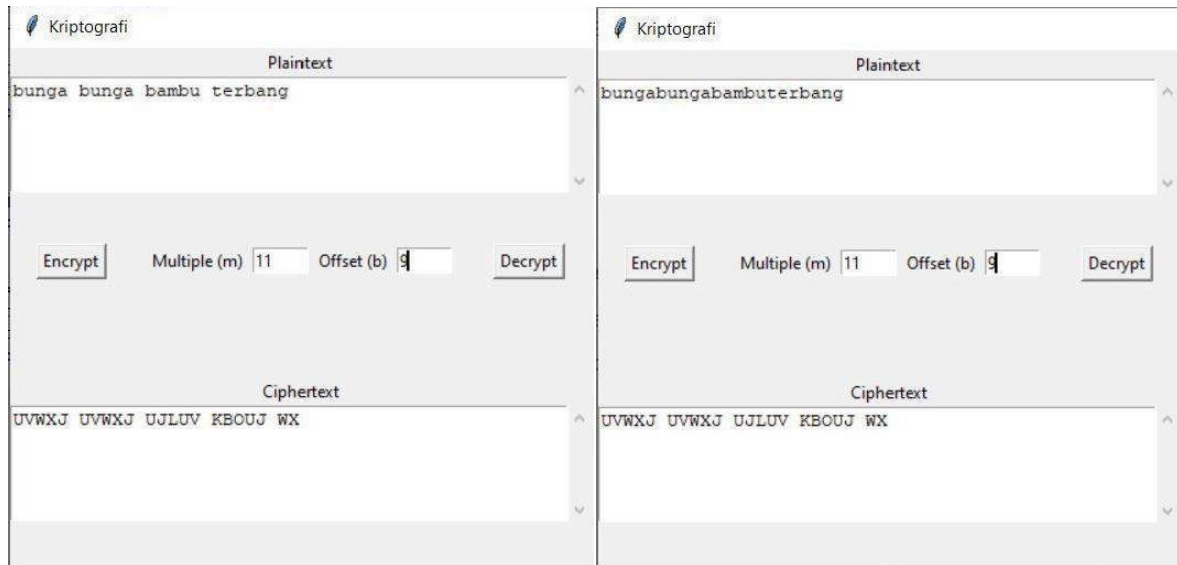
Right Screenshot:

- Plaintext:** ibumencucibajudipasar (Note: spaces are removed)
- Encryption Settings:** Multiple (m) = 3, Offset (b) = 10
- Ciphertext:** INSUW XQSQI NKLST IDKMK J

Gambar 14. Pengujian Affine Cipher 1

Selanjutnya, digunakan input plaintext “bunga bunga bambu terbang” dengan $m = 11$ dan $b = 9$ yang memberikan hasil sebagai berikut.

dekripsi yang terjadi berjalan dengan



Gambar 15. Pengujian Affine Cipher 2

Dari hasil yang diperoleh, terlihat bahwa proses enkripsi-dekripsi yang terjadi berjalan dengan baik, hanya saja terjadi kehilangan informasi pada teks selain alfabet (seperti spasi). Hal ini sudah sesuai dengan spesifikasi yang diharapkan.

4. Link Ke GitHub

<https://github.com/Najwar09/Tugas1-202018-Muh.-Najwar-Ramadhan-5TKKO-E>

5. Keterangan Keberhasilan Program

NO	SPEK	Berhasil (V)	Kurang berhasil (V)	Keterangan
1	Vigenere Cipher			
2	Enigma Cipher			
3	Auto-Key Vigenere Cipher			
4	Extended Vigenere Cipher			Terdapat 32 karakter ASCII yang tidak dapat ditampilkan, membuat proses dekripsi rentan salah (dapat ditingkatkan melalui penyimpanan enkripsi dengan <i>file of bytes</i>), untuk file binary ekstensi hanya .bin
5	Playfair Cipher			
6	Bonus: Affine Cipher			