

# **LAPORAN TUGAS AKHIR**

## **STEGANOGRAPHY WITH GUI**

Disusun untuk memenuhi UAS Praktikum Sistem Multimedia

Mata Kuliah: Praktikum Sistem Multimedia



**Disusun Oleh:**

**Waqyu Setiadi      1197050130**

**Nur Galih Suntara   1197050099**

**KELAS E**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SUNAN GUNUNG DJATI BANDUNG**

**2022**

## A. PENDAHULUAN

Dalam proyek Python ini, kami akan membangun Steganografi Gambar berbasis GUI menggunakan modul Tkinter dan PIL. Ini adalah proyek tingkat menengah, dan Anda akan dapat melakukan spycraft dengan pesan rahasia menggunakan gambar dan menerapkan konsep yang dibahas dalam proyek ini dalam kehidupan nyata. Steganografi umumnya mengacu pada encoding dan decoding pesan data berbasis teks menjadi file non-teks seperti file audio, gambar dan file video. Dalam project ini, kita hanya akan meng-encode dan decode data menjadi gambar saja. Tujuan dari proyek ini adalah untuk membuat proyek Steganografi Gambar Python berbasis GUI. Untuk membangun ini, Anda memerlukan pemahaman menengah tentang Tkinter dan pemahaman dasar tentang perpustakaan PIL.

*Graphical user interface* (GUI) adalah sistem komponen visual interaktif untuk *software* komputer. Menurut [Computer Hope](#), GUI menampilkan objek yang bisa menyampaikan informasi dan merepresentasikan aksi dari pengguna. Dengan adanya GUI, kita bisa mengetahui bahwa apa yang kita input telah diterima dan responsnya ditunjukkan secara visual. GUI bisa terlihat dari perubahan warna, ukuran, visibilitas, dan sejenisnya ketika terjadi sebuah interaksi. Awalnya, GUI dikembangkan di tahun 1970 akhir oleh Xerox Palo Alto. Setelah berhasil, sistem ini digunakan oleh Apple untuk Macintosh-nya dan juga sistem operasi Windows yang dimiliki oleh Microsoft. Ditemukannya GUI memberikan solusi bagi permasalahan respon yang dirasakan oleh pengguna.

## B. KEBUTUHAN DASAR

- 1) Tkinter digunakan untuk membuat GUI
- 2) PIL atau Python Image Manipulation Library digunakan untuk menjalankan operasi pada gambar dalam python

## C. STRUKTUR PROJEK

- 1) Menginisialisasi jendela root dan menempatkan semua komponen di dalamnya
- 2) Mendefinisikan semua fungsi enkripsi dan dekripsi backend
- 3) Defining the GUI functions of decoding and encoding

#### D. KODE DAN PENJELASAN (BAGIAN-BAGIAN DARI KODING)

##### 1) Modul Library yang dibutuhkan

```
# Importing the necessary modules
from tkinter import *
from tkinter import messagebox as mb
from PIL import Image
```

Berisi seluruh modul yang dari library yang dibutuhkan sebagai penunjang untuk menjalankan kode GUI steganography

##### 2) Menginisialisasi jendela root dan menempatkan semua komponen di dalamnya

```
# Initializing the window
root = Tk()
root.title('ProjectUAS With GUI Image Steganography')
root.geometry('300x200')
root.resizable(0, 0)
root.config(bg='NavajoWhite')

Label(root, text='ProjectUAS With GUI Image Steganography', font=('Comic Sans
MS', 15), bg='NavajoWhite',
      wraplength=300).place(x=40, y=0)

Button(root, text='Encode', width=25, font=('Times New Roman', 13),
      bg='SteelBlue', command=encode_image).place(
      x=30, y=80)

Button(root, text='Decode', width=25, font=('Times New Roman', 13),
      bg='SteelBlue', command=decode_image).place(
      x=30, y=130)

# Finalizing the window
root.update()
root.mainloop()
```

Kami akan menetapkan kelas Tk() ke variabel root untuk menginisialisasi jendela. Metode dan atribut yang perlu diatur selama inisialisasi adalah:

- title() digunakan untuk memberi judul pada jendela.
- geometry() digunakan untuk menentukan dimensi awal jendela dalam piksel.
- resizable() menentukan apakah pengguna akan diizinkan untuk mengubah ukuran jendela atau tidak. Yang dibutuhkan hanya nilai benar dan salah yang berupa (lebar, tinggi).

- config() digunakan untuk mengonfigurasi atribut lain dari jendela, seperti atribut bg yang menunjukkan warna latar belakang jendela.
- update() dan .mainloop() menempatkan jendela dalam satu lingkaran untuk mencegahnya menutup hingga diberi tahu sebaliknya.

Catatan:

- Kedua baris ini akan menjadi baris terakhir dalam skrip GUI yang akan dijalankan oleh juru bahasa Python di dalamnya.
- Kelas Label digunakan untuk membuat Label pada jendela yang menampilkan teks statis pada jendela. Parameternya yang perlu diatur selama penugasan adalah:
- Parameter master, akar argumen posisional di sini, adalah widget induk yang dikaitkan dengan widget ini.
- Parameter teks mengacu pada teks yang akan ditampilkan pada label.
- Parameter font mendefinisikan keluarga font, ukuran dan efek yang akan diterapkan pada teks yang ditampilkan pada label.
- Parameter wraplength menentukan jumlah piksel setelah teks harus dipindahkan ke baris berikutnya.
- Kelas Button digunakan untuk menambahkan tombol ke jendela yang menjalankan fungsi saat ditekan. Parameternya adalah:
- Parameter perintah digunakan untuk menentukan fungsi yang akan dijalankan saat tombol ditekan. Anda tidak memerlukan kata kunci lambda jika fungsi yang dijalankan tidak memerlukan argumen apa pun.
- Metode .place() digunakan untuk menempatkan widget pada widget induknya seolah-olah induknya adalah Bidang Kartesius, dengan sudut Barat Laut sebagai asal bidang itu.
- Parameter x,y menentukan offset horizontal dan vertikal widget, masing-masing.
- Parameter relx,rely mendefinisikan offset horizontal dan vertikal sebagai angka desimal antara 0,0 dan 1,0

### 3) Mendefinisikan semua fungsi enkripsi dan dekripsi backend

```
def generate_data(pixels, data):
    # This function will convert the incoming data to 8-bit binary format
    # using its ASCII values and return them
    data_in_binary = []

    for i in data:
        binary_data = format(ord(i), '08b')
        data_in_binary.append(binary_data)

    length_of_data = len(data_in_binary)
    image_data = iter(pixels)

    for a in range(length_of_data):
        pixels = [val for val in image_data.__next__(
        )[:3] + image_data.__next__()[:3] + image_data.__next__()[:3]]
```

```

        for b in range(8):
            if (data_in_binary[a][b] == '1') and (pixels[b] % 2 != 0):
                pixels[b] -= 1
            elif (data_in_binary[a][b] == '0') and (pixels[b] % 2 == 0):
                if pixels[b] == 0:
                    pixels[b] += 1
                pixels[b] -= 1

        if (length_of_data-1) == a:
            if pixels[-1] % 2 == 0:
                if pixels[-1] == 0:
                    pixels[-1] += 1
                else:
                    pixels[-1] -= 1

        pixels = tuple(pixels)

        yield pixels[:3]
        yield pixels[3:6]
        yield pixels[6:9]

def encryption(img, data):
    # This method will encode data to the new image that will be created
    size = img.size[0]
    (x, y) = (0, 0)

    for pixel in generate_data(img.getdata(), data):
        img.putpixel((x, y), pixel)
        if size-1 == x:
            x = 0
            y += 1
        else:
            x += 1

def main_encryption(img, text, new_image_name):
    # This function will take the arguments, create a new image, encode it and
    save it to the same directory
    image = Image.open(img, 'r')

    if (len(text) == 0) or (len(img) == 0) or (len(new_image_name) == 0):
        mb.showerror(
            "Error", 'Kamu belum menyimpan Nilai! Simpan semua nilai yang
            dibutuhkan sebelum menekan tombol!')

    new_image = image.copy()
    encryption(new_image, text)

```

```

new_image_name += '.png'

new_image.save(new_image_name, 'png')

def main_decryption(img, strvar):
    # This function will decode the image given to it and extract the hidden
    message from it
    image = Image.open(img, 'r')

    data = ''
    image_data = iter(image.getdata())

    decoding = True

    while decoding:
        pixels = [value for value in image_data.__next__(
        )[:3] + image_data.__next__():[:3] + image_data.__next__():[:3]]

        # string of binary data
        binary_string = ''

        for i in pixels[:8]:
            if i % 2 == 0:
                binary_string += '0'
            else:
                binary_string += '1'

        data += chr(int(binary_string, 2))
        if pixels[-1] % 2 != 0:
            strvar.set(data)

```

- Dalam fungsi ini, kita akan mendapatkan data piksel dari gambar, dan kemudian mengubah data tersebut agar tampilan gambar tetap sama tetapi mengkodekan teks yang ingin kita sembunyikan dalam data.
- Kelas gambar digunakan untuk membuat file gambar dikenali oleh Python. Metodenya adalah:
  - i. open() digunakan untuk membuka file Gambar dalam mode bendera.
  - ii. getdata() digunakan untuk mendapatkan data piksel dari gambar.
  - iii. copy() digunakan untuk menyalin gambar dan menetakannya ke variabel lain.
  - iv. save(<filename>, <filetype>) digunakan untuk menyimpan file dengan nama file 'filename' dan ketik 'filetype'.
- Kehadiran kata kunci hasil memberi tahu kita bahwa fungsi generate\_data() adalah generator.
- Kata kunci hasil digunakan untuk mengembalikan nilai dari generator tanpa merusak variabel lokal atau dengan mengakhiri generator.
- fungsi iter() digunakan untuk mengembalikan nilai yang diberikan padanya sebagai argumen dalam bentuk tipe data yang dapat diubah.

- Fungsi `__next__()` digunakan untuk mendapatkan nilai berikutnya dari iterator yang menerapkannya.
- 4) Mendefinisikan fungsi GUI dari encoding dan decoding

```
# Creating the button functions
def encode_image():
    encode_wn = Toplevel(root)
    encode_wn.title("Encode an Image")
    encode_wn.geometry('600x220')
    encode_wn.resizable(0, 0)
    encode_wn.config(bg='AntiqueWhite')
    Label(encode_wn, text='Encode an Image', font=(
        "Comic Sans MS", 15), bg='AntiqueWhite').place(x=220, rely=0)

    Label(encode_wn, text='Enter the path to the image(with extension):',
font=("Times New Roman", 13),
        bg='AntiqueWhite').place(x=10, y=50)
    Label(encode_wn, text='Enter the data to be encoded:', font=("Times New
Roman", 13), bg='AntiqueWhite').place(
        x=10, y=90)
    Label(encode_wn, text='Enter the output file name (without extension):',
font=("Times New Roman", 13),
        bg='AntiqueWhite').place(x=10, y=130)

    img_path = Entry(encode_wn, width=35)
    img_path.place(x=350, y=50)

    text_to_be_encoded = Entry(encode_wn, width=35)
    text_to_be_encoded.place(x=350, y=90)

    after_save_path = Entry(encode_wn, width=35)
    after_save_path.place(x=350, y=130)

    Button(encode_wn, text='Encode the Image', font=('Helvetica', 12),
bg='PaleTurquoise', command=lambda:
        main_encryption(img_path.get(), text_to_be_encoded.get(),
after_save_path.get()))).place(x=220, y=175)

def decode_image():
    decode_wn = Toplevel(root)
    decode_wn.title("Decode an Image")
    decode_wn.geometry('600x300')
    decode_wn.resizable(0, 0)
    decode_wn.config(bg='Bisque')

    Label(decode_wn, text='Decode an Image', font=(
```

```

        "Comic Sans MS", 15), bg='Bisque').place(x=220, rely=0)

    Label(decode_wn, text='Enter the path to the image (with extension):',
font=("Times New Roman", 12),
        bg='Bisque').place(x=10, y=50)

    img_entry = Entry(decode_wn, width=35)
    img_entry.place(x=350, y=50)

    text_strvar = StringVar()

    Button(decode_wn, text='Decode the Image', font=('Helvetica', 12),
bg='PaleTurquoise', command=lambda:
        main_decryption(img_entry.get(), text_strvar)).place(x=220, y=90)

    Label(decode_wn, text='Text that has been encoded in the image:',
font=("Times New Roman", 12), bg='Bisque').place(
        x=180, y=130)

    text_entry = Entry(decode_wn, width=94, text=text_strvar,
state='disabled')
    text_entry.place(x=15, y=160, height=100)

```

- Kelas Toplevel berguna untuk membuat widget toplevel, atau jendela anak yang terkait dengan jendela root. Anda perlu mengatur atribut dan metode yang sama ke kelas ini seperti yang dilakukan selama pembuatan jendela root.
- Kelas Entry adalah untuk membuat bidang entri ke jendela tempat pengguna dapat memasukkan data. Parameter dan metode kelas ini yang tidak dibahas adalah:
  - i. Parameter status adalah untuk menentukan tindakan yang dapat dilakukan pengguna di dalam widget Entry.
  - ii. Standarnya, normal, memungkinkan pengguna untuk memasukkan data juga.
  - iii. Status readonly hanya memungkinkan pengguna untuk membaca dan menyalin data.
  - iv. Status dinonaktifkan hanya memungkinkan pengguna untuk membaca data saja.
  - v. Parameter teks adalah untuk mengatur variabel yang akan memanipulasi data di widget atau ke teks awal di dalam widget.
  - vi. Metode .get() kelas ini adalah untuk mendapatkan nilai di dalam widget yang dibuat kelas ini.

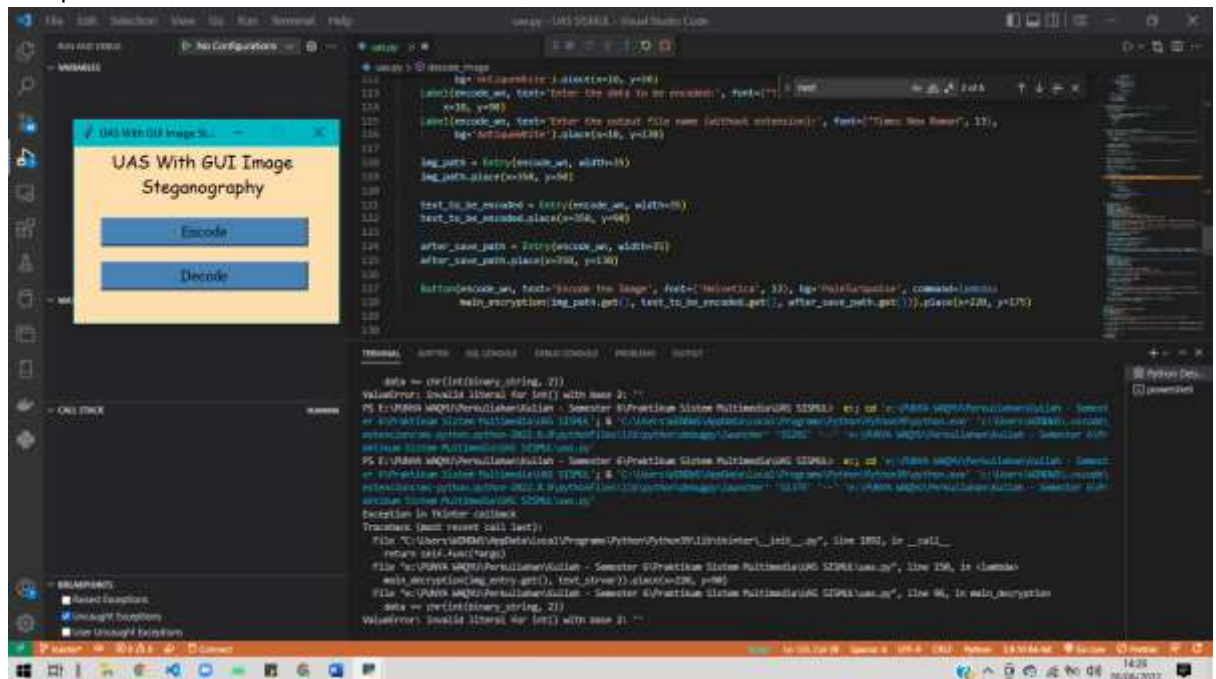
Kelas StringVar adalah untuk memanipulasi teks di dalam Label, Entry, dan banyak widget lainnya.

- set() dari kelas ini mengubah nilai widget ini menjadi argumen yang diberikannya.

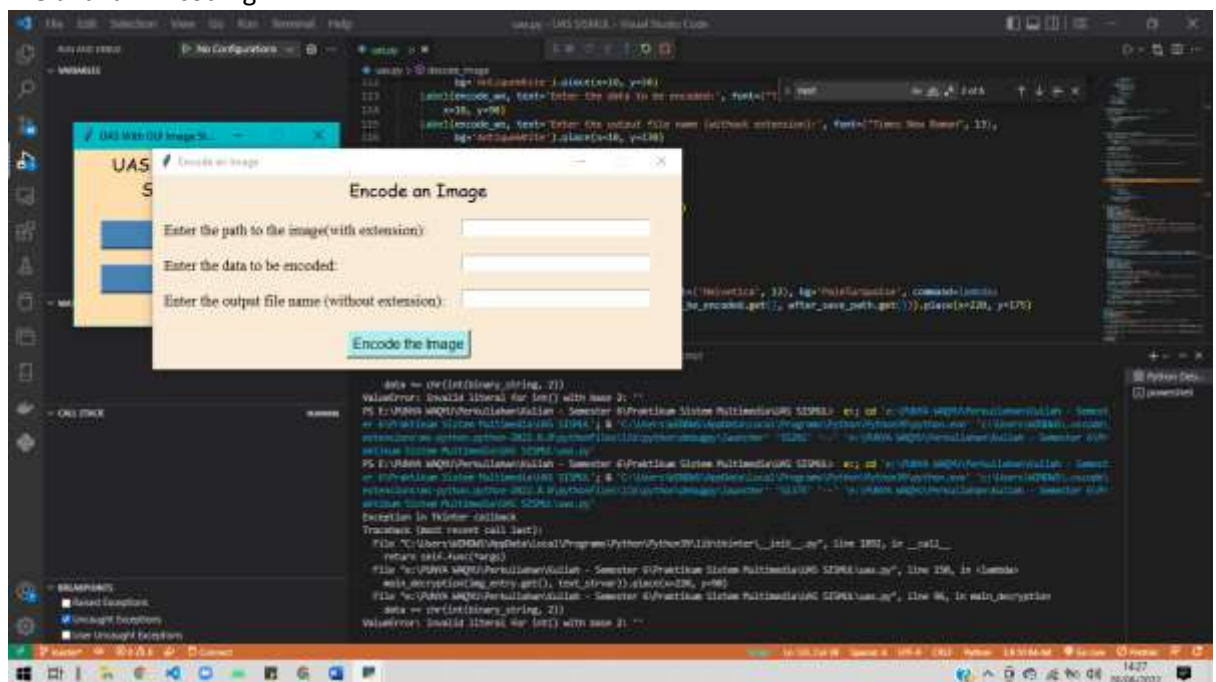


## E. SIMULASI APLIKASI

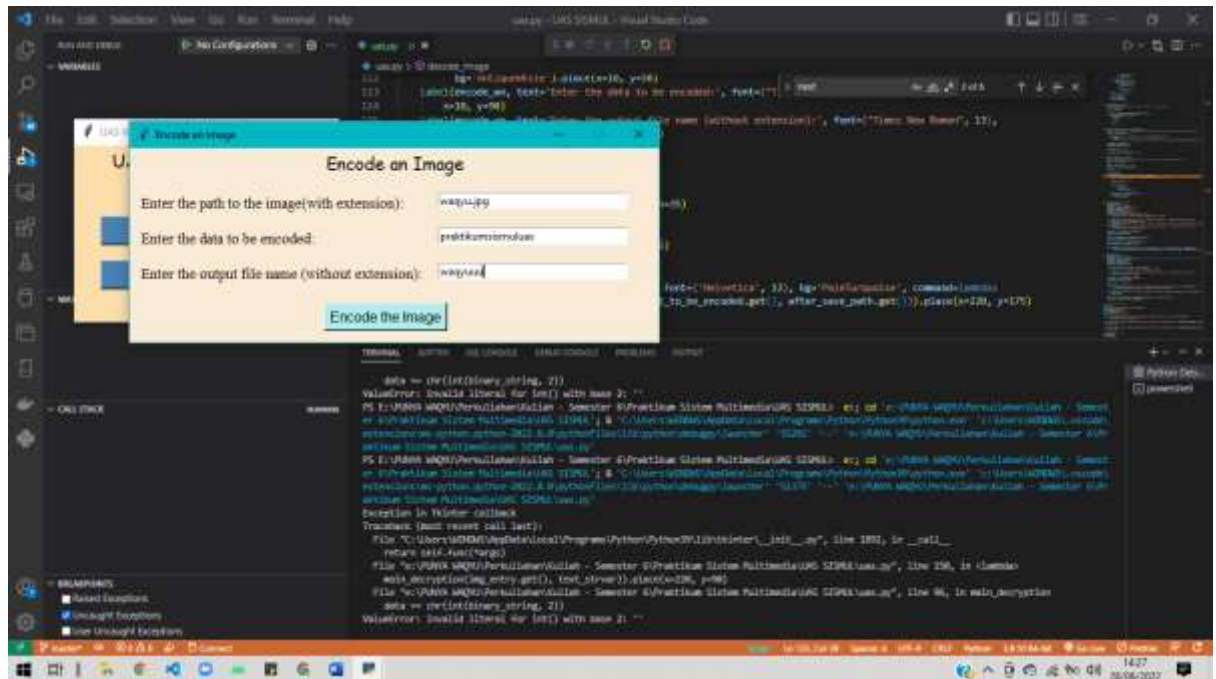
### 1) Tampilan awal



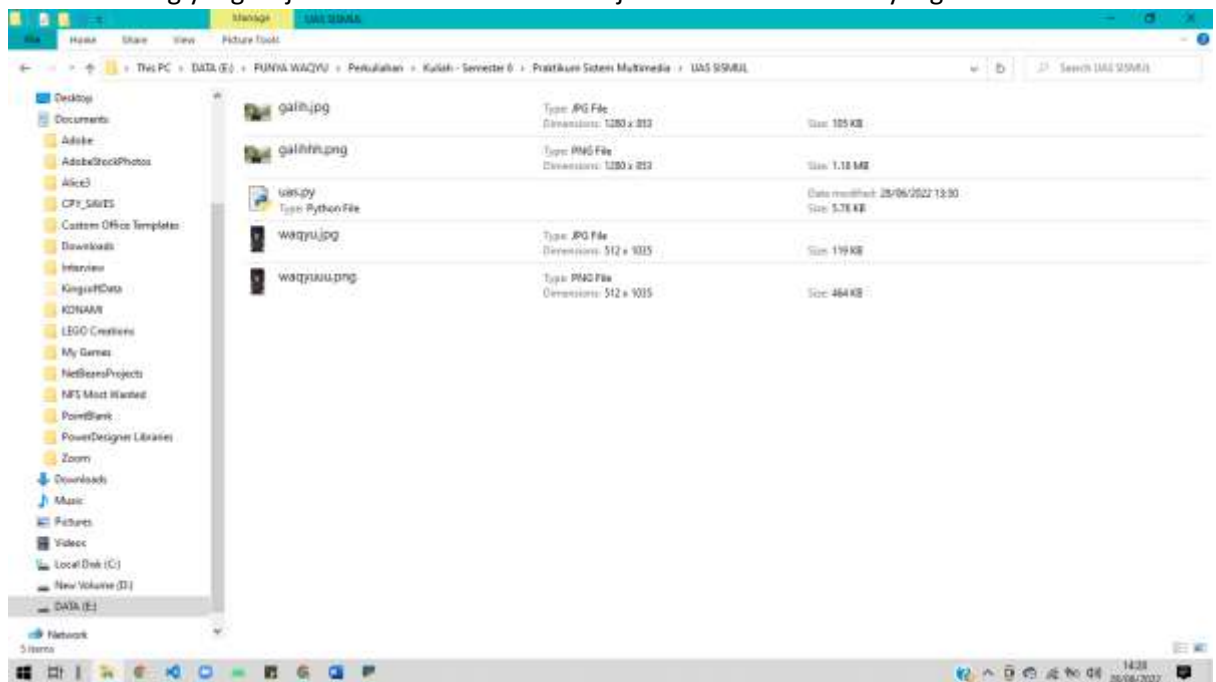
### 2) Melakukan Encoding



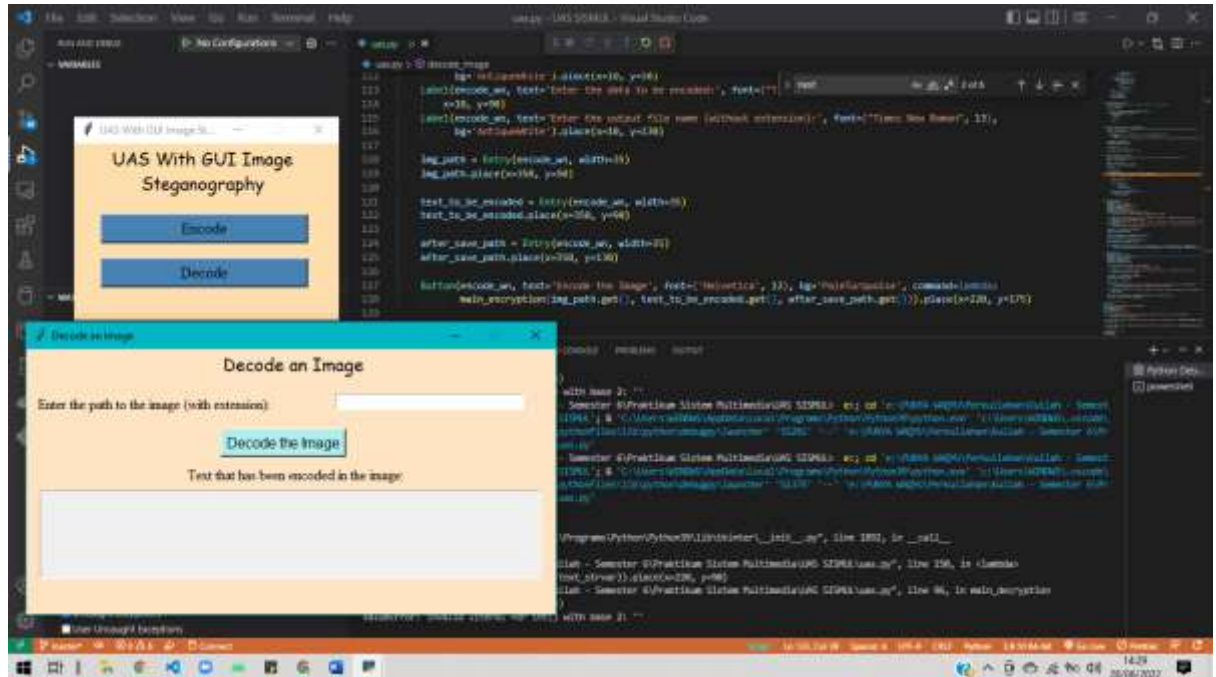
Masukan seluruh kebutuhan



Hasil encoding yang terjadi dimana menambahkan jenis file baru dan size yang bertambah



### 3) Melakukan Decoding



Memasukan segala kebutuhan dengan menghasilkan

