# Aplikasi Enkripsi dan Dekripsi AES-256 dengan GUI Sederhana  
# Dibuat menggunakan Python 3  
# Diperlukan library: cryptography (instal dengan: pip install cryptography)  
# GUI menggunakan Tkinter (sudah built-in di Python)  
  
import tkinter as tk  
from tkinter import messagebox  
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes  
from cryptography.hazmat.primitives import padding  
from cryptography.hazmat.backends import default\_backend  
import os  
import base64  
  
def generate\_key():  
 """Generate a random 256-bit key."""  
 return os.urandom(32)  
  
def pad(data):  
 """Pad the data to be multiple of block size."""  
 padder = padding.PKCS7(algorithms.AES.block\_size).padder()  
 padded\_data = padder.update(data) + padder.finalize()  
 return padded\_data  
  
def unpad(padded\_data):  
 """Unpad the data."""  
 unpadder = padding.PKCS7(algorithms.AES.block\_size).unpadder()  
 data = unpadder.update(padded\_data) + unpadder.finalize()  
 return data  
  
def encrypt(text, key):  
 """Encrypt the text using AES-256-CBC."""  
 try:  
 iv = os.urandom(16) # Initialization Vector  
 cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default\_backend())  
 encryptor = cipher.encryptor()  
 padded\_text = pad(text.encode('utf-8'))  
 ct = encryptor.update(padded\_text) + encryptor.finalize()  
 # Combine IV and ciphertext, then base64 encode for easy display  
 return base64.urlsafe\_b64encode(iv + ct).decode('utf-8')  
 except Exception as e:  
 return str(e)  
  
def decrypt(ciphertext, key):  
 """Decrypt the ciphertext using AES-256-CBC."""  
 try:  
 data = base64.urlsafe\_b64decode(ciphertext)  
 iv = data[:16]  
 ct = data[16:]  
 cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default\_backend())  
 decryptor = cipher.decryptor()  
 padded\_text = decryptor.update(ct) + decryptor.finalize()  
 return unpad(padded\_text).decode('utf-8')  
 except Exception as e:  
 return str(e)  
  
# GUI Setup  
root = tk.Tk()  
root.title("AES-256 Encrypt/Decrypt")  
root.geometry("500x400")  
  
# Key Section  
key\_label = tk.Label(root, text="Key (32 bytes, base64 encoded):")  
key\_label.pack(pady=5)  
key\_entry = tk.Entry(root, width=50)  
key\_entry.pack()  
  
def generate\_key\_btn():  
 key = generate\_key()  
 key\_b64 = base64.urlsafe\_b64encode(key).decode('utf-8')  
 key\_entry.delete(0, tk.END)  
 key\_entry.insert(0, key\_b64)  
 messagebox.showinfo("Key Generated", "Key telah digenerate dan dimasukkan ke field.")  
  
gen\_key\_button = tk.Button(root, text="Generate Key", command=generate\_key\_btn)  
gen\_key\_button.pack(pady=5)  
  
# Input Text  
input\_label = tk.Label(root, text="Input Text:")  
input\_label.pack(pady=5)  
input\_text = tk.Text(root, height=5, width=50)  
input\_text.pack()  
  
# Output Text  
output\_label = tk.Label(root, text="Output:")  
output\_label.pack(pady=5)  
output\_text = tk.Text(root, height=5, width=50)  
output\_text.pack()  
  
def encrypt\_btn():  
 text = input\_text.get("1.0", tk.END).strip()  
 key\_b64 = key\_entry.get().strip()  
 if not text or not key\_b64:  
 messagebox.showerror("Error", "Masukkan text dan key!")  
 return  
 try:  
 key = base64.urlsafe\_b64decode(key\_b64)  
 if len(key) != 32:  
 raise ValueError("Key harus 32 bytes!")  
 encrypted = encrypt(text, key)  
 output\_text.delete("1.0", tk.END)  
 output\_text.insert(tk.END, encrypted)  
 except Exception as e:  
 messagebox.showerror("Error", str(e))  
  
def decrypt\_btn():  
 ciphertext = input\_text.get("1.0", tk.END).strip()  
 key\_b64 = key\_entry.get().strip()  
 if not ciphertext or not key\_b64:  
 messagebox.showerror("Error", "Masukkan ciphertext dan key!")  
 return  
 try:  
 key = base64.urlsafe\_b64decode(key\_b64)  
 if len(key) != 32:  
 raise ValueError("Key harus 32 bytes!")  
 decrypted = decrypt(ciphertext, key)  
 output\_text.delete("1.0", tk.END)  
 output\_text.insert(tk.END, decrypted)  
 except Exception as e:  
 messagebox.showerror("Error", str(e))  
  
encrypt\_button = tk.Button(root, text="Encrypt", command=encrypt\_btn)  
encrypt\_button.pack(side=tk.LEFT, padx=10, pady=10)  
  
decrypt\_button = tk.Button(root, text="Decrypt", command=decrypt\_btn)  
decrypt\_button.pack(side=tk.RIGHT, padx=10, pady=10)  
  
root.mainloop()

### **Cara Menjalankan:**

1. Instal Python 3 di Windows (unduh dari python.org).
2. Instal library cryptography: Buka Command Prompt, jalankan pip install cryptography.
3. Simpan kode di atas ke file, misalnya aes\_app.py.
4. Jalankan dengan double-click atau via Command Prompt: python aes\_app.py.

Aplikasi ini menggunakan AES-256 dalam mode CBC dengan padding PKCS7. Key digenerate secara acak (32 bytes) dan ditampilkan dalam base64 untuk kemudahan. Masukkan teks ke input, generate atau masukkan key, lalu klik Encrypt atau Decrypt. Output akan muncul di bawah. Pastikan gunakan key yang sama untuk decrypt!