

# Laporan Proyek Capture Visitor

## Desripsi Umum Proyek Capture Visitor

Capture Visitor adalah sebuah program yang berfungsi untuk mengetahui visitor (pembuka aplikasi) pada sebuah program (aplikasi) dengan capturing (menangkap) foto dari pada visitor melalui webcam device. Data foto hasil capturing akan disimpan pada sebuah folder pada directory main program. Capture Visitor ini berfungsi untuk security system bagi para pengguna.

## Module yang Digunakan

*Module Built-in Python:*

TkInter : Module untuk membuat sebuah GUI (Graphical User Interface) yaitu suatu aplikasi desktop. Pada proyek ini, hanya digunakan fitur widget button, entry, dan messagebox untuk visualisasi dan memudahkan user menginputkan suatu data dan menerima informasi lain.

OS (Operating System) : Module untuk mengakses sistem operasi internal device. Pada proyek ini hanya digunakan fitur `os.getcwd()` yaitu untuk mengetahui directory main program yang sedang dijalankan dan `os.mkdir()` yaitu untuk membuat suatu folder.

SYS (System) : Module untuk mengakses sistem yang dalam hal ini interpreter python. Pada proyek ini hanya digunakan fitur `sys.path.insert()` untuk meletakkan directory pada main program.

\*Terletak pada program sekunder

*Module External Python:*

OpenCV : Module yang berguna untuk image processing atau pengolahan data foto (visual). Pada proyek ini hanya digunakan fitur:

`VideoCapture()` : untuk mengaktifkan webcam pada device yang digunakan

`VideoCapture.read()` : untuk menangkap data hasil dari rekaman (tangkapan) webcam

`waitKey()` : untuk memberikan jeda pada setiap milidetik tangkapan webcam

`imwrite()` : untuk menyimpan data hasil tangkapan yang berupa foto pada sebuah directory input-an user.

`VideoCapture.release()` : untuk menghentikan / menonaktifkan webcam

`destroyAllWindows()` : untuk menutup semua windows program yang ada.

## Alasan Memilih Module

TkInter : Alasan saya memilih module ini karena TkInter sudah pernah dipelajari di kelas DDP1 Fasilkom UI, memiliki fungsi-fungsi yang jelas serta disesuaikan dengan nama codenya, program untuk GUI built-in yang mudah dipelajari oleh para pengguna module.

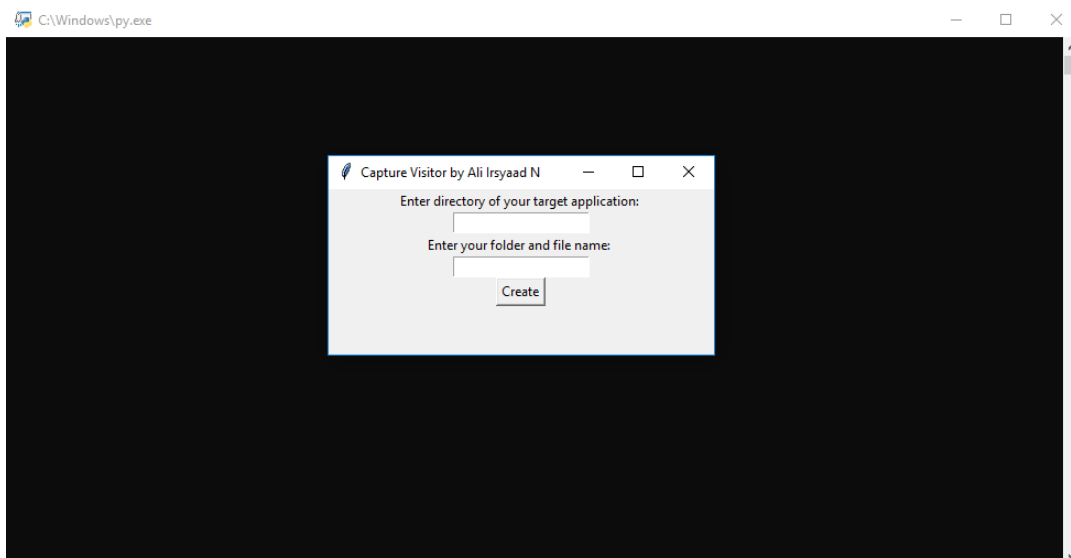
OS : Alasan saya memilih module ini karena OS merupakan module yang penting dan mungkin satu-satunya untuk mengakses sistem operasi.

SYS : Alasan saya memilih module ini karena SYS merupakan module yang penting dan mungkin satu-satunya untuk mengakses sistem

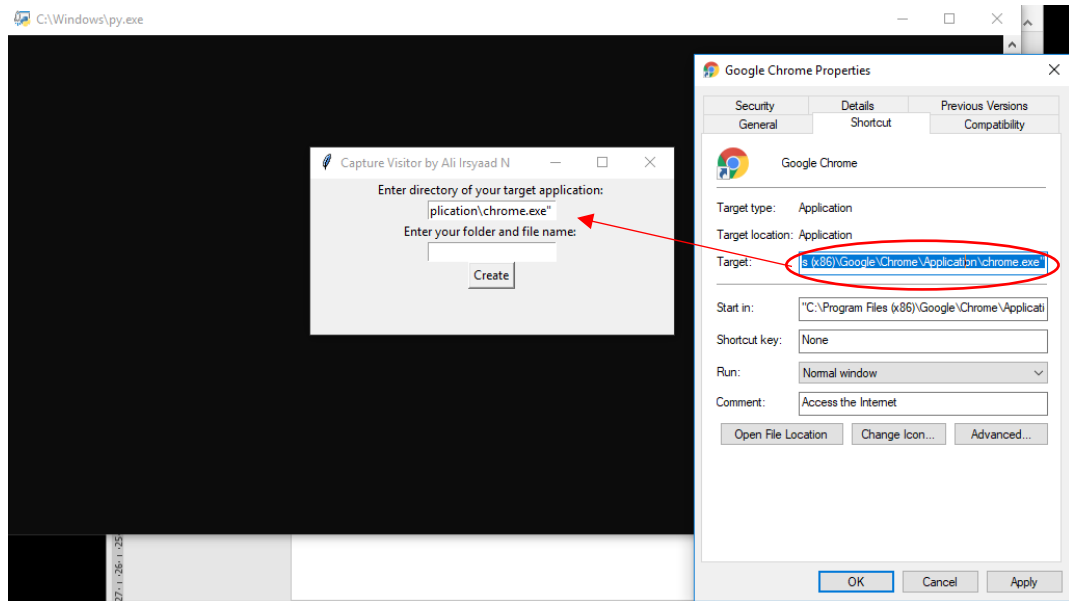
OpenCV : Alasan saya memilih module ini karena OpenCV merupakan module yang dapat mengakses webcam paling mudah. Selain itu OpenCV juga merupakan module yang direkomendasikan oleh python. Walaupun banyak sekali kegunaannya dalam hal mengolah data visual, tetapi pada proyek saya hanya digunakan untuk menangkap visual dari webcam dan menyimpannya.

## Cara Menjalankan Capture Visitor

1. Buka program Config Capture Visitor -> Akan tampil GUI untuk mengonfigurasi target yaitu berupa shortcut aplikasi dan nama folder serta file untuk disimpan.

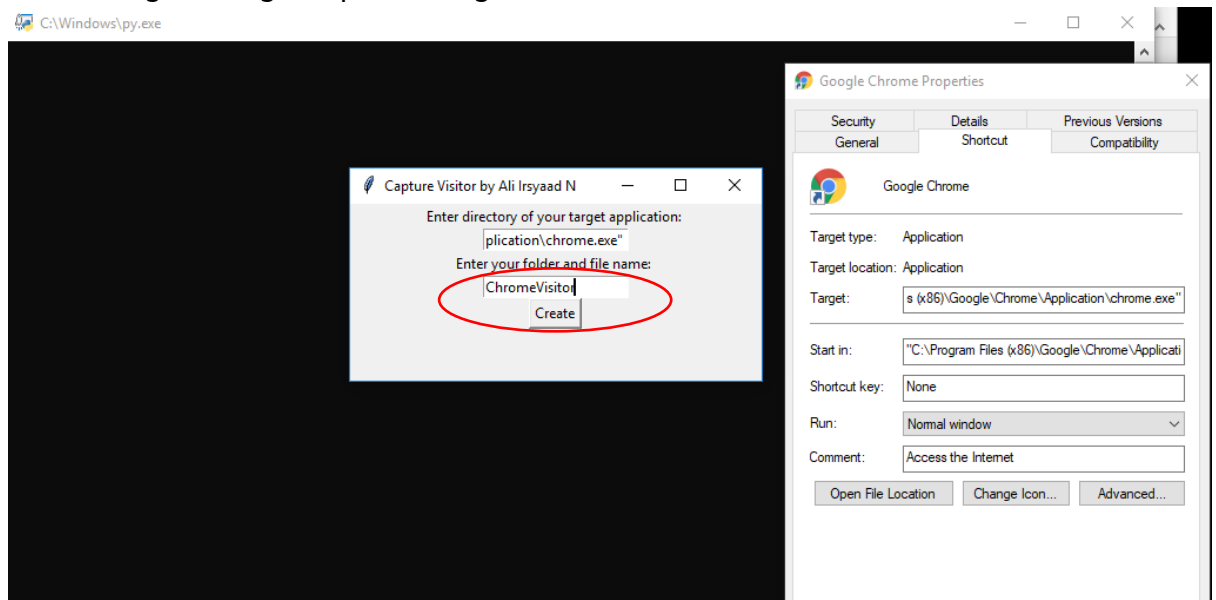


2. Entry pertama isikan target dari shortcut aplikasi yang ingin user tujuan (Contoh Google Chrome).



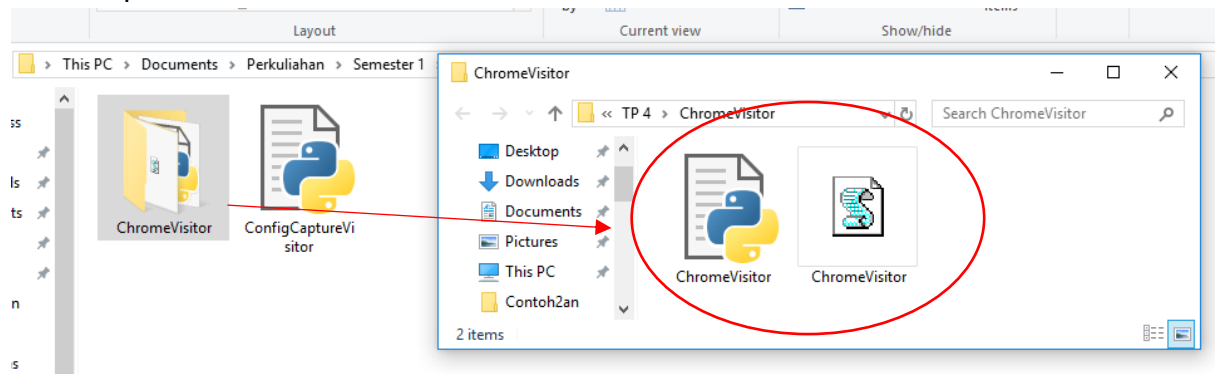
\*Pastikan entry target benar dan tidak kosong

3. Entry kedua isikan nama folder dan file yang diinginkan user (disarankan berhubungan dengan aplikasi target misal: ChromeVisitor) -> Lalu tekan Create

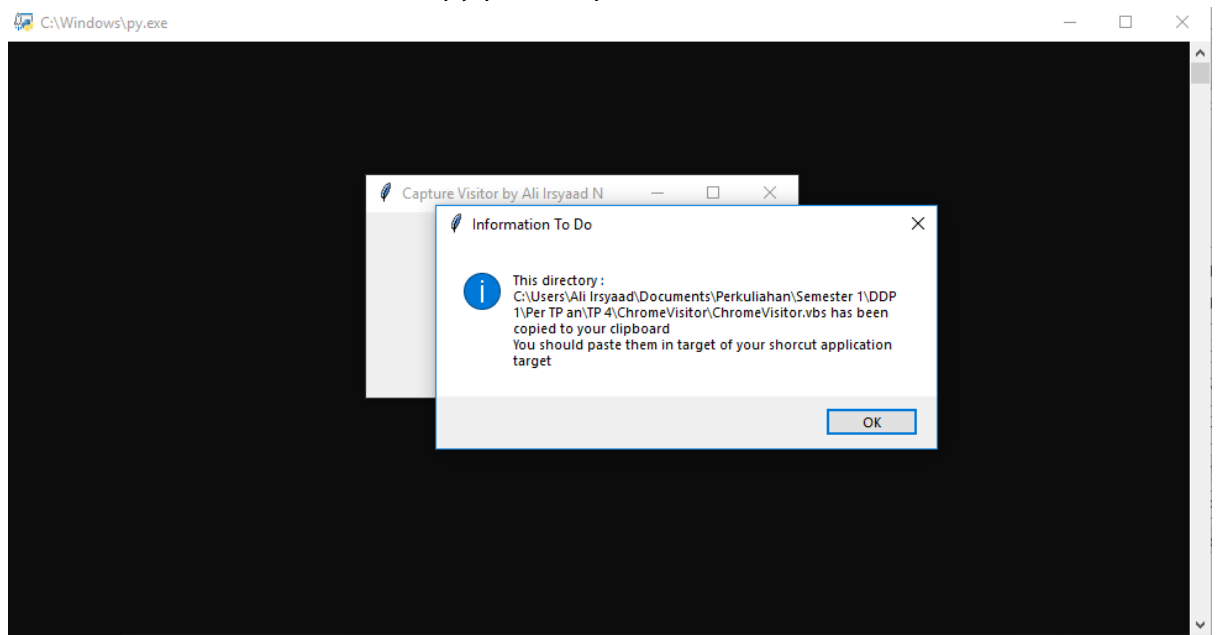


\*Pastikan entry nama folder dan file tidak sama dengan yang sudah terbuat dan tidak kosong

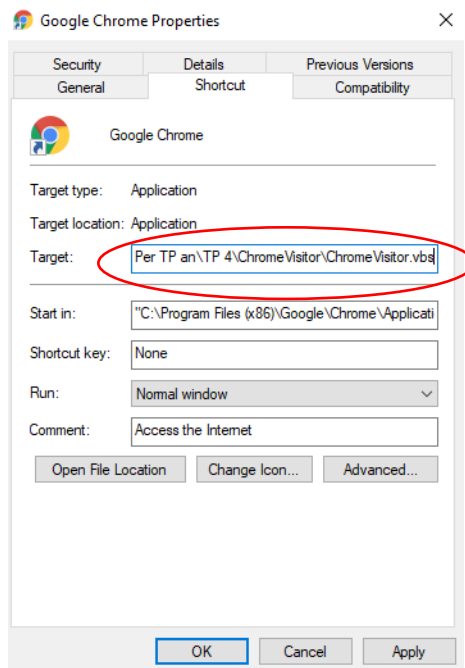
4. Disisi lain, ketika menekan button Create. Akan terbuat folder yang berisi 2 file yaitu file python dan file vbs. Folder dan file-file tersebut memiliki nama sama yaitu pada contoh ini adalah ChromeVisitor dengan akhiran yang berbeda sesuai tipe file.



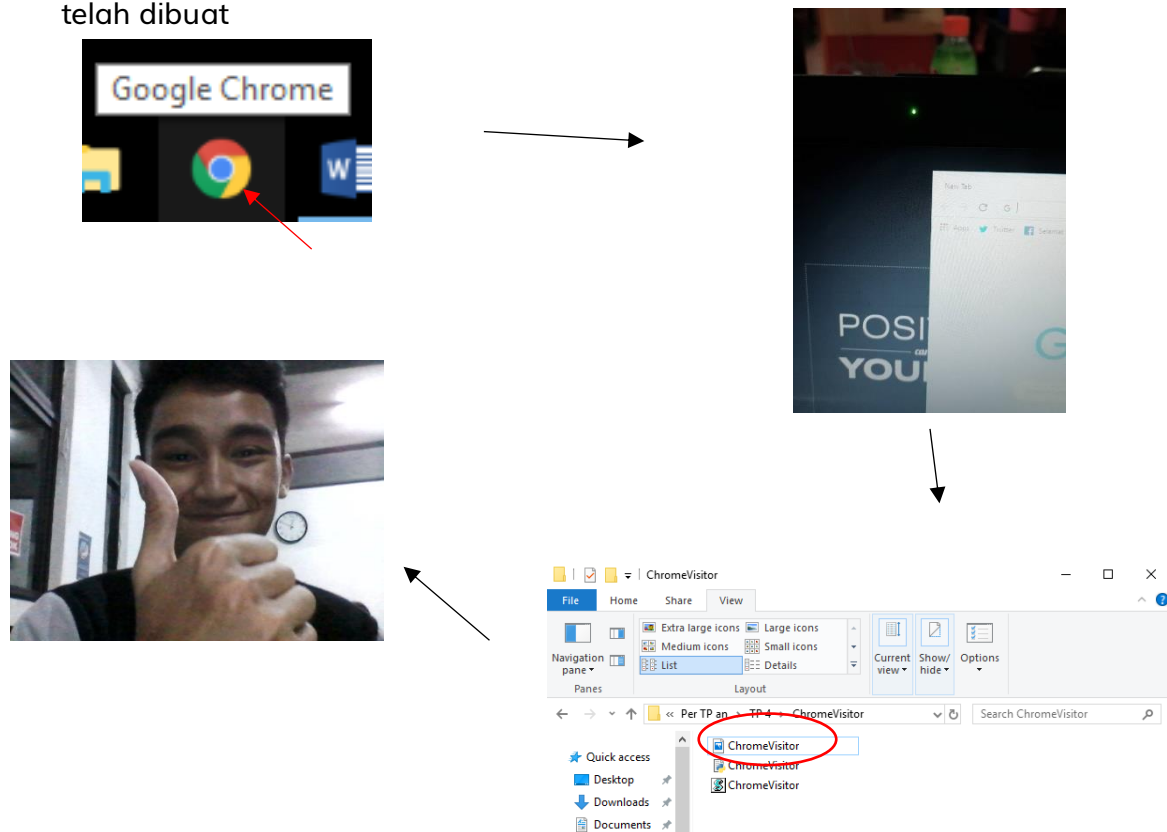
5. Disisi main program, akan tampil showinfo seperti ini. Maka directory yang disebutkan disana sudah ter-copy pada clipboard.



6. User diminta mem-pastekannya pada target aplikasi folder tersebut -> Klik OK.



7. Program siap dijalankan. Contoh: Visitor membuka aplikasi target -> file vbs akan dijalankan -> aplikasi target berjalan dan file python berjalan -> webcam aktif dan menangkap foto visitor -> data hasil foto disimpan dalam folder yang telah dibuat



## Cara Kerja Capture Visitor

Dalam proyek ini, terdapat beberapa komponen antara lain: main file yang bernama ConfigCaptureVisitor.py yang akan menghasilkan komponen lain yaitu suatu folder yang berisi file python dan file vbs yang ketiga tersebut memiliki nama yang sama sesuai masukkan user. Penjelasan rinci :

### ConfigCaptureVisitor.py

Dalam file python ini mengimport 2 module built-in python yaitu TkInter, dan os serta 1 module external python yaitu OpenCV.

```
import cv2 as cv
from tkinter import *
from tkinter import messagebox
import os
```

Dalam program ini terdapat 2 Class yaitu Class GUI dan Class Capture.

#### 1. Class GUI :

```
class GUI :
    '''Class untuk membuat GUI yang berguna mengonfigurasi target dari Capture Visitor yang diinginkan'''
    def __init__(self):
        '''Method inisiasi untuk membuat GUI root dan memanggil method lain'''
        self.root = Tk()
        self.root.title("Capture Visitor by Ali Ihsyad R")
        self.root.geometry("350x150")
        self.make_widget()
        self.pack_widget()
        self.root.mainloop()

    def make_widget(self):
        '''Method untuk membuat widget pada GUI'''
        self.label_dir_target = Label(self.root, text="Enter directory of your target application: ")
        self.dir_target = Entry(self.root)
        self.label_file_name = Label(self.root, text="Enter your folder and file name: ")
        self.file_name = Entry(self.root)
        self.enter_button = Button(self.root, text="Create", command=self.make_folder_py_vbs)

    def pack_widget(self):
        '''Method untuk mengpack widget pada GUI'''
        self.label_dir_target.pack()
        self.dir_target.pack()
        self.label_file_name.pack()
        self.file_name.pack()
        self.enter_button.pack()

    def make_folder_py_vbs(self):
        '''Method untuk membuat folder dan file baru pada directory main program'''
        try:
            name = self.file_name.get()
            target = self.dir_target.get().strip("/")
            dir_folder_cv = os.getcwd()[2:]
            os.mkdir(name)
            dir_folder = f'{os.getcwd()}\\{name}'
            dir_py = f'{dir_folder}\\{name}.py'
            dir_vbs = f'{dir_folder}\\{name}.vbs'
            name_pic = f'{name}.png'

            # Untuk membuat file python dan vbs
            with open(f'{dir_py}', 'w') as file:
                file.write(f'import sys\nsys.path.insert(0, r"{dir_folder_cv}")\nfrom ConfigCaptureVisitor import Capture\nrun = Capture(r"{dir_folder}", "{name_pic}")')

            with open(f'{dir_vbs}', 'w') as file:
                file.write(f'Set WshShell = CreateObject("WScript.Shell")\nWshShell.Run chr(34) & "{target}" & chr(34), 0\nWshShell.Run chr(34) & "{dir_py}" & chr(34), 0\nSet WshShell = Nothing')

            self.root.clipboard_append(f'{dir_vbs}')
            self.root.update()
            messagebox.showinfo(title="Information To Do", message=f'This directory :{dir_vbs} has been copied to your clipboard\nYou should paste them in target of your shortcut application target')
            self.root.destroy()

        # Except untuk menangani jika user tidak memasukkan sesuatu ke entry
        except FileNotFoundError:
            messagebox.showerror(message="Please fill your folder and file name")
```

Class ini berguna untuk membuat tampilan GUI yang berfungsi untuk menerima masukkan user yaitu lokasi program aplikasi target dan nama folder serta file yang akan disimpan. Terdapat 1 method inisiasi , 1 method main, dan 3 method buatan yang memiliki fungsi yang berbeda-beda.

1) Method inisiasi ( def \_\_init\_\_() )

```
class GUI :  
    '''Class untuk membuat GUI yang berguna mengonfigurasi target dari Capture Visitor yang diinginkan'''  
    def __init__(self):  
        '''Method inisiasi untuk membuat GUI root dan memanggil method lain'''  
        self.root = Tk()  
        self.root.title("Capture Visitor by Ali Irsyaad N")  
        self.root.geometry("350x150")  
        self.make_widget()  
        self.pack_widget()  
        self.root.mainloop()
```

Method yang berfungsi membuat GUI root, membuat nama judul GUI , mengatur lebar dari GUI, serta memanggil method pembuat widget dan pengepack widget. Lalu membuat GUI root tersebut tetap berjalan (main loop).

2) Method def make\_widget()

```
def make_widget(self):  
    '''Method untuk membuat widget pada GUI'''  
    self.label_dir_target = Label(self.root, text='Enter directory of your target application: ')  
    self.dir_target = Entry(self.root)  
    self.label_file_name = Label(self.root, text='Enter your folder and file name: ')  
    self.file_name = Entry(self.root)  
    self.enter_button = Button(self.root, text='Create', command=self.make_folder_py_vbs)
```

Method yang berfungsi untuk membuat 5 widget yaitu label directory target, entry directory target, label folder and file name, entry folder and file name, dan enter button.

3) Method def pack\_widget()

```
def pack_widget(self):  
    '''Method untuk mengepack widget pada GUI'''  
    self.label_dir_target.pack()  
    self.dir_target.pack()  
    self.label_file_name.pack()  
    self.file_name.pack()  
    self.enter_button.pack()
```

Method yang berfungsi mengepack 5 widget yang telah dibuat agar dapat ditampilkan di GUI root.

#### 4) Method def make\_folder\_py\_vbs()

```
def make_folder_py_vbs(self):
    '''Method untuk membuat folder dan file baru pada directory main program'''
    try:
        name = self.file_name.get()
        target = self.dir_target.get().strip('*')
        dir_folder_cv = os.getcwd()[2:]
        os.mkdir(name)
        dir_folder = f'{os.getcwd()}\\{name}'
        dir_py = f'{dir_folder}\\{name}.py'
        dir_vbs = f'{dir_folder}\\{name}.vbs'
        name_pict = f'{name}.png'

        # Untuk membuat file python dan vbs
        with open(f'{dir_py}', 'w') as file:
            file.write(f'import sys\nsys.path.insert(0,r"{dir_folder_cv}")\nfrom ConfigCaptureVisitor import Capture\nrun = Capture(r"{dir_folder}", "{name_pict}")')

        with open(f'{dir_vbs}', 'w') as file:
            file.write(f'Set WshShell = CreateObject("WScript.Shell")\nWshShell.Run chr(34) & "{target}" & chr(34), 0\nWshShell.Run chr(34) & "{dir_py}" & chr(34),')

        self.root.clipboard_append(f'{dir_vbs}')
        self.root.update()
        messagebox.showinfo(title='Information To Do', message=f'This directory :{dir_vbs} has been copied to your clipboard\nYou should paste them in target of')
        self.root.destroy()

    # Except untuk menhandle jika user tidak memasukkan sesuatu ke entry
    except FileNotFoundError:
        messagebox.showerror(message='Please fill your folder and file name')

    # Except untuk menhandle jika user memasukkan entry yang sudah ada dan ada contain special karakter
    except OSError:
        messagebox.showerror(message='Your folder and file name already exist or can\'t contain any of the following characters:\n \ / : * ? " < > | ')
```

Method yang berfungsi untuk mengakses masukkan user setelah menekan button. Masukkan user tersebut adalah directory target dan nama folder dan file. Setelah itu masukkan user itu akan diolah dan akan dibuat folder sesuai masukkan yang berisi 2 file yaitu file python dan file vbs.

File Python berisi :

```
import sys
sys.path.insert(0,r"C:\Users\Ali Irsyaad\Documents\Perkuliahan\Semester 1\DDP 1\Per TP an\TP 4")
from ConfigCaptureVisitor import Capture
run = Capture(r"C:\Users\Ali Irsyaad\Documents\Perkuliahan\Semester 1\DDP 1\Per TP an\TP 4\ChromeVisitor", "ChromeVisitor.png")
```

Mengimport sys dan menggunakan sys.path.insert() untuk memindahkan directory ke main program yaitu ConfigCaptureVisitor. agar dapat mengimport class Capture dari main program ConfigCaptureVisitor. Setelah itu class Capture tersebut akan digunakan dengan menginisiasinya pada suatu variable run. Intinya file python ini digunakan sebagai program sekunder dari ConfigCaptureVisitor.py agar dapat mengambil class Capture tanpa menampilkan GUI oleh karena itu pada ConfigCaptureVisitor.py terdapat source code:

```
def main():
    run = GUI()

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

Yang berguna agar def main() tidak dijalankan jika program ConfigCaptureVisitor.py dijalankan pada program sekunder (import). Fungsi class Capture akan dijelaskan dibawah.



File Vbs berisi:

```
ChromeVisitor - Notepad
File Edit Format View Help
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run chr(34) & "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" & chr(34), 0
WshShell.Run chr(34) & "C:\Users\Ali Irsyaad\Documents\Perkuliah\Semester 1\DDP 1\Per TP an\TP 4\ChromeVisitor\ChromeVisitor.py" & chr(34), 0
Set WshShell = Nothing
```

Yang intinya berguna untuk menjalankan 2 program dengan tersembunyi (hidden mode). 2 Program tersebut adalah file python diatas dan aplikasi target yang didapatkan dari masukkan user.

## 2. Class Capture:

```
class Capture:
    '''Class untuk menangkap foto dan menyimpannya pada suatu directory'''
    def __init__(self,dir_folder,file_name):
        '''Method untuk menginisiasi lokasi folder dan nama folder serta memanggil method capture'''
        self.dir_folder = dir_folder
        self.file_name = file_name
        self.capture(self.dir_folder, self.file_name)

    def capture(self,dir_folder,file_name):
        '''Method untuk menangkap foto dan menyimpannya pada suatu directory'''
        cam = cv.VideoCapture(0)
        img_counter = 0
        while True:
            frame = cam.read()[1]
            cv.waitKey(1)
            if img_counter == 25 :
                img_directory = f"{dir_folder}\{file_name}"
                cv.imwrite(img_directory, frame)
                cam.release()
                cv.destroyAllWindows()
                break
            img_counter += 1
```

Class ini berguna untuk menangkap visual (foto) lewat webcam lalu menyimpannya pada sebuah directory yang ditentukan user. Pada class ini terdapat 1 method inisiasi dan 1 method buatan

### 1) Method def \_\_init\_\_():

```
class Capture:
    '''Class untuk menangkap foto dan menyimpannya pada suatu directory'''
    def __init__(self,dir_folder,file_name):
        '''Method untuk menginisiasi lokasi folder dan nama folder serta memanggil method capture'''
        self.dir_folder = dir_folder
        self.file_name = file_name
        self.capture(self.dir_folder, self.file_name)
```

Method yang berfungsi untuk menginisiasi masukkan kepada object pemanggil dan memanggil method capture dengan argument masukkan user.

## 2) Method def capture():

```
def capture(self, dir_folder, file_name):  
    '''Method untuk menangkap foto dan menyimpannya pada suatu directory'''  
    cam = cv.VideoCapture(0)  
    img_counter = 0  
    while True:  
        frame = cam.read()[1]  
        cv.waitKey(1)  
        if img_counter == 25 :  
            img_directory = f"{dir_folder}\\{file_name}"  
            cv.imwrite(img_directory, frame)  
            cam.release()  
            cv.destroyAllWindows()  
            break  
        img_counter += 1
```

Fungsi ini berfungsi untuk menangkap visual (foto) dan menyimpannya pada suatu directory yang ditentukan oleh user.

Jadi setiap komponen berhubungan secara kompleks satu sama lain sehingga menghasilkan tangkapan foto jika user menekan shortcut target aplikasi namun program aplikasi tersebut tetap jalan.

## Pengembangan Lebih Lanjut:

1. Efisiensi code dan komponennya  
Mungkin masih ada code yang kurang efisien dan meminimalisir komponennya jika masih bisa agar proyek yang lebih baik
2. Save foto visitor bisa lebih dari 1 untuk setiap target aplikasi  
Saat ini jika program target dijalankan berkali-kali maka foto visitor yang berhasil ditangkap akan disimpan dengan menimpa (menghapus) foto lama.
3. Foto dapat dikirimkan via email  
Data hasil program dapat dikirimkan secara sembunyi ke email user ketika pada visitor membuka program target mereka.
4. Dibuka secara umum sebagai aplikasi open source