

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada saat ini teknologi informasi dan komunikasi semakin berkembang dengan pesat, seiring berkembangnya teknologi, berbagai perangkat mulai bergantung pada jaringan internet. Jaringan internet merupakan jaringan komunikasi yang menghubungkan berbagai media elektronik secara global dan terbuka. Jaringan internet memiliki berbagai macam layanan yang dapat digunakan oleh pengguna seperti E-mail, dan chatting. Adapun dampak positif dari jaringan internet yaitu dapat bertukar informasi dengan cepat secara realtime, selain itu memudahkan dalam berkomunikasi dimana saja dan kapan saja. Tidak hanya dampak positif, jaringan internet juga memiliki dampak negatif yaitu munculnya suatu tindak kejahatan di dunia maya atau disebut cybercrime. Cybercrime muncul disebabkan oleh penggunaan jaringan internet yang tidak terbatas sehingga memberikan peluang kepada oknum untuk melakukan tindak kejahatan seperti mengambil informasi untuk kepentingan pribadi atau golongan tertentu.

Pada aplikasi yang dirancang oleh Nur Taufik Hidayat pada tahun 2018, aplikasi tersebut menggunakan metode AES, namun hanya dapat mengenkripsi 4 jenis file yaitu file docx, pptx, xlsx dan pdf.

Berdasarkan uraian tersebut penulis ingin membuat aplikasi yang digunakan untuk pengamanan suatu informasi pada semua jenis file dengan menyembunyikan byte asli pada file, untuk menyembunyikan byte asli digunakan enkripsi metode AES-256 (Advanced Encryption Standard) untuk mengubah setiap block code pada memori komputer. Pembuatan aplikasi ini menggunakan bahasa pemrograman Python yang diketikkan pada editor Sublime Text.

## **BAB 2**

### **TEORI**

#### **2.1 Kriptografi AES (Advanced Encryption Standar)**

AES (Advanced Encryption Standard) adalah teknik enkripsi yang dijadikan standar FIPS oleh NIST tahun 2001. AES dimaksudkan akan, secara bertahap, menggantikan DES sebagai standard enkripsi di Amerika Serikat untuk abad ke 21. DES sebagai standar FIPS telah dicabut, Mei 2005 (Kromodimoeljo, 2009).

Pada Kriptografi AES terdapat 3 jenis yaitu AES 128-bit, AES 192-bit dan AES 256-bit. Pengelompokkan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka di belakang kala AES menggambarkan Panjang kunci yang digunakan pada tiap-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round. (Ainul. 2017).

#### **2.2 Proses Dekripsi AES**

Transformasi cipher dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES (Yuniati, Indriyanta, dan Rachmat, 2009). Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows

## **BAB 3**

### **ANALISI DAN PERANCANGAN**

#### **3.1 Kebutuhan Perangkat Keras**

Pada tahap ini melakukan analisa dari kebutuhan perangkat keras yang digunakan dalam proses pembuatan program yaitu satu buah laptop dengan spesifikasi antara lain:

- Laptop-1055CEQO
- Intel Core i5-1135g7 @ 2.40Hz
- Memori 8GB RAM

#### **Kebutuhan Perangkat Lunak**

Pada tahap ini melakukan analisa dari kebutuhan perangkat lunak yang digunakan dalam proses pembuatan program. Perangkat lunak yang digunakan antara lain:

- Sistem Operasi Windows 11
- Sublime Text
- Python 3

## **BAB 4**

### **IMPLEMENTASI DAN UJI COBA**

#### **4.1. Pemrograman**

Berdasarkan rancangan yang telah dibuat sebelumnya, rancangan tersebut di implementasikan ke dalam kode program. Pembuatan program menggunakan perangkat lunak Sublime Text dan Python sebagai bahasa pemrograman.

##### **4.1.1. Pembuatan Kelas Aplikasi**

Sebelum membuat aplikasi enkripsi pada python, kodingan berikut untuk mengimport library yang akan digunakan seperti os, io, sys, binaseii, pyAesCrypt, tkinter dan PIL.

```
import os import io import sys
import binascii
import pyAesCrypt
from tkinter import *
from tkinter import messagebox
from tkinter import filedialog
from PIL import Image,
```

Selanjutnya terdapat sebuah class baru untuk aplikasi enkripsi dengan nama AES\_encrp dengan atribut variabel lokasi, nama, text\_in, cipher\_text, sandi dan pilihan. Variabel tersebut akan digunakan untuk proses enkripsi dan dekripsi file maupun teks.

```
class AES_encrp:
    lokasi=""
    nama="" text_in="" cipher_text="" sandi="" pilihan=""
```

##### **4.1.2. Pembuatan Menu Utama**

Menu utama merupakan tampilan awal yang muncul ketika aplikasi dijalankan. Terdapat tiga buah button yang memiliki fungsi berbeda-beda. Button 1 sebagai button file yang berfungsi untuk menampilkan halaman menu File jika ingin melakukan proses enkripsi dan dekripsi berjenis file, button2 sebagai button teks yang berfungsi untuk menampilkan halaman menu Teks jika ingin melakukan proses enkripsi dan dekripsi berjenis teks, sedangkan butlon3 sebagai button bantuan yang berfungsi untuk menampilkan halaman bantuan mengenai cara penggunaan aplikasi.

Kodingan berikut ini untuk membuat fungsi yang akan digunakan sebagai fungsi main

dari aplikasi ini. Buat obyek Tk() yang berarti tkinter dengan variabel window. Perintah geometry untuk mengatur ukuran dari window yang akan ditampilkan, selanjutnya perintah title untuk memberi nama window yang akan dibuat pada window bar, lalu label 1 untuk membuat judul aplikasi dan dengan pengaturan teks berwarna biru dan bertipe arial, teks ini akan diletakkan pada koordinat x=58 dan y=100 dengan menggunakan perintah place.

```
def main():  
    window=Tk()  
    window.geometry("500x400")  
    window.resizable(False, False)  
    window.title("Welcome")
```

Selanjutnya kodingan berikut ini untuk menambahkan background pada menu utama dengan menggunakan file untuk background yang bernama "background.jpg"

```
image=Image.openf("background.jpg")  
photo=ImageTk.PhotoImage(image)  
bg=Label(image=photo).pack()
```

Selanjutnya terdapat 3 buah button, button pertama akan menampilkan keterangan "File", button ke dua akan menampilkan keterangan "Teks", dan button ketiga akan menampilkan keterangan "Bantuan", dengan pengaturan yang sama pada ke 3 button yaitu lebar atau width sebesar 25 dan tinggi atau height sebesar 2 serta warna button "blue" dan warna keterangan "black", setelah itu terdapat command yang akan berfungsi jika tombol tersebut diklik, didalam command ini memanggil 2 fungsi, untuk memanggil 2 fungsi sekaligus pada python dapat menggunakan perintah lambda, fungsi pertama yaitu fungsi untuk menutup window menu utama, selanjutnya fungsi ke dua untuk menampilkan window yang dituju. Selanjutnya perintah mainloop() untuk menampilkan window yang telah dibuat.

```
b1=Button(window, text="File", width=25, height=2, bg='#539fd3', fg='black',  
command=lambda:[window.destroy(),AES_encrp.file_aes_window()]).place( x=50, y=150)  
b2=Button(window, text="Teks", width=25, height=2, bg='#539fd3', fg='black',  
command=lambda:[window.destroy(),AES_encrp.text_aes_window()]).place( x=50,y=200)  
b3=Button(window, text="Bantuan", width=25, height=2, bg='#539fd3', fg='black',  
command=lambda:[window .destroy(),AES_encrp.bantuan_window()]).place( x=50,y=250)  
window.mainloop()
```

Selanjutnya kodingan berikut untuk membuat fungsi yang akan menjalankan fungsi main pada class AES\_encrp pada saat program dijalankan.

```
if __name__ == "__main__":  
    AES_encrp.main()
```

#### 4.1.3. Pembuatan Menu File

Menu file merupakan halaman yang digunakan untuk melakukan proses enkripsi dan dekripsi yang berjenis file. Pada halaman menu file menggunakan 3 button, 2 label, dan 2 radiobutton. Button 1 sebagai button cari file yang berfungsi untuk memasukkan file yang akan diproses. Button 2 sebagai button proses yang akan melakukan pemrosesan jika sudah memasukkan file, kata sandi dan memilih jenis proses. Button 3 sebagai button reset yang berfungsi untuk menghapus pada halaman menu file. Selanjutnya radiobutton 1 digunakan untuk memilih proses dengan jenis enkripsi sedangkan radiobutton 2 digunakan untuk memilih proses dengan jenis dekripsi.

Kodiangan berikut ini untuk membuat fungsi baru yang digunakan untuk membuat window file. Buat obyek Tk() yang berarti tkinter dengan variabel window. Selanjutnya perintah geometry untuk mengatur ukuran dari window yang akan ditampilkan, perintah title untuk memberi nama window yang akan dibuat pada window bar dan perintah resizable dengan pengaturan false, sehingga window yang dibuat tidak dapat diubah ukurannya. Selanjutnya terdapat variabel yang akan digunakan pada fungsi file\_aes\_window yaitu pass\_file, lokasi\_file dan radio\_var dengan tipe data string.

```
def file_aes_window():  
    window=Tk()  
    window.title('Enkripsi File')  
    window.geometry ("500x400" )  
    window.Resizable(False, False)  
    pass_file=StringVar()  
    lokasi_file=StringVar()  
    radio_var=StringVar()
```

Selanjutnya kodingan berikut untuk membuat label, entri teks dan button. Pada label 1 terdapat keterangan "Pilih File" lalu pada entri teks hubungkan dengan variabel lokasi\_file, setelah itu pada button berikan keterangan "Cari File" dan tambahkan command dengan lambda untuk menjalankan 2 perintah sekaligus yaitu perintah untuk menampilkan window yang digunakan untuk mengambil file pada direktori penyimpanan komputer dan perintah untuk menyimpan lokasi file yang dipilih pada atribut class lokasi. Setelah itu posisikan penempatan komponen dengan perintah place.

```
labe!2_l=Label(window, text="Pilih File", font=("arial", 12, "bold")).place( x=10, y=10)  
b2_l=Button(window, text="Cari File", width=14, bg='#b5b5b5', fg='black',
```

```
command=lambda:[ AES_encrp.openfile(), lokasi_file.set(AES_encrp. lokasi)])
.place(x=330,y=10)
entry2_1=Entry(window,textvar=lokasi_file,width=24).place(x= 170,y=12)
```

Kodingan berikut ini untuk membuat label dengan keterangan “Kata Sandi” lalu pada entri teks hubungkan dengan variabel pass\_file, selanjutnya posisikan dengan menggunakan perintah place, lalu tambahkan image sebagai tombol untuk menyembunyikan kata sandi dan label untuk menampilkan keterangan kata sandi.

```
label2_2=Label(window, text="Kata Sandi", font=("arial", 12,"bold")).place( x=10, y=7 5)
photo=PhotoImage(file="hide.png")
potohide = photo.subsample(2, 2)
b2_4 = Button(window,text="Hide",image=potohide,width=14,command=
lambda: [entry2_2.config(show=""),b2_5.place(x=445,y=75), b2_4.place_forget()])
```

```
b2_4.place(x=445,y=75)
b2_5 = Button(window,text="Hide",image=potohide,width=14, commands
lambda: [entry2_2.config(show="*"),b2_4.place(x=445,y=75),
b2_5.place_forget()])
label2_5=Label(window,text="*Kata sandi minimal 6 digit",font=("arial",
8, "bold")). place (x=170,y=102)
label2_6=Label(window,text="*Kata sandi dapat mengandung huruf, angka dan
simbol",font=("arial", 8,"bold")).piace (x=170,y=120)

entry2_2=Entry(window, textvar=pass_file, width=24).place(x=170, y=77)
```

Selanjutnya kodingan berikut ini untuk membuat label dengan keterangan “Jenis Proses” dan radio button yang digunakan untuk memilih proses yang akan dilakukan. Terdapat 2 radio button yang masing-masing memiliki keterangan “Enkripsi” dan “Dekripsi” , lalu hubungkan radio button dengan variabel radio\_var, setelah itu posisikan komponen tersebut menggunakan perintah place.

```
label2_3=Label(window, text="Jenis Proses", font=("arial", 12,"bold")).place(
x=10,y=140)
```

```

r1=Radiobutton( window,text="Enkripsi",variable=radio_var,value="Enkripsi").
place(x= 165,y= 140)

```

```

r1 =Radiobutton( window,text="Dekripsi",variable=radio_var,value="Deskripsi"
).place(x=270,y= 1 40)

```

Setelah itu terdapat button dengan keterangan “Proses”, pada button ini terdapat command yang akan menjalankan 3 perintah sekaligus ketika tombol tersebut diklik dengan menggunakan lambda, perintah pertama yaitu akan menyimpan opsi yang dipilih melalui radio button ke dalam variabel atribut pada class program, selanjutnya perintah kedua yaitu akan menyimpan kata sandi yang diinputkan user kedalam variabel atribut pada class program, perintah terakhir yaitu memanggil fungsi proses\_aes\_file(), fungsi ini akan menjalankan proses enkripsi atau dekripsi file.

```

b2_2=Button( window, text="Proses", width=14, bg='#b5b5b5', fg='black',
command=lambda:[setattr(AES_encrp,'pilihan',radio_var.get()),
setattr(AES_encrp, 'sandi', pass_file.get()),
AES_encrp.proses_aes_file())].place(x= 100,y=2 10)

```

Selanjutnya kodingan berikut untuk membuat button dengan keterangan “Reset”, pada button ini terdapat command yang akan menjalankan 2 perintah sekaligus untuk mengosongkan field teks kata sandi dan field teks lokasi. Setelah itu tambahkan perintah mainloop() untuk menampilkan tampilan window enkripsi file.

```

b2_3=Button(window,text="Reset", width=14, bg='#b5b5b5', fg='black',
command=lambda:[lokasi_file.set(""), pass_file.set("")] ).place(x=250,y=2 10)
window. Mainloop()

```

#### 4.1.4. Pembuatan Menu Teks

Selanjutnya adalah halaman menu teks yang menggunakan 2 button, 2 radiobutton dan 3 label. Button 1 sama seperti pada halaman menu file sebagai button proses yang akan melakukan pemrosesan jika sudah memasukkan file, kata sandi dan memilih jenis proses. Sedangkan button 2 sebagai button reset yang akan menghapus file dan kata sandi setelah dimasukkan. Kemudian radiobutton 1 digunakan untuk memilih proses dengan jenis enkripsi sedangkan radiobutton 2 digunakan untuk memilih proses dengan jenis dekripsi.

Kodingan berikut untuk membuat fungsi baru yang digunakan untuk membuat window menu teks. Buat obyek Tk() yang berarti tkinter dengan variable window. Selanjutnya terdapat perintah geometry untuk mengatur ukuran dari window yang akan ditampilkan, perintah title



untuk memberi nama window yang akan dibuat pada window bar dan perintah resizable dengan pengaturan false, sehingga window yang dibuat tidak dapat diubah ukurannya. Selanjutnya terdapat variabel yang akan digunakan pada fungsi text\_aes\_window yaitu input\_text\_aes, pass\_text, radio\_var dan cipher\_text\_aes dengan tipe data string pada masing-masing variabel.

```
def text_aes_window():  
    window=Tk()  
    window.title("Enkripsi Text")  
    window.geometry("500x400")  
    window.resizable(False,False)  
    input_text_aes=StringVar()  
    pass_text=StringVar()  
  
    radio_var=StringVar()  
    cipher_text_aes=StringVar()
```

Selanjutnya kodingan berikut untuk membuat label dengan keterangan “Informasi Teks” yang memiliki pengaturan sama seperti label sebelumnya, lalu tampilkan dengan menggunakan perintah place. Selanjutnya terdapat entri teks yang akan digunakan untuk menerima informasi pesan yang akan dienkripsi atau dekripsi, hubungkan entri teks yang dibuat ke variabel input\_text\_aes, lalu tampilkan dengan menggunakan perintah place.

```
label3_1=Label(window, text="Informasi Text", font=("arial", 12, "bold"))  
label3_1.place(x=10,y=10)  
entry3_1=Entry(window,textvar=input_text_aes,width=35).place(x=170,y=120)
```

Selanjutnya kodingan berikut untuk membuat label dengan keterangan “Kata Sandi” yang memiliki pengaturan sama seperti label sebelumnya, lalu tampilkan dengan menggunakan perintah place. Terdapat entri teks yang akan digunakan untuk menerima inputan kata sandi yang digunakan untuk proses enkripsi atau dekripsi, lalu hubungkan entri teks yang dibuat ke variabel pass\_text, lalu tampilkan dengan menggunakan perintah place, lalu tambahkan image sebagai tombol untuk menyembunyikan kata sandi dan label untuk menampilkan keterangan kata sandi.

```
label3_2=Label(window, text="Kata Sandi", font=("arial", 12, "bold")),  
label3_2.place(x=10,y=120)  
photo=PhotoImage(file="hide.png")
```

```

potohide - photo.subsample(2, 2)
b2_4 = Button(window,text="Hide",image=potohide,width=14,command=
lambda:[entry]
3_2.config(show=""),b2_5.place(x=475,y=80),b2_4.place_forget())
b2_4.place(x=475,y=80)
b2_5 = Button(window,text="Hide",image=potohide,width=14,command=
lambda: [entry]
3_2.config(show="*"),b2_4.place(x=475,y=80),b2_5.place_forge
T(l)label3_5=Label(window,text="*Kata sandi minimal 6 digit",font=("arial",8, "bold")).
place (x=170,y=102)label3_6=Label(window,text="*Kata sandi dapat mengandung huruf,
angka dan simbol",font=("arial", 8,"bold")).place (x=170,y=120)entry
3_2=Entry(window,textvar=pass_text,width=40).place(x= 170, y= 122)

```

Setelah itu kodinga berikut ini untuk membuat label dengan keterangan “Jenis Proses” dan radio button yang digunakan untuk memilih proses yang akan dilakukan. Terdapat 2 radio button yang masing-masing memiliki keterangan “Enkripsi” dan “Dekripsi”, lalu hubungkan radio button dengan variabel radio\_var, setelah itu posisikan komponen tersebut menggunakan perintah place.

```

label3_3=Label(window, text="Jenis Proses", font=("arial", 12,"bold")).place(x=10,y=160)

r3=Radiobutton( window, text="Enkripsi",variable=radio_var,value="Enkripsi")

.place(x=165,y=161) r4=Radiobutton(
window,text="Deskripsi",variable=radio_var,value="Deskripsi").place(x=270,y=161)

```

Selanjutnya kodingan berikut untuk membuat label dengan keterangan Hasil” yang memiliki pengaturan sama seperti label sebelumnya, lalu tampilkan dengan menggunakan perintah place. Terdapat entri teks yang akan digunakan untuk menampilkan hasil proses enkripsi atau dekripsi, lalu hubungkan entri teks yang dibuat ke variabel cipher\_text\_aes, lalu tampilkan dengan menggunakan perintah place.

```

labe!3_4=Label(window, text="Hasil", font=("arial", 12, "bold"))
.place (x=10,y=200)

entry3_4=Entry( window, textvar=cipher_text_aes, width =35)
.place(x= 170,y=200)

```

Setelah itu ketikkan kodingan berikut untuk membuat button dengan keterangan “Proses”, pada button ini terdapat command yang akan menjalankan 5 perintah sekaligus ketika tombol tersebut diklik dengan menggunakan lambda, perintah pertama yaitu akan menyimpan opsi yang dipilih melalui radio button kedalam variabel pilihan pada atribut class program. Selanjutnya perintah kedua yaitu akan menyimpan kata sandi yang diinputkan user kedalam variabel sandi pada atribut class program. Selanjutnya perintah ketiga yaitu akan menyimpan informasi teks yang diinputkan user kedalam variabel input\_text\_aes pada atribut class program. Selanjutnya perintah keempat yaitu memanggil fungsi proses\_aes\_text(), fungsi ini akan menjalankan proses enkripsi atau dekripsi file. Selanjutnya perintah terakhir yaitu dengan mengambil hasil enkripsi atau dekripsi pada variable cipher\_text pada atribut class program untuk ditampilkan pada entri teks “Hasil”. Lalu tampilkan tombol yang dibuat dengan perintah place.

```
def main():  
    window=Tk()  
    window, geometry! "500x400")  
    window .resizable(False, False)  
    window. title("Welcome")
```

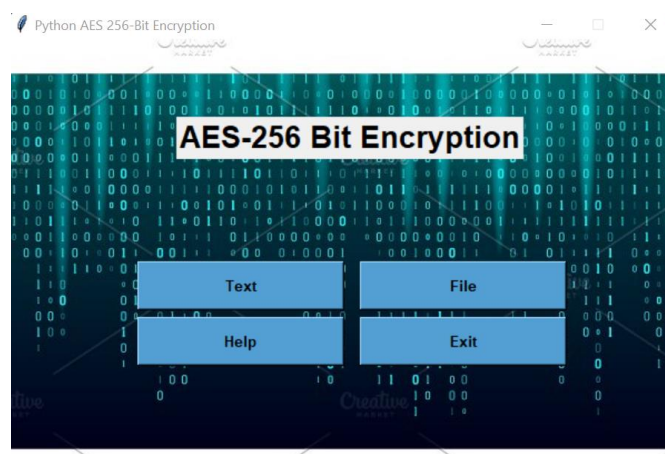
## 4.2. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk mengetahui hasil dari enkripsi dan dekripsi berbagai jenis file dan informasi teks. Pengujian dilakukan pada perangkat keras yang memiliki spesifikasi berbeda serta melakukan uji coba blackbox testing.

### 4.2.1. Uji Coba pada Perangkat Keras

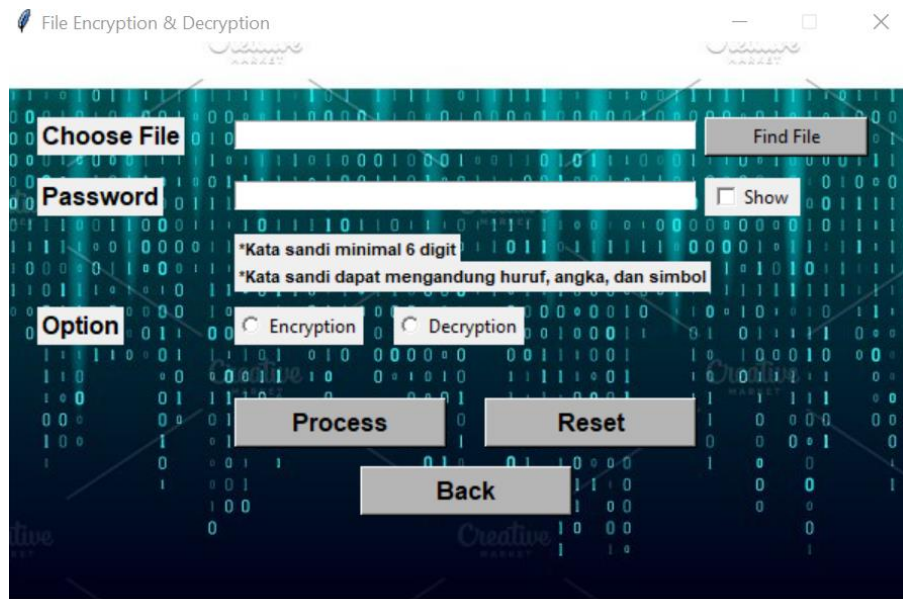
Pada pengujian aplikasi ini dilakukan pada beberapa perangkat keras dengan spesifikasi yang berbeda-beda untuk mengetahui apakah aplikasi dapat berjalan dengan baik dan untuk mengetahui minimal spesifikasi perangkat keras yang dapat digunakan untuk menjalankan aplikasi. Pertama, melakukan proses enkripsi berjenis file dengan klik tombol File pada menu utama seperti pada Gambar 4.1.

**Gambar 4.1 Menu Utama**



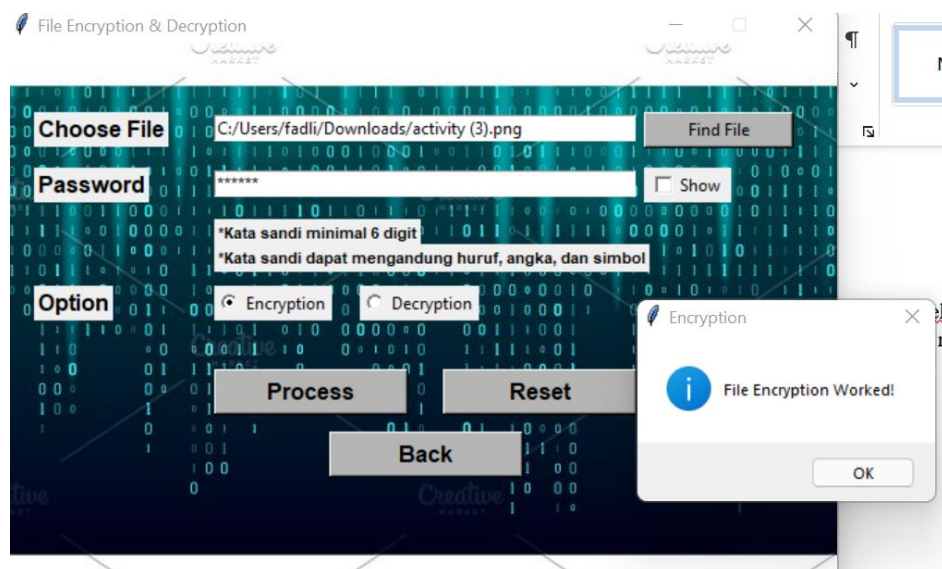
Aplikasi akan menampilkan halaman menu File jika tombol File diklik. Gambar 4.2 merupakan tampilan dari halaman menu File.

**Gambar 4.2 Menu File**



Isi semua bagian pada halaman menu File dan memilih jenis proses yaitu enkripsi kemudian klik tombol proses untuk melakukan pemrosesan. Pada saat tombol proses diklik maka akan muncul message box “Enkripsi File Berhasil!”. Gambar 4.3 merupakan hasil dari enkripsi berjenis file yang disimpan dalam folder.

**Gambar 4.3 Hasil Enkripsi File**



Selanjutnya uji coba melakukan proses enkripsi berjenis teks dengan klik tombol Teks pada menu utama. Gambar 4.4 merupakan tampilan dari halaman menu Teks.

**Gambar 4.4 Menu Teks**

The screenshot shows a window titled "Text Encryption & Decryption" with a dark blue background featuring a binary code pattern. On the left side, there are four labels: "Text", "Password", "Option", and "Result". The "Text" label is next to an empty text input field. The "Password" label is next to a password input field with a "Show" button to its right. Below the password field, there are two lines of instructional text: "\*Kata sandi minimal 6 digit" and "\*Kata sandi dapat mengandung huruf, angka, dan simbol". The "Option" label is next to two radio buttons: "Encryption" (which is selected) and "Decryption". The "Result" label is next to an empty text output field. At the bottom of the window, there are three buttons: "Process", "Reset", and "Back".

Ketikkan teks yang akan diproses pada informasi teks, isi semua bagian pada halaman menu Teks dan pilih jenis proses yaitu enkripsi. kemudian klik tombol proses untuk melakukan proses. Maka hasil akan muncul di kolom Hasil. Gambar 4.5 merupakan hasil dari enkripsi berjenis teks.

**Gambar 4.5 Hasil Enkripsi Teks**

This screenshot shows the same application window as in Gambar 4.4, but with data entered and the encryption process completed. The "Text" input field now contains the text "lalala". The "Password" input field contains six asterisks "\*\*\*\*\*", and the "Show" button is still present. The "Encryption" radio button remains selected. The "Result" output field now displays a long alphanumeric string: "4145530200001b435245415445445f42590070794165734372". The "Process", "Reset", and "Back" buttons are still visible at the bottom.

Isi semua bagian pada halaman menu File dan memilih jenis proses yaitu enkripsi kemudian klik tombol proses untuk melakukan pemrosesan. Pada saat tombol proses diklik maka akan muncul message box “Enkripsi File Berhasil!”. Gambar 1 1 merupakan hasil dari enkripsi berjenis file yang disimpan dalam folder.

**Tabel 4.1 Spesifikasi Perangkat Keras**

No	Nama Perangkat	Spesifikasi
1.)	Laptop-1055CEQO	<ul style="list-style-type: none"> <li>• Intel Core i5-1135G7</li> <li>•GPU NVIDIA GT-730m</li> <li>•Memori 8GB RAM</li> </ul>

Berdasarkan hasil uji coba perangkat keras yang telah dilakukan, didapatkan spesifikasi minimum untuk menjalankan aplikasi Enkripsi, yaitu perangkat keras laptop dengan sistem operasi windows 11, Intel Core i5-1135G7, GPU NVIDIA GT-730m . dan Memori 8GB RAM

#### **4.2.2. Uji Coba Berjenis File**

Pada bagian ini dilakukan uji coba terhadap hasil berjenis file. Uji coba dilakukan untuk mengetahui apakah file dapat dienkripsi dan dibuka Kembali setelah melakukan proses dekripsi. Proses dilakukan dengan menggunakan beberapa file dengan ekstensi berbeda. Tabel 4.2 merupakan hasil uji coba enkripsi untuk berjenis file.

**Tabel 4.2 Hasil Uji Coba Enkripsi File**

No	Jenis File	Ukuran File Sebelum Enkripsi	Ukuran File Sesudah Enkripsi	Hasil Enkripsi
1	DOCX	25.906 bytes	26.215 bytes	Berhasil
2	PDF	51.017 bytes	51.319 bytes	Berhasil
3	XLSX	10.727 bytes	11 .03 1 bytes	Berhasil
4	PPTX	1.512.314 bytes	1.5 12.615 bytes	Berhasil
5	RAR	29.427.57bytes	29.427.879 bytes	Berhasil

**Tabel 4.3 Hasil Uji Coba Dekripsi File**

No	Jenis File	Ukuran File Sebelum Enkripsi	Ukuran File Sesudah Enkripsi	Hasil Enkripsi
1	DOCX	26.215 bytes	25.906 bytes	Berhasil
2	PDF	51.319 bytes	51.017 bytes	Berhasil
3	XLSX	11.031 bytes	10.727 bytes	Berhasil
4	PPTX	1.512.615 bytes	1.512.3 14 bytes	Berhasil
5	RAR	29.427.879bytes	29.427.571 bytes	Berhasil

Dari hasil uji coba Tabel 4.2 dan Tabel 4.3 dapat dilihat hasil yang didapat bahwa aplikasi berhasil melakukan proses enkripsi dan dekripsi berbagai jenis file, ukuran file yang dihasilkan dari proses enkripsi akan lebih besar dari file aslinya, hal tersebut dikarenakan setiap byte binary file yang dienkripsi akan bertambah sehingga membutuhkan blok binary baru untuk menyimpan informasi byte yang sudah dienkripsi.

#### 4.2.3. Uji Coba Berjenis Teks

Pada bagian ini dilakukan uji coba terhadap hasil berjenis file. Uji coba dilakukan untuk mengetahui apakah teks dapat dienkripsi dan mengembalikan informasi seperti semula setelah melakukan proses dekripsi. Proses dilakukan dengan memasukan informasi berupa teks. Tabel 4.4 merupakan hasil uji coba enkripsi untuk berjenis teks.

Tabel 4.4 Hasil Uji Coba Enkripsi Teks

No	Jenis Teks	Teks Sebelum	Teks Sesudah	Kata Sandi	Jumlah Karakter	Hasil Enkripsi
1	Huruf	saya adalah mahasiswa gunadarma	41455 302000 01b4352 454154 45445f42590 070794165 73437279707 420302e342 e33008 000000 000000 0000000 000000000000 000000000000 000000000000 000000000000 000000000000 00000000 000000 000000 00000000 00000000 00000000 000000000000 000000000000 000000000000 00000000 00000000 000000000000 00000000 00000000	123456	32	BERHASIL

			0000000000000 0000000 00000000 000000000000000  00000004cbe03a5 6dae6c21b66d7e f126d60 77f5 2a855 d2c7305abaa5a35 25b8c4192bf24c79 112da62623fac3 d5f1b6f57446ff6f7 0d0bf434c98bff 13894bf4493ec497 2cc8fbf509f13c 8616f31e6cac91a 4d67f56e10ea6 da5210e16300c01 d5f531eb 1b92770 eed928 a2c5bd 5626d475fe995c ce3c826efcde 1ebf1a938f3aa4 1b006ae4e9d 020bc68ff073 5b72 e88c17 504e09b24 c9c33145 39565cb47eb03256f5			
--	--	--	--	--	--	--



**Tabel 4.5 Hasil Uji Coba Dekripsi Teks**

No	Jenis Teks	Teks Sebelum	Teks Sesudah	Kata Sandi	Jumlah Karakter	Hasil Enkripsi
1	Huruf	4145530200001b 435245415445 445f425900707 94165734372797 07420302e342e3 3008000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 00000000 000000000 0000000000000000 0000000 000000000 0000000000000000 00000000 00000000 00000 00000000 00000000 0000000 000000 00000000 0004cbe03a56dae 6c21b66d7ef126 d6077f5 2a855d2c 7305abaa5a3525 b8c4192 bf24c7911 2da62623fac3d5f 1b6f574 46ff6f70d 0bf434c98bff138 94bf4493 ec4972c c8fbf509f13c861 6f31e6cac91a4d6 7f56e10ea6da52 10e16300c01d5 f531eb1b92770ee d928a2c5bd56 26d475fe995cc e3c826efcde1eb f1a938f3aa41b006ae4e9	saya adalah mahasiswa gunadarma	123456	32	BERHASIL

		d020bc68ff0735b72e88c 17504e09b24c9c3314 539565cb47eb03256f5				
--	--	--	--	--	--	--

#### 4.2.4. Blackbox Testing

Blackbox Testing merupakan salah satu metode pengujian yang dilakukan untuk mengamati hasil dari eksekusi suatu aplikasi atau perangkat lunak. Pengamatan dilakukan dengan memeriksa fungsional dari aplikasi yang di uji. Hasil pengujian tersebut dapat dilihat pada Tabel 4.6.

**Tabel 4.6 Hasil Uji Coba BlackBox Testing**

No	Uji Kasus	Skenario Uji Coba	Hasil yang diharapkan	Hasil yang diperoleh	Hasil
1	Menjalankan aplikasi Enkripsi	Menjalankan aplikasi Enkripsi melalui python	Dapat berjalan dengan sempurna	Dapat berjalan dengan sempurna	VALID
2	Tampilan GUI Menu Utama aplikasi enkripsi	Klik tombol File	Aplikasi menampilkan halaman File	Aplikasi menampilkan halaman File	VALID
		Klik tombol Teks	Aplikasi menampilkan halaman Teks	Aplikasi menampilkan halaman Teks	VALID
		Klik tombol Bantuan	Aplikasi menampilkan halaman Bantuan	Aplikasi menampilkan halaman Bantuan	VALID
3	Proses enkripsi berjenis file	Klik Radio Button Enkripsi	Jenis proses berubah menjadi Enkripsi	Jenis proses berubah menjadi Enkripsi	VALID
		Klik tombol Proses	Aplikasi memulai proses Enkripsi file	Aplikasi akan memulai proses Enkripsi file	VALID
		Klik tombol Reset	Aplikasi menghapus bagian text box	Aplikasi menghapus bagian text box	VALID
4	Proses dekripsi berjenis file	Klik tombol Cari File	Aplikasi menampilkan halaman untuk memilih file yang	Aplikasi menampilkan halaman untuk memilih file yang	VALID

			akan didekripsi	akan didekripsi	
		Klik Radio Button Dekripsi	Jenis proses berubah menjadi Dekripsi	Jenis proses berubah menjadi Dekripsi	VALID
		Klik tombol Proses	Aplikasi memulai proses Dekripsi file	Aplikasi akan memulai proses Dekripsi file	VALID
		Klik tombol Reset	Aplikasi menghapus bagian textbox pada halaman File	Aplikasi menghapus bagian textbox pada halaman File	VALID
5	Proses enkripsi berjenis teks	Mengetikkan Teks yang akan dienkripsi pada textbox Informasi Teks	Teks akan tampil pada textbox Informasi Teks	Teks akan tampil pada textbox Informasi Teks	VALID
		Klik Radio Button Enkripsi	Jenis proses berubah menjadi Enkripsi	Jenis proses berubah menjadi Enkripsi	VALID
		Klik tombol Proses	Aplikasi memulai proses Enkripsi Teks	Aplikasi memulai proses Enkripsi Teks	VALID
		Klik tombol Reset	Aplikasi menghapus bagian textbox pada halaman Teks	Aplikasi menghapus bagian textbox pada halaman Teks	VALID
6	Proses dekripsi berjenis teks	Mengetikkan Teks yang akan didekripsi pada textbox Informasi Teks	Jenis proses berubah menjadi Dekripsi	Jenis proses berubah menjadi Dekripsi	VALID
		Klik Radio Button Dekripsi	Aplikasi memulai proses Dekripsi	Aplikasi memulai proses Dekripsi	VALID

			Teks	Teks	
		Klik tombol Proses	Aplikasi menghapus bagian textbox pada halaman Teks	Aplikasi menghapus bagian textbox pada halaman	VALID
		Klik tombol Reset	Aplikasi menghapus bagian textbox pada halaman Teks	Aplikasi menghapus bagian textbox pada halaman Teks	VALID

## **BAB 5**

### **KESIMPULAN**

Tugas project ini telah dilakukan dan menghasilkan aplikasi yang dapat mengenkripsi dan dekripsi berbagai jenis file dan informasi teks dengan menggunakan metode AES 256 bit. Kemudian aplikasi dilakukan uji coba untuk mengetahui apakah aplikasi dapat berfungsi dengan baik. Aplikasi dapat digunakan untuk type file DOCX, PDF, XLSX, PPTX, RAR, ZIP, JPEG, PNG, BMP, GIF, WMV, MP3, MP4, HTML, EXE, dengan ukuran minimum 3.335 bytes dan maksimum 29.427.571 bytes pada proses enkripsi sedangkan untuk proses dekripsi ukuran minimum 3.639 bytes dan maksimum 29.427.879 bytes, dan untuk proses enkripsi jenis teks panjang karakter minimum 11 karakter dan maksimum 419 karakter, sedangkan untuk proses dekripsi jenis teks panjang karakter minimum 404 karakter dan maksimum 1071 karakter. Berdasarkan hasil uji coba aplikasi untuk mengenkripsi berbagai jenis file, aplikasi ini mampu melakukan enkripsi dengan baik, sehingga informasi didalam file tersebut tidak mudah diketahui oleh orang lain yang tidak berhak. Selanjutnya pada uji coba dekripsi file, aplikasi ini mampu melakukan dekripsi dengan baik tanpa merusak informasi asli didalam file, sehingga file dapat dibuka dan digunakan kembali. Selanjutnya pada uji coba aplikasi untuk mengenkripsi dan dekripsi informasi teks dengan berbagai jenis teks huruf, angka dan simbol, aplikasi ini mampu mengenkripsi informasi teks dan mendekripsi kembali menjadi informasi teks semula dengan akurat