PRAK 4 One-Time Pad (OTP) Cipher



Oleh:

Fernanda Arya Putra(22230008)

PROGRAM STUDI SISTEM INFORMASI FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS RESPATI YOGYAKARTA

1. ENCRYPT

```
(i) localhost 8888/notebooks/Untitled4.ipynb?
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       0
Jupyter Untitled4 Last Checkpoint: 27 days ago
                                                 File Edit View Run Kernel Settings Help
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Trusted
                                               1 + % □ □ > ■ C >> Code ∨
                                                                                                                                                                                                                                                                                                                                                                          JupyterLab □ # Python [conda env:base] * ○ ■ ■
                                                             [5]: import random
                                                                             import random
import string
def generate_key(length):
    """Generate a random key of uppercase letters."""
    return ''.join(random.choice(string.ascii_uppercase) for _ in range(length))
def text_to_numbers(text):
    """Convert A-Z to 0-25""
    return [ord(c) - ord('A') for c in text]
def numbers to text(numbers):
                                                                              def numbers, to_text(numbers):
    """convert 0-25 to A-2""
    return ''.join(chr(n + ord('A')) for n in numbers)
def otp_encrypt(plaintext, key):
                                                                             def otp_encrypt(plaintext, key):
    plain_nums = text_to_numbers(plaintext)
    key_nums = text_to_numbers(key)
    cipher_nums = (p + k) % 26 for p, k in zip(plain_nums, key_nums)]
    return numbers_to_text(cipher_nums)

def otp_decrypt(ciphertext, key):
    cipher_nums = text_to_numbers(ciphertext)
    key_nums = text_to_numbers(key)
    nlain_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums, key_nums = (fc - k + 2c) % 26 for c, k in zinfcipher_nums.
                                                                              plain_nums = [(c - k + 26) % 26 for c, k in zip(cipher_nums, key_nums)]
return numbers to_text(plain_nums)

def clean_input(text):
                                                                                                                                                ove non-letter characters.
                                                                                        return ''.join(filter(str.isalpha, text.upper()))
                                                                              if __name__ == "__main__":
    print("=== One-Time Pad Cipher ===")
    mode = input("Mode (encrypt/decrypt): ").strip().lower()
```

```
C (i) localhost:8888/notebooks/Untitled4.ipynb?
                                                                                                                                                                                                                                                                                                     Q ☆ ☆ 🚇 ... 🦚
                                                                                                                                                                                                                                                                                                                                               2
Jupyter Untitled4 Last Checkpoint: 27 days ago
                                   File Edit View Run Kernel Settings Help
                                                                                                                                                                                                                                                                                                                                            Trusted
                                 B + % □ □ > ■ C >> Code
                                                                                                                                                                                                                                                               JupyterLab □ ■ Python [conda env:base] * ○ ■ ■
                                                       def clean_input(text):
                                                              """Uppercase and remove non-letter characters."""
return ''.join(filter(str.isalpha, text.upper()))
                                                       if __name__ == "__main__":
    print("=== One-Time Pad Cipher ===")
    mode = input("Mode (encrypt/decrypt): ").strip().lower()
                                                            if mode == "encrypt":
    plaintext = clean_input(input("Enter plaintext: "))
    key = generate_key(lein(plaintext))
    ciphertext = otp_encrypt(plaintext, key)
    print("N---- Encryption Result ---")
    print("Plaintext :*, plaintext)
    print("Key :*, key)
    print("Ciphertext:", ciphertext)
                                                                     v mous == decrypt:
ciphertext = clean_input(input("Enter ciphertext: "))
key = clean_input(input("Enter key (same length): "))
if len(ciphertext) != len(key):
    print('Error: Key length must match ciphertext length.")
                                                                            plaintext = otp_decrypt(ciphertext, key)
print("\n-- Decryption Result ---")
print("ciphertext:", ciphertext)
print("key :", key)
print("Plaintext :", plaintext)
                                                                     print("Invalid mode. Use 'encrypt' or 'decrypt'.")
```

OUTPUT:

```
=== One-Time Pad Cipher ===

Mode (encrypt/decrypt): encrypt
Enter plaintext: KEAMANANI SISTEM INFORMASI JARINGAN
--- Encryption Result ---
Plaintext : KEAMANANISITEMINFORMASIJARINGAN
Key : BMTZLRYPOHPXLORRRYQRWITWZZHJCPY
Ciphertext: LATILEYCGOPHYNAZENAHOM.NFZQPWIPL
```

2. DESCRYPT

```
Q 🖒 🗯 🚇
                             (i) localhost:8888/notebooks/Untitled4.ipynb?
                                     Jupyter Untitled4 Last Checkpoint: 27 days ago
File Edit View Run Kernel Settings Help
                                                                                                                                                                                                                                                                                                                                                                Trusted
                                   1 + % □ □ 1 • ■ C → Code
                                                                                                                                                                                                                                                                               JupyterLab □ # Python [conda env:base] * ○ ■ ■
                                             [12]: import random
                                                                                                                                                                                                                                                                                                                   ★ 10 个 ↓ 占 〒 1
                                                          import random
import string
def generate_key(length):
    """Generate a random key of uppercase letters."""
                                                          return ''.join(random.choice(string.ascii_uppercase) for _ in range(length))

def text_to_numbers(text):
    """convert A-Z to 0-25"""
    return [ord(c) - ord('A') for c in text]
                                                         return [ord(c) - ord('A') for c in text]

def numbers_to_text(numbers):
    ""Convert 0-25 to A-2""
    return ".join(chr(n + ord('A')) for n in numbers)

def otp_encrypt(plaintext, key):
    plain_nums = text_to_numbers(plaintext)
    key_nums = text_to_numbers(key)
    cipher_nums = [(p+k) % 26 for p, k in zip(plain_nums, key_nums)]
    return numbers_to_text(cipher_nums)

def otp_decrypt(ciphertext, key):
    cipher_nums = text_to_numbers(ciphertext)
                                                                 orp_decrypt(cipnertext, key):
cipher_nums = text_to_numbers(ciphertext)
key_nums = text_to_numbers(key)
plain_nums = [(c - k + 26) % 26 for c, k in zip(cipher_nums, key_nums)]
return numbers_to_text(plain_nums)
                                                          def clean input(text):
                                                                 """Uppercase and remove non-letter characters."""
return ''.join(filter(str.isalpha, text.upper()))
                                                          if __name__ == "__main__":
    print("=== One-Time Pad Cipher ===")
    mode = input("Mode (encrypt/decrypt): ").strip().lower()
```

```
C (i) localhost 8888/notebooks/Untitled4.ipynb?
                                                                                                                                                                                                                                        ☆ ☆ 🚇 … 🥠
                                                                                                                                                                                                                                                                  2
                 Jupyter Untitled4 Last Checkpoint: 27 days ago
                 File Edit View Run Kernel Settings Help
                                                                                                                                                                                                                                                               Trusted
                def clean_input(text):
                                                                                                                                                                                                  JupyterLab □ ■ Python [conda env:base] * ○ ■ ■
                                             Oppercase and remove non-letter characters.""
                                       return ''.join(filter(str.isalpha, text.upper()))
                                if __name__ == "_main__":
    print("=== One-Time Pad Cipher ===")
    mode = input("Mode (encrypt/decrypt): ").strip().lower()
                                             plaintext = clean input(input("Enter plaintext: "))
                                            plaintext = clean_input(input("Enter plai
key = generate_key(len(plaintext))
ciphertext = otp_encrypt(plaintext, key)
print("\n-- Encryption Result ---")
print("Plaintext :", plaintext)
print("Rey :", key)
print("Ciphertext:", ciphertext)
                                      elif mode == "decrypt":
    ciphertext = clean_input(input("Enter ciphertext: "))
    key = clean_input(input("Enter key (same length): "))
    if len(ciphertext) != len(key):
                                            print("Error: Key length must match ciphertext length.")
else:
                                                  plaintext = otp_decrypt(ciphertext, key)
                                                   print("\n-- Decryption Result --
print("Ciphertext:", ciphertext)
print("Key :", key)
print("Plaintext :", plaintext)
                                       else:
                                             print("Invalid mode. Use 'encrypt' or 'decrypt'.")
```

OUTPUT:

```
=== One-Time Pad Cipher ===

Mode (encrypt/decrypt): decrypt
Enter ciphertext: MANCING SERU
Enter key (same length): ZXGYHJ JAYY
--- Decryption Result ---
Ciphertext: MANCINGSERU
Key : ZXGYHJ JAYY
Plaintext : MOHEROXJETU
```